UNIVERSIDAD CATÓLICA DE SANTA MARÍA

FACULTAD DE CIENCIAS E INGENIERÍAS FÍSICAS Y FORMALES

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

“Estudio y análisis de entornos comerciales mediante la evaluación, comparación y experimentación de algoritmos de minería de datos”

Tesis presentada por el bachiller:

OBANDO VELÁSQUEZ, DANIEL ANDRÉ

Para optar el Título Profesional de:

INGENIERO DE SISTEMAS

Asesor: CALDERÓN RUIZ, GUILLERMO E.

AREQUIPA - PERÚ

2017

PRESENTACIÓN

Sra. Directora de la Escuela Profesional de Ingeniería de Sistemas

Ing. Karina Rosas Paredes

Sres. Miembros del Jurado Examinador de Tesis

Ing. Guillermo Calderón Ruiz

Ing. José Sulla Torres

De conformidad con las disposiciones del reglamento de grados y títulos del programa profesional de Ingeniería de Sistemas, remito a vuestra consideración el estudio de investigación titulado: "ESTUDIO Y ANÁLISIS DE TENDENCIAS COMERCIALES MEDIANTE LA EVALUACIÓN, COMPARACIÓN Y EXPERIMENTACIÓN DE ALGORITMOS DE MINERÍA DE DATOS", el mismo que de ser aprobado me permitirá optar por el título profesional de Ingeniero de Sistemas.

Arequipa, Enero del 2017

Daniel André Obando Velásquez

DEDICATORIA

A mis amorosos padres Alexander e Ivone.

Por haberme motivado e inspirado desde el inicio, por siempre apoyar todos mis sueños, por darme un hogar que no todos gozan lleno de amor, comprensión y buenos valores, por estar siempre ahí para mí, pero más que nada, por ser los mejores padres del mundo.

A mis hermanos Gonzalo y Gala.

Por haber incentivado la competitividad en mí desde que tengo uso de razón, por siempre estar a mi lado, por ser los primeros compañeros que tuve en mi vida y por el inmenso amor que nos tenemos.

A mis tías Hilda, Vilma y Marcela

Por ser mis segundas madres, por darme tantas lecciones importantes de vida, por ser siempre mis confidentes, por siempre apoyarme en todo lo que este a su alcance y por el inmenso amor y estima que siempre me han demostrado.

A mis primas Karla y Vanesa

Por apoyarme siempre que he necesitado ayuda para superar dificultades.

AGRADECIMIENTOS

A mis padres, hermanos y familiares, que desde temprana edad inculcaron en mis buenos valores y el deseo de superarme cada día, nunca ser conformista, aprovechar cada oportunidad y miles de cosas más que no podría terminar de mencionar.

A mi amigo Edward, por compartir conmigo tantas cosas, por ser el único que sabe casi todo sobre mí y aún sigue conmigo, por la confianza y amistad que nos tenemos, por ser mi mejor amigo.

A mi asesor Guillermo Calderón Ruiz, por ser el docente que más me inspiro en mi formación universitaria, por darme a entender que el mundo es un lugar muy grande y competitivo y uno siempre debe buscar ser el mejor, y por ayudarme en todo el desarrollo del presente trabajo.

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE SIGLAS

CRM (Customer Relationship Managment)	Gestión de relaciones con los clientes
LDA (Latent Dirichlet Allocation)	Asignación Latente Dirichlet
DBMS (DataBase Managment System)	Sistema de gestión de base de datos
OLTP (Online Transaction Proccesing)	Procesamiento de transacciones en línea
OLAP (Online Analitical Proccesing)	Procesamiento analítico en línea
LAD (Logical Analysis of Data)	Análisis lógico de datos.
CART (Classification And Regression Trees)	Árboles de clasificación y regresión
CLARA (Clustering Large Applications)	Agrupación de aplicaciones grandes
CLARANS (Clustering Large Applications based upon Randomized Search)	Agrupación de aplicaciones grandes basado en la búsqueda aleatoria.
PAM (Partitioning Around Medoids)	Particionamiento alrededor de medoides
CURE (Clustering Using Representatives)	Agrupamiento utilizando representantes
ROCK (Robust Clustering using Link)	Agrupamiento robusto utilizando enlaces
EM (Expectation – Maximization)	Expectación - Maximización

RESUMEN

En la actualidad el campo de la minería de datos tiene una gran importancia en el entorno comercial y empresarial altamente competitivo. Aplicaciones de minería de datos son amplia y frecuentemente utilizadas en marketing o comercialización directa, comercio electrónico, gestión de relaciones con los clientes (Customer Relationship Managment CRM), telecomunicaciones y en el sector financiero.

Muchas empresas de hoy, al no tener pleno conocimiento del gran potencial que significa el utilizar técnicas de minería de datos, les suena como el típico tipo de actividad monótona que requiere de un gran equipo, una gran cantidad de información y poca supervisión humana; la minería de datos, sin embargo, es un proceso crucial para cualquier organización que requiere una gran cantidad de tiempo y paciencia. Por otro lado, también existe una gran cantidad de empresas que año tras año recolectan grandes cantidades de información que simplemente son almacenadas; y decisiones de gran importancia para el devenir de la empresa son tomadas meramente en suposiciones y corazonadas, algo que hoy en día, especialmente en el sector comercial de cualquier país, representa una gran desventaja competitiva.

El objetivo principal de la investigación es identificar los algoritmos de minería de datos más eficientes para el análisis de entornos comerciales, sin embargo, también se busca comprobar que el uso de dichas herramientas es de vital importancia para la toma de decisiones de cualquier institución comercial.

Para llevar a cabo la presente investigación se enfocó principalmente en dos etapas, una primera etapa investigativa y de análisis que permitió identificar en primera instancia aquellos algoritmos que cumplirían con procesar y analizar eficientemente repositorios o bases de datos asociados a entornos comerciales, respaldados a su vez por trabajos y artículos científicos; y una segunda etapa experimental en la que los algoritmos identificados en la primera etapa fueron puestos a prueba a través del software WEKA y utilizando repositorios de datos diseñados para la experimentación, es gracias a esta etapa que se corroboró los resultados obtenidos en la etapa de análisis. Se identificó científica y experimentalmente aquellos algoritmos de minería de datos que trabajan de manera más efectiva, es decir, aquellos generan la mejor calidad de conocimiento y brindan los mejores resultados respecto a las características tomadas en consideración dependiendo del tipo de algoritmo (clasificación, clusterización o asociación); tomando en cuenta su aplicación, específicamente, a entornos asociados al sector comercial. A su vez se demostrará el gran potencial que tienen estos en su aplicación a grandes y pequeños conjuntos de datos, y lo crucial que es el conocimiento que se puede extraer para cualquier empresa, compañía u organización.

Los resultados obtenidos reflejan lo variante que pueden ser los resultados, ya sean modelos, conjuntos o reglas, dependiendo de las características particulares de cada repositorio, así mismo se demostró la importancia del conocimiento obtenido a partir de su procesamiento. Las conclusiones finales del trabajo se enfocaron en resolver los objetivos planteados así como los hallazgos realizados durante su realización.

Palabras Clave: Minería de datos, Almacén de datos, Entornos Comerciales, Evaluación de algoritmos, Repositorios de datos.

ABSTRACT

At present the field of data mining has a great importance in the commercial and business enviroment highly competitive. Data mining applications are widely and frequently used in direct marketing, e-commerce, customer relationship managment, telecommunications and in the financial sector.

Many companies today, not having full knowledge of the great potential of using data mining techniques, sounds like the typical type of monotonous activity that requires a great team, a lot of information nad Little human supervisión; data mining, however, is a crucial process for any organization that requires a great deal of time and patiente. On the other hand, there are also a large number of companies that year after year collect large amounts of information that are simply sotred, and decisions of great importance for the future of the Company are taken merely in suppositions and hunchings, something that today, especially in the commercial sector of any country, represents a great comptetitive disadvantage.

The main objective of the research is to identify the most efficient mining algorithms for the analysis of commercial environments, however, it is also sought to verify that the use of such tolos is of vital importance for the decisión making of any commercial institution.

In order to carry out the present investigation it was mainly focused in two stages, a first investigation and analysis stage that allowed to identify in the first instance those algorithms that would fulfill with efficent processing and analysis of repositories or databases associated

to commercial enviroments, supported in turn for scientific papers and articles; and second experimental stage in which the algorithms identifed in the first stage were tested through the WEKA software and using data repositories designed for the esperimentation, it is thanks to this stage that the results obtained in the analysis stage were corroborated. It was scientifically and experimentally identified those data mining algorithms that work more effectively, that is, those generate the best quality of knowledge and provide the best results regarding the characteristics taken into account depending on the type of algorithm (classification, clustering or association); taking into account its application, specifically, to environments associated with the commercial sector. IN turn will demonstrate the great potential they have in their application to large and small datasets, and how crucial is the knowledge that can be extracted for any company or organization.

The obtained results reflect the variant that can be the results, either models, sets or rules, depending on the particular characteristics of each repository, as well as demonstrated the importance of the knowledge obtained from its proccesing. The final conclusions of the work were focused on solving the objectives set as well as the findings made during their realization.

Keywords: Data Mining, Data Warehouse, Business Environments, Evaluation of Algorithms, Data Repositories.

INTRODUCCIÓN

Con el pasar de los años cada vez existen herramientas más poderosas de recolección de datos, herramientas utilizadas por las empresas y que, sin embargo, no son explotadas en su máximo potencial. Así como éstas, la minería de datos es una nueva y poderosa tecnología con el gran potencial de ayudar a empresas, compañías u organizaciones a centrarse en la parte más importante de los datos que han recolectado, no solo de transacciones realizadas por la empresa, sino también sobre el comportamiento de sus clientes y potenciales clientes. Estamos en la era de la información, cualquier persona que se desenvuelva en el campo de la computación lo sabe, así como que el conocimiento se está convirtiendo en el recurso fundamental de cada vez más organizaciones, ya que esta siempre brindara una ventaja competitiva y abrirá el camino para la gestión del conocimiento.

Algoritmos de minería de datos, ya sean de clasificación como el algoritmo J48, de clusterización como el algoritmo K-Means o de asociación como el algoritmo A priori, nos permiten analizar correctamente, bajo diferentes enfoques, la información que es recolectadada durante periodos de tiempo determinados; sin embargo al tratar dicho tema surgen preguntas como: ¿Efectivamente el análisis de entornos comerciales a través de técnicas de minería de datos ayudaría a una empresa o institución comercial a la buena toma de decisiones?, ¿El uso de técnicas de minería de datos para el análisis de repositorios de datos brinda un conocimiento significativo que otro tipo de análisis no brindaría, justificando de esta manera el costo y capital humano empleado para ello? o ¿Emplear diferentes algoritmos de minería de datos pertenecientes a una misma categoría, por ejemplo “clusterización”, afectan de manera significativa a los resultados obtenidos?

En el presente trabajo de investigación, específicamente en la etapa de experimentación, se va a utilizar la plataforma de software WEKA, una poderosa y de las más utilizadas herramientas de minería de datos, dado que, es un software libre bajo la licencia pública general de GNU, al estar implementado en java es capaz de correr en casi cualquier plataforma, contiene una de las más extensas colecciones de técnicas para el procesamiento y modelado de los datos, finalmente es fácil de utilizar gracias a su interfaz gráfica amigable con el usuario; así mismo, se utilizaran repositorios de datos modelados para la experimentación y obtenidos a través de la red, éstos estarán estrechamente ligados al sector empresarial o comercial, y contaran con distintas características con el fin de analizar diferentes situaciones bajo un mismo contexto.

La tesis constará de 4 capítulos; en el primer capítulo se hace el planteamiento teórico; en el segundo capítulo se desarrollan los fundamentos teóricos básicos que le dan sustento a la tesis; el tercer capítulo se enfoca en el análisis teórico de los algoritmos estudiados, analizando en detalle cada uno de sus aspectos y características particulares, además se realizará una comparación entre aquellos pertenecientes a la misma categoría y se dará como resultado final los identificados como más eficaces para el estudio y análisis de tendencias comerciales; en el cuarto capítulo se hace un análisis experimental a los algoritmos identificados en el capítulo tres, corroborando que los resultados previamente obtenidos eran o no acertados, así mismo, se analizarán todos los aspectos que conlleva el trabajar con este tipo de repositorios.

CAPÍTULO I:
PLANTEAMIENTO TEÓRICO

1.1. PLANTEAMIENTO DE LA INVESTIGACIÓN

1.1.1. Planteamiento del Problema

Como muchos otros fenómenos impulsados por la revolución digital, la capacidad de recolectar datos de los consumidores y de almacenar dicha información ha aumentado de manera exponencial en los últimos años (Advertising Age, 2016), razón por la cual la utilización de técnicas de minería de datos se ha popularizado de manera proporcional a este hecho.

De acuerdo a una encuesta realizada por SurveyMe entre 2500 negocios que utilizan su aplicación, un tercio de las empresas del mundo considera que saber qué preguntarle a los consumidores es uno de los mayores desafíos a la hora de recolectar datos; sin embargo, además de este problema, otro de los desafíos importantes para las empresas es saber qué hacer con la enorme cantidad de datos correspondientes a ventas y marketing obtenidos en periodos de tiempo establecidos, por ejemplo los obtenidos a través de plataformas web (Advertising Age, 2016).

Con este proyecto se busca realizar una evaluación y análisis de algoritmos de minería de datos para el estudio de entornos comerciales

a través de repositorios orientados a este ámbito en general, brindando como salida final un informe con los resultados obtenidos.

1.1.2. Objetivos de la Investigación

El objetivo principal de la investigación es identificar los algoritmos de minería de datos más eficientes para el análisis de entornos comerciales, siendo estos los que procesen la data y muestren los mejores resultados para los usuarios de dichos algoritmos, en este caso empresas o instituciones comerciales; el análisis abarcará tanto el lado investigativo, a través de trabajos y artículos científicos como base para el estudio, como experimental, a través de la aplicación de los algoritmos seleccionados en la etapa de análisis en repositorios de datos.

Los objetivos específicos son:

a) Identificar y evaluar los algoritmos de minería a ser utilizados en la etapa de análisis, dada su relevancia respecto al estudio y análisis de tendencias comerciales, para finalmente obtener aquellos que, teóricamente, actuarían de manera eficiente en nuestra base de datos transaccional.

b) Realizar un análisis experimental de los algoritmos identificados y seleccionados en la etapa de análisis a través de repositorios de datos orientados al ámbito comercial, y diseñados para la experimentación. Se comprobará si estos efectivamente brindarán los resultados esperados, tanto teórica como experimentalmente.

1.1.3. Preguntas de Investigación

a) ¿Efectivamente el análisis de entornos comerciales a través de técnicas de minería de datos ayudaría a una empresa o institución comercial a la buena toma de decisiones?

b) ¿El uso de técnicas de minería de datos para el análisis de repositorios de datos brinda un conocimiento significativo que otro tipo de análisis no brindaría, justificando de esta manera el costo y capital humano empleado para ello?

c) ¿Emplear diferentes algoritmos de minería de datos pertenecientes a una misma categoría, por ejemplo "clusterización", afectan de manera significativa a los resultados obtenidos?

1.1.4. Línea y Sub-Línea de Investigación

a) Línea: Sistemas de Información y Bases de Datos

b) Sub-Línea: Data Mining – Data Warehouse

1.1.5. Palabras Clave

a) Minería de datos

b) Almacén de datos

c) Entornos comerciales

d) Evaluación de Algoritmos

e) Repositorios de datos

1.1.6. Solución Propuesta

a. Justificación e Importancia

Los resultados obtenidos a partir de esta investigación permitirán a estudiantes comprender mejor el uso de técnicas y herramientas de minería de datos, acceder a un caso de aplicación a datos reales, y entender la importancia de este campo para el manejo de información de cualquier empresa o institución comercial.

Este estudio también contribuirá al campo de la computación, ya que producirá como resultado un estudio comparativo de algunos de los algoritmos de minería de datos más utilizados por estudiantes e investigadores aplicados a un entorno específico, en este caso un entorno comercial, permitiendo de esta manera la posibilidad de realizar estudios e investigaciones a los algoritmos tratados respecto a otros entornos estudiados.

La información obtenida de la investigación también contribuirá al desarrollo de aplicaciones, software o técnicas que permitan a empresas e instituciones comerciales desarrollarse y tomar decisiones que contribuyan en su devenir corporativo en base a estudios científicos, dejando de lado métodos ya considerados “arcaicos” como el de seguir una simple corazonada.

b. Descripción de la solución

Durante la investigación se realizará una evaluación comparativa de algoritmos de minería de datos, determinando de esta manera, teórica y experimentalmente, cuáles trabajarían de manera más eficiente al ser aplicados a repositorios de datos orientados a entornos comerciales. Así mismo, durante dicha investigación se estudiara aspectos relevantes a tener en consideración al momento de aplicar los algoritmos, características de los reportorios de datos que afecten de manera positiva o negativa a los resultados, beneficios que conllevarían su utilización, etc.

1.2. FUNDAMENTOS TEÓRICOS

1.2.1. Estado del Arte

Claudia Elena Dinucă (2011) afirma que, hoy en día, la web es parte importante de nuestra vida diaria, tanto así que actualmente es el mejor medio para hacer negocio. También comenta que cada vez es más común que las grandes empresas replanteen su estrategia de negocio haciendo uso de la web para mejorarlo, además actividades ejercidas por la web ofrecen la oportunidad que potenciales clientes o socios puedan encontrar sus productos o negocios. En su investigación también propone que las empresas triunfadoras son aquellas que se han dado cuenta que la transacciones de comercio electrónico son más que solo compra y venta, sino que adecuadas estrategias son esenciales para

mejorar su capacidad competitiva; y una de las técnicas más eficaces que se usa para este propósito es el Data Mining o Minería de Datos.

Sobre este tema también nos hablan Onur Doğan, Hakan Aşan y Ejder Ayçın (2015), quienes nos dicen que en el mundo competitivo de hoy en día las organizaciones necesitan tomar las decisiones correctas para prolongar su existencia; el uso de métodos no científicos y tomar decisiones emocionales ya no es viable, el uso de métodos científicos es lo que se usa actualmente en el ámbito competitivo, dentro del cual muchos modelos de soporte de decisiones todavía se están desarrollando con el fin de ayudar a la toma de decisiones. En su investigación comentan que es fácil recoger una cantidad masiva de datos, sea cual sea el rubro de la organización; el problema recae en cómo utilizar estos datos para lograr avances económicos.

Rafiqul Islam y Ahsan Habib (2015) plantean un caso en el que el uso de algoritmos de tipo “Decision Tree” o árboles de decisión son eficaces para el procesamiento de datos referentes a procesos bancarios como la predicción de potenciales sectores de negocios para el trámite de préstamos. En su proyecto ellos trabajan con uno de los principales objetivos de cada organización, retener clientes existentes y alcanzar nuevos, sin embargo algunos clientes suelen ser de alto riesgo, lo que se clasifica en factores cualitativos y cuantitativos para la empresa; para lograr el análisis planteado se utilizó un enfoque de minería de datos adoptado para el procesamiento de los puntos críticos, donde la técnica

de clasificación "Árbol de Decisión" se utilizó para desarrollar el modelo y realizar las pruebas correspondientes en el software WEKA.

Safia Abbas (2015) propone que el uso de algoritmos del tipo "Decision Tree" también es eficaz para el procesamiento de datos reales asociados al Marketing. En su estudio expone que, actualmente, los gerentes de marketing están en necesidad de incrementar las campañas de marketing, mientras que las organizaciones evaden los gastos y por ende la expansión del negocio; con el fin de resolver este problema ella propone el uso de técnicas de minería de datos como factor determinante, haciendo específicamente uso de la técnica "Árbol de decisión" en un conjunto de datos de marketing reales obtenidos a partir de una campaña de marketing portuguesa. El experimento finalmente brindó una detección de los datos más significativos y reglas predictivas.

Suhail Ansari, Ron Kohavi, Llew Mason, y Zijian Zheng (2000) afirma que las herramientas de minería de datos ayudan al descubrimiento de patrones en los datos, algo vital en el ámbito competitivo actual ya que hasta hace poco se determinó que empresas que se han concentrado en la construcción de herramientas horizontales de modelamiento de minería de datos han tenido poco éxito comercial. Ellos también nos señalan en su trabajo que particularmente el campo del comercio es el dominio asesino para la minería de datos, ya que este es ideal debido a que muchos de los ingredientes necesarios para una minería de datos exitosa se satisfacen fácilmente: los registros de los

datos son abundantes, la recopilación electrónica proporciona datos confiables, el conocimiento extraído se puede convertir fácilmente en acciones y el retorno de la inversión realizada puede medirse.

B. Naveena Devi y O. Sreevani (2010) afirman en su trabajo que predecir el comportamiento de los clientes es una de las más importantes tecnologías con las que debe contar una empresa u organización. Los resultados de la predicción pueden ser utilizados para la toma de decisiones, la mejora de la estrategia de marketing, promoción, suministro de productos, obtención de información de marketing, previsión de las tendencias del mercado y el aumento de la competitividad de las empresas.

1.3. MARCO METODOLÓGICO

1.3.1. Alcances y Limitaciones

a) Alcances

El presente estudio evaluará algoritmos de minería de datos de clasificación, clusterización y asociación, referentes al procesamiento de información para la obtención de patrones frecuentes e información relevante en repositorios de datos orientado al ámbito comercial. Dicho procesamiento se realizará a través de dos etapas: una etapa investigativa que buscar determinar los mejores algoritmo para su aplicación en este campo a través de trabajos y artículos científicos, es decir teóricamente; posteriormente se tendrá una etapa experimental

que buscará probar experimentalmente que los resultados obtenidos en la etapa investigativa son consistentes.

b) Limitaciones

La falta de almacenes de datos reales asociados empresas del ámbito comercial, debido a que no existe un convenio actual que permita acceder a sus instalaciones con el fin de recabar información de sus repositorios de datos para ser analizados.

1.3.2. Aporte

Una vez culminado el proyecto de investigación se obtendrá un análisis corroborado de los algoritmos seleccionados para el procesamiento y obtención de patrones frecuentes e información relevante en repositorios de datos orientados al ámbito comercial, este análisis abarcará varias características al igual que la experimentación, además al final se contará con diversas conclusiones que incluirán desde aquellas que tengan correlación directa con el resultado final esperado hasta los hallazgos realizados durante el proceso.

1.3.3. Tipo y Nivel de Investigación

a) Tipo: Aplicada

La presente investigación se realizará para determinar la eficiencia de algoritmos de minería de datos referentes a la clasificación, clusterización y asociación con el fin de analizar

patrones frecuentes respecto a repositorios de datos orientados al ámbito comercial.

b) Nivel: Comparativa / Experimental o Evaluatoria

Con esta investigación se pretende realizar una evaluación teórica y experimental de varios algoritmos de minería de datos en base a su eficiencia de predicción a la hora de hallar patrones frecuentes aplicados a repositorios de datos orientados a un entorno comercial, brindando como resultado final una comparación de resultados y conclusiones finales.

1.3.4. Población y Muestra o Universo

a) Popularidad de Noticias en Linea (Online News Popularity)

Conjunto de datos con características de artículos publicados en internet por un periodo de 2 años.

b) Marketing Bancario (Bank Marketing)

Datos relacionados a campañas de marketing de una institución bancaria portuguesa.

1.3.5. Métodos, Técnicas e Instrumentos de Recolección de Datos

La tabla final mostrará los resultados del análisis realizado, mostrando a través de ella una comparación entre los algoritmos estudiados y el resultado generado a partir de la experimentación realizada a los algoritmos seleccionados en la etapa de análisis.

Los datos que se emplearán en la etapa de experimentación se obtendrán a partir de repositorios de datos diseñados para la experimentación y obtenidos de la página: http://archive.ics.uci.edu/ml/ correspondiente a UCI Machine Learning Repository.

CAPÍTULO II: MARCO TEÓRICO

En este capítulo se verá el marco teórico; el marco teórico que fundamenta esta investigación proporcionará al lector una idea más clara acerca del tema "Aplicación de algoritmos de minería de datos", así como una breve introducción a lo conocido como un "entorno comercial" desde el punto de vista de análisis de los datos. Se encontrarán conceptos muy básicos, complementarios y específicos.

2.1. DEFINICIONES DE ENTORNO COMERCIAL

Para comprender que es lo que diferencia un repositorio o base de datos orientada a un entorno comercial respecto a repositorios orientados a otros entornos hay que tener en cuenta ciertos aspectos o características; dichas características pueden ser identificadas a través de simples preguntas como las propuestas por The National Academy Press (1999), donde a través de ellas son capaces de hallar las principales características de repositorios de datos orientados a diferentes entornos o campos laborales. Las características halladas fueron las siguientes:

i. Este tipo de repositorios están diseñados para almacenar información de transacciones y procedimientos de organizaciones cuyo propósito primario es el comercio, es decir, comprar y/o vender bienes o servicios o generar publicidad o anuncios que transmiten un mensaje orientado a captar compradores o usuarios.

ii. Algunas de las principales funciones de este tipo de repositorios son: cumplir con gestionar todas las transacciones de compra y/o venta de bienes o servicios, así como de operaciones internas y campañas de publicidad de la organización, proveer la información necesaria para la correcta gestión de procedimientos importantes como el abastecimiento de mercancía, proporcionar información confiable, oportuna y relevante para los distintos sectores de la organización, etc.

iii. La principal fuente de datos para este tipo de repositorios son las cientos o miles de transacciones de compra y/o venta de bienes y servicios realizadas y registradas por día en cualquier institución comercial, las cuáles comúnmente están asociadas al producto o servicio proporcionado, así como a los consumidores, fecha de la transacción, etc.

iv. Las principal barrera que tiene este tipo de repositorios para conseguir sus datos es el error humano, dado que muchas veces las transacciones suelen tener campos vacíos o erróneamente completados debido a la gran cantidad que se suelen registrar al día, esto da lugar a típicos problemas a hora del análisis como lo son el Ruido (Noise) o Valores Faltantes (Missing Values), problemas vistos con mayor frecuencia en este tipo de repositorios.

v. Una de las principales características de este tipo de repositorios es que es la misma organización o empresa la que suele ser la única fuente de datos; otros repositorios suelen combinar datos generados por la misma empresa con fuentes externas.

vi. Dada la presente era tecnológica, la mayoría de repositorios orientados a este entorno suele registrar datos que tienen como fuente centros de distribución física y medios electrónicos, es de conocimiento general que el comercio electrónico, o lo que es lo mismo adquirir productos o servicios mediante internet, es una práctica cada vez más común entre consumidores; el realizar y registrar transacciones tanto a través de medios físicos como virtuales no es común en la mayoría de repositorios.

vii. Algunos de los principales problemas que enfrenta este tipo de repositorios son: como se mencionó anteriormente, los problemas de Ruido y Valores faltantes, también es común encontrar conjuntos de datos sesgados debido a que por temporadas las personas suelen consumir en mayor o menor cantidad cierto tipo de productos o servicios.

viii. Este tipo de repositorios suele tener un acceso concurrente por parte de múltiples usuarios mayor al de otros tipos de repositorios de datos, dado que otros tipos de repositorios suelen ser manipulados únicamente por parte de una misma organización, sin embargo, al estar en constante contacto con medios virtuales y físicos, el nivel de concurrencia de manipulación de datos en este tipo de repositorios suele ser un poco mayor.

Los conceptos básicos a tener en consideración son:

2.1.1. TRANSACCIÓN

Una transacción, en nuestro contexto, es una secuencia de intercambio de información y trabajo relacionado, la cual es tratada como una unidad con el propósito de satisfacer una solicitud (WhatIs.com, 2016). Una transacción puede estar relacionada al ámbito financiero, logístico o de trabajo de una empresa, pero en el entorno que estamos estudiando usualmente estarán relacionadas a las órdenes de compra.

2.1.2. DATOS TRANSACCIONALES

Los datos transaccionales, en el contexto de gestión de datos (minería de datos), es la información registrada de transacciones. Como parte de los registros de transacciones, los datos transaccionales se agrupan con los datos maestros y los datos de referencia (WhatIs.com, 2016).

2.1.3. DATOS MAESTOS

Los datos maestros son los “datos núcleo” que son esenciales para las operaciones en un negocio o unidad de negocio específicos. Los tipos de información tratados como datos maestros varían de una industria a otra e incluso de una empresa a otra dentro de la misma industria (Search Data Managment, 2016).

2.1.4. DATOS DE REFERENCIA

Los datos de referencia, en el contexto de gestión de los datos, son los objetos de datos relevantes para las transacciones, consisten en los conjuntos de valores, estados o esquemas de clasificación. Los datos de referencia son generalmente uniformes en toda la empresa y pueden crearse en un país o por organismos de normalización externos (WhatIs.com, 2016).

2.2. DEFINICIONES DE MINERÍA DE DATOS

2.2.1. MINERÍA DE DATOS

Este concepto se refiere a la extracción o "minería" de conocimiento a partir de grandes cantidades de datos. El término es en realidad un nombre inapropiado, teniendo en cuenta por ejemplo que la extracción de oro de rocas o arena es referida como minería de oro en lugar de minería de rocas o arena; por tanto la minería de datos debería haber sido nombrada minería de conocimiento a partir de los datos, o para resumir "Minería del Conocimiento" (Jiawei Han y Micheline Kamber, 2006).

2.2.2. PATRONES FRECUENTES

Son patrones que se producen con frecuencia en los datos; hay muchos tipos de patrones de frecuencia, incluyendo los conjuntos de elementos, subsecuencias y subestructuras. Un conjunto de elementos frecuentes típicamente se refiere a un conjunto de elementos que aparecen con frecuencia juntos en un conjunto de datos transaccionales. Una subsecuencia ocurre con frecuencia cuando, por ejemplo, existe un patrón donde los clientes tienden a comprar primero un PC, seguido de una cámara digital y luego una tarjeta de memoria, este es un patrón secuencial frecuente. Una subestructura puede referirse a diferentes forma estructurales, tales como árboles o gráficos, los cuales pueden combinarse con los conjuntos de elementos o las subsecuencias; si una subestructura se produce con frecuencia se le llama un patrón estructura frecuente, la minería de patrones frecuentes conduce al descubrimiento de asociaciones interesantes y correlaciones dentro de los datos (Jiawei Han y Micheline Kamber, 2006).

2.2.3. CLASIFICACIÓN Y PREDICCIÓN

La clasificación es el proceso de encontrar un modelo (o función) que describa y distinga las clases o conceptos de los datos, con el fin de ser capaz de usar el modelo para predecir la clase de los objetos cuya etiqueta de clase es desconocida; el modelo derivado se basa en un conjunto de datos de entrenamiento. Mientras que la clasificación predice etiquetas categóricas (discretas, no ordenadas), los modelos de predicción se utilizan para predecir los valores de datos numéricos que faltan o no están disponibles (Jiawei Han y Micheline Kamber, 2006).

2.2.4. LIMPIEZA DE DATOS

También conocida como depuración de los datos, se refiere al pre-procesamiento de los datos con el fin de remover o reducir el ruido mediante técnicas como el suavizamiento (smoothing), y el tratamiento de valores perdidos (missing values) mediante, por ejemplo, la sustitución de un valor faltante con el valor que ocurre con mayor frecuencia para ese atributo (Jiawei Han y Micheline Kamber, 2006).

2.2.5. ANÁLISIS DE RELEVANCIA

Muchos de los atributos de los datos pueden ser redundantes; el análisis de correlación se puede utilizar para identificar si dos atributos dados están estadísticamente relacionados. Una base de datos también puede contener atributos irrelevantes; la selección de subconjuntos de atributo puede ser utilizada para encontrar un conjunto reducido de atributos, de tal manera

que la distribución de probabilidad resultante de las clases de los datos sea lo más cercana posible a la distribución original. Por lo tanto, el Análisis de Relevancia, en la forma de análisis de correlación y selección de subconjuntos de atributo, puede ser utilizado para detectar atributos que no contribuyen a las tareas de clasificación y predicción (Jiawei Han y Micheline Kamber, 2006).

2.2.6. TRANSFORMACIÓN DE LOS DATOS Y REDUCCIÓN

Los datos pueden ser transformados por normalización, particularmente cuando se utilizan métodos que implican mediciones de distancia en la etapa de aprendizaje (Jiawei Han y Micheline Kamber, 2006). En este paso se construyen nuevos atributos a partir de los atributos originales, facilitando una mejor interpretación de la información (Miguel Cárdenas Montes, 2016).

2.2.7. NORMALIZACIÓN

El atributo es escalado a un rango específico, normalmente de -1 a 1 o de 0 a 1, donde 1 representa al caso más general. La normalización es empleada cuando se tienen atributos con órdenes de magnitud muy diferentes. Gracias a la normalización se evita que atributos con valores más altos ganen un peso significativamente más importante en el modelo final que aquellos valores más bajos (Miguel Cárdenas Montes, 2016).

2.2.8. DISCRETIZACIÓN

El atributo es transformado de valores numéricos en valor valores categóricos, de esta forma se reduce el número de posibles valores. La Discretización suaviza el efecto del ruido y permite modelos más simples (Miguel Cárdenas Montes, 2016).

2.2.9. DERIVACIÓN

La derivación permite crear nuevos atributos partiendo de otros anteriores, esto se realiza a través de alguna operación matemática: por ejemplo agrupamiento de valores de tiempo en unidades de orden superior (segundos en minutos), agrupamiento de valores (meses en trimestres), reemplazar valores por medias (suavización), etc., (Miguel Cárdenas Montes, 2016).

2.2.10. ANÁLISIS CLÚSTER

A diferencia de la clasificación y predicción, que analizan objetos de datos con clase etiquetada, el clustering analiza los objetos de datos sin la necesidad de consultar una etiqueta de clase conocida. En general, las etiquetas de clase no están presentes en los datos de entrenamiento, simplemente porque al inicio estas no se conocen; clustering se puede utilizar para generar dichas etiquetas. Los objetos están clusterizados (agrupados) basándose en el principio de maximizar la similitud intraclase y la minimización de la similitud interclase; es decir, las agrupaciones o clústeres de objetos están formadas de tal manera que los objetos dentro de un clúster tienen una alta similitud unos con otros, pero son muy diferentes

a objetos dentro de otros clústeres. Cada grupo que se forma se puede ver como una clase de objetos, de la cual se pueden derivar normas (Jiawei Han y Micheline Kamber, 2006).

2.2.11. ESCALABILIDAD

Muchos algoritmos de agrupamiento o clustering funcionan bien en pequeños conjuntos de datos, los cuales contienen menos de varios cientos de objetos de datos; sin embargo, una base de datos grande puede contener millones de objetos. Debido que pueden dar lugar a resultados sesgados es necesario algoritmos de clustering escalables (Jiawei Han y Micheline Kamber, 2006).

2.2.12. CAPACIDAD PARA HACER FRENTE A DISTINTOS TIPOS DE ATRIBUTOS

Muchos algoritmos están diseñados para agrupar datos en base a un intervalo numérico; sin embargo, las aplicaciones pueden requerir la agrupación de otros tipos de datos, tales como binaria, categórica, datos ordinales, o incluso la mezcla de estos tipos de datos (Jiawei Han y Micheline Kamber, 2006).

2.2.13. DESCUBRIMIENTO DE CLÚSTERES DE FORMA ARBITRARIA

Muchos algoritmos determinan los clústeres basados en las medidas de distancia Euclideana o de Manhattan. Algoritmos basados en este tipo de medidas de distancia tienden a encontrar clústeres esféricos con un tamaño

y densidad similar, sin embargo un grupo podría ser de cualquier forma. Es importante desarrollar algoritmos que puedan detectar grupos de forma arbitraria (Jiawei Han y Micheline Kamber, 2006).

2.2.14. REQUISITOS MÍNIMOS EN LOS PARÁMETROS DE ENTRADA

Muchos algoritmos de clustering requieren que los usuarios introduzcan determinados parámetros para el proceso de análisis (como el número de grupos deseados); los resultados del clustering pueden ser muy sensibles a los parámetros de entrada (Jiawei Han y Micheline Kamber, 2006).

2.2.15. RUIDO

La mayoría de las bases de datos del mundo real contienen valores atípicos, faltantes, desconocidos o erróneos. Algunos algoritmos de clustering son sensibles a estos datos y pueden dan lugar a clústeres de mala calidad (Jiawei Han y Micheline Kamber, 2006).

2.2.16. CLUSTERING O AGRUPACIÓN INCREMENTAL

Algunos algoritmos de clustering no pueden incorporar datos recién ingresados (es decir, actualizaciones a la base de datos) en las estructuras existentes de agrupamiento, en cambio, debe determinar una nueva agrupación a partir de cero. Algunos algoritmos de agrupación son muy sensibles al orden de entrada de los datos; es decir, dado un conjunto de objetos de datos, tal algoritmo puede devolver drásticamente diferentes agrupamientos según el orden de entrada de los objetos. Es importante

desarrollar algoritmos de agrupaciones incrementales e insensibles al orden de entrada (Jiawei Han y Micheline Kamber, 2006).

2.2.17. ALTA DIMENSIONALIDAD

Una base de datos puede contener varias dimensiones o atributos. Muchos algoritmos son buenos en manejo de datos de baja dimensionalidad, lo cual implica solo dos o 3 dimensiones; los ojos humanos son buenos para juzgar la calidad del clustering hasta un máximo de 3 dimensiones. Encontrar clústeres de objetos de datos en un alto espacio dimensional es difícil, sobre todo teniendo en cuenta que los datos pueden ser escasos y muy sesgados (Jiawei Han y Micheline Kamber, 2006).

2.2.18. CLUSTERING BASADO EN RESTRICCIONES

Aplicaciones del mundo real pueden necesitar llevar a cabo el clustering bajo varios tipos de restricciones. Suponiendo, por ejemplo, que se requiere elegir las localizaciones de un número determinado de nuevos cajeros automáticos en una ciudad; para llevar a cabo esto es posible agrupar los hogares mientras se considera restricciones tales como ríos, redes de carretera y tipo y número de clientes por clúster (Jiawei Han y Micheline Kamber, 2006).

2.2.19. INTEPRETABILIDAD Y FACILIDAD DE USO

Los usuarios esperan que los resultados del clustering sean interpretables, comprensibles y utilizables; es decir, el clustering puede

necesitar estar atado a interpretaciones semánticas específicas y aplicaciones (Jiawei Han y Micheline Kamber, 2006).

2.2.20. MEDIDAS DE EFICIENCIA DE ALGORITMOS DE MINERÍA DE DATOS

(1) Sensibilidad = *VP / (VP + VN)*

(2) Especificidad = *VP / (VP + FP)*

(3) Precisión = *(VP + VN) / (VP + FP + FN + VN)*

VP -> *Verdaderos Positivos*

FN -> *Falsos Negativos*

FP -> *Falsos Positivos*

VN -> *Verdaderos Negativos*

2.2.21. ALGORITMO DE HUNT

Cada nodo en el árbol de decisión tiene asociado un subconjunto de los datos de entrenamiento, inicialmente el nodo raíz tiene asociado todo el subconjunto de entrenamiento; a continuación se construye un árbol parcial que tiene tres tipos de nodos: expandidos (interiores), hojas (serán hojas en el árbol final y tendrán asociada una clase) y nodos por expandir (hojas del árbol parcial que deben ser expandidas), finalmente la expansión de un nodo t sería de la siguiente manera: encontrar el mejor Split para t, particionar los datos de t en nodos hijos de acuerdo al Split, etiquetar t y sus nodos hijos con el mejor Split (Carlos Hurtado L., 2016).

2.2.22. ALGORITMO PAM

Algoritmo desarrollado por Kaufman y Rousseeuw. Con el fin de encontrar "k" clústeres, el enfoque que utiliza PAM es determinar un objeto representativo para cada grupo, este objeto representativo, llamado medoide, está destinado a ser el objeto más céntrico de la agrupación. Una vez que los medoides son seleccionados, cada objeto no seleccionado se agrupa con el medoide con el que sea más similar. Todos los valores de disimilitud son dados como entradas para PAM. Para este algoritmo, la calidad de una agrupación se mide por la disimilitud promedio entre cada objeto y el medoide de su clúster. Para encontrar los "k" medoides, PAM comienza con una selección arbitraria de "k" objetos; luego, en cada paso, un intercambio entre un objeto seleccionado "Om" y un objeto no seleccionado "Op" puede ser realizado, siempre que este intercambio de lugar a una mejor calidad de agrupación (Raymond T. Ng y Jiawei Han, 2002).

2.2.23. HASH

Se refiere a una función o método para generar claves o llaves que representen de manera casi inequívoca a un documento, registro, archivo, etc., resumir o identificar un dato a través de la probabilidad utilizando una función o algoritmo hash. Un hash es el resultado de dicha función o algoritmo (Juan, Jorge y Daniel, 2009).

2.2.24. COEFICIENTE GINI

El coeficiente Gini es una medida de la desigualdad ideada por el estadístico italiano Corrado Gini. Normalmente se utiliza para medir la desigualdad en los ingresos dentro de un país, pero puede utilizarse para medir cualquier forma de distribución de desigualdad. El coeficiente Gini es un número entre 0 y 1, donde 0 se corresponde con la perfecta igualdad y 1 se corresponde con la perfecta desigualdad (Wikipedia, 2016).

2.2.25. ÍNDICE GINI

El índice de Gini es el coeficiente de Gini expresado en referencia a 100 como máximo, en vez de como 1, y es igual al coeficiente de Gini multiplicado por 100 (Wikipedia, 2016).

2.2.26. LATENT DIRICHLET ALLOCATION

En estadística, Latent Dirichlet Allocation (LDA) es un modelo generativo que permite que conjuntos de observaciones puedan ser explicados por grupos no observados que explican por qué algunas partes de los datos son similares. Por ejemplo si las observaciones son palabras en documentos, presupone que cada documento es una mezcla de un pequeño número de categorías (también conocidas como tópicos) y la aparición de cada palabra en un documento se debe a una de las categorías a las que el documento pertenece.

2.3. DEFINICIONES DE BASE DE DATOS

2.3.1. SISTEMA DE BASE DE DATOS

Un sistema de base de datos, llamado también un sistema de gestión de base de datos (DataBase Management System "DBMS"), consiste en un conjunto de datos relacionados entre sí, conocido como una base de datos, y un conjunto de programas de software que permitan gestionar y acceder a estos datos. Los programas de software implican mecanismos para la definición de la estructura de las bases de datos; para el almacenamiento de los datos; para el acceso de datos concurrente, compartido o distribuido; y para asegurar la consistencia y seguridad de la información almacenada, a pesar de los fallos del sistema o los intentos de acceso no autorizado (Jiawei Han y Micheline Kamber, 2006).

2.3.2. BASE DE DATOS RELACIONAL

Una base de datos relacional es una colección de tablas, cada una de las cuáles es asignada con un nombre único. Cada tabla consiste en un conjunto de atributos (columnas o campos), y por lo general almacena un gran conjunto de tuplas (registros o filas). Cada tupla en una tabla relacional representa a un objeto identificado por una clave única y es descrita por un conjunto de valores atributo. Un modelo de datos semántico, tal como un modelo entidad – relación (ER), a menudo se construye para las bases de datos relacionales. Un modelo de datos ER representa a la base de datos como un conjunto de entidades y sus relaciones (Jiawei Han y Micheline Kamber, 2006).

2.3.3. ATRIBUTOS CATEGÓRICOS Y NUMÉRICOS

2.3.3.1. Los atributos Categóricos (Cualitativos) representan categorías más que números. Operaciones como suma y resta no tienen sentido. Se dividen a su vez en: Nominales (sin orden significativo) y Ordinales (con orden definido) (Wikipedia, 2016).

2.3.3.2. Los atributos Numéricos (Cuantitativos) son atributos que son números y pueden ser tratados como tal. A su vez se dividen en: Intervalo (no existe cero y la división no tiene sentido) y Radio (el cero existe y la división tiene sentido) (Wikipedia, 2016).

2.3.4. ATRIBUTOS DISCRETOS Y CONTINUOS

2.3.4.1. Un atributo discreto tiene un número finito o contable de valores. En general se representa como números enteros. Atributo binarios son un caso especial de ellos. Atributos categóricos o cualitativos siempre son discretos, los atributos numéricos pueden ser discretos o continuos (Wikipedia, 2016).

2.3.4.2. Un atributo continuo tiene un número infinito de valores posibles. Es representado por número reales o de punto flotante. Se pueden obtener tan precisos como sea el instrumento de medición. Los atributos numéricos pueden ser discretos o continuos (Wikipedia, 2016).

2.3.5. BASE DE DATOS TRANSACCIONAL

En general, una base de datos transaccional consiste en un archivo en el que casa registro representa una transacción. Una transacción implica,

típicamente, un número de identificación único y una lista de artículos o ítems que componen la transacción. Las bases de datos transaccionales pueden tener tablas adicionales asociadas con ella, las cuales contienen otro tipo de información relacionada a la transacción en sí, tales como la fecha de la transacción, el número de identificación del cliente, el número de identificación del vendedor, y así sucesivamente (Jiawei Han y Micheline Kamber, 2006).

2.4. DEFINICIONES DE DATAWAREHOUSE (ALMACÉN DE DATOS)

2.4.1. ALMACÉN DE DATOS (DATA WAREHOUSE)

Un almacén de datos es un repositorio de información recopilada de varias fuentes, se almacena en un esquema unificado, y normalmente reside en un solo sitio. Los almacenes de datos son construidos a través de un proceso de depuración de los datos, integración de los datos, transformación de los datos, carga de los datos, y una actualización periódica de los datos (Jiawei Han y Micheline Kamber, 2006).

Según Chaudhuri y Dayal (1997), un Data Warehouse o almacén de datos es la integración de datos consolidados, almacenados en un dispositivo de memoria no volátil, proveniente de múltiples y posiblemente diferentes fuentes de datos; con el propósito de análisis y, a partir de este, tomar decisiones en función de mejorar la gestión del negocio. Contiene un conjunto de cubos de datos que permiten, a través de las técnicas de OLAP, consolidar, ver y resumir los datos acorde a diferentes dimensiones de estos.

2.4.2. DATA MARTS

Es un subconjunto del Data Warehouse, usado normalmente para el análisis parcial de los datos. El objetivo de subdividir está dado por la complejidad computacional del análisis global de todas las dimensiones del Data Warehouse, y por la necesidad de rapidez (Microsoft Data Warehouse Training, 2000).

2.4.3. OLTP (ONLINE TRANSACTION PROCCESING)

Se les llama así a las aplicaciones orientadas principalmente a la inserción, actualización y eliminación de datos, diseñada casi siempre usando el modelo relacional. Estos sistemas están optimizados para realizar estas operaciones en un periodo corto de tiempo (Microsoft Books Online, 2000).

2.4.4. OLAP (ONLINE ANALITICAL PROCCESING)

Son los sistemas que se utilizan para analizar los datos que las OLTP introducen en la base de datos. A diferencia de los primeros estos casi siempre usan el modelo multidimensional para organizar los datos en la base de datos, ya que bridan mejores resultados a la hora del análisis de estos (Microsoft Books Online, 2000).

CAPÍTULO III: ANÁLISIS DE ALGORITMOS

En este capítulo se verá el análisis de algoritmos de minería de datos de clasificación, clusterización y asociación; el análisis y posterior cuadro comparativo se realizó en base a la eficiencia que tendrían los algoritmos aplicados a datasets o repositorios de datos orientados al ámbito comercial. El análisis de cada grupo de algoritmos esta sub dividido en 3 etapas: descripción de cada algoritmo, análisis de características a través de cuadros comparativos, conclusiones y selección final de los algoritmos que serán probados en la etapa experimental.

3.1. CLASIFICACIÓN

3.1.1. ALGORITMOS

A. J48 – GRAFT TREE

El árbol J48-Graft genera un árbol de decisión injertado, la técnica de injerto es un proceso inductivo que añade nodos a árboles de decisión inferidos con el fin de reducir los errores de predicción. El algoritmo J48-Graft clasifica la región del espacio multidimensional de atributos no ocupados por los ejemplos de entrenamiento, proceso que se muestra con frecuencia para mejorar la exactitud de la predicción; según K. Wisaeng (2013) brindó el siguiente resultado aplicado a un dataset orientado al marketing bancario: alcanzó una sensibilidad del 76,50%, una especificidad del 78,60%, una exactitud del 76,52%, error absoluto medio de 0.32+, error cuadrático medio de 0.42+, y error absoluto relativo de 71.33+.

Según Yoichi Hayashi y Satoshi Nakano (2015), el concepto de árbol injertado se basa en el deseo de descartar el método de "lo simple es lo mejor" para la selección de un buen árbol. En contraste, en el árbol injertado, la atención se centra en el hecho de que los objetos similares tienden a tener la más alta probabilidad de pertenencia a la misma clase; en otras palabras, si el resultado final es un mejor modelo de clasificación, la necesidad de producir árboles más complejos se elimina. El injerto es un post-proceso que se puede aplicar fácilmente a árboles de decisión. Su principal objetivo es la reclasificación de regiones de un espacio de instancia donde no existan datos de entrenamiento o donde solo haya datos mal clasificados. El injerto identifica los cortes más adecuados de las regiones hoja actuales y luego hace el proceso de ramificación con el fin de crear nuevas hojas con clasificaciones las cuales difieren de las originales. En este proceso el árbol se hace más complejo naturalmente; sin embargo, sólo la ramificación que no introduce errores de clasificación en datos que ya hayan sido clasificados es considerada, asegurando que el nuevo árbol reduce errores.

B. LAD TREE

El Árbol LAD (Logical Analysis of Data) es el clasificador para variables destino binarias basado en el aprendizaje de una expresión lógica que puede distinguir entre muestras positivas y negativas de un conjuntos de datos, el concepto general de este algoritmo es el de clasificación, clusterización y otros problemas. La construcción de un modelo LAD para un determinado conjunto de datos típicamente implica la generación de

grandes patrones establecidos y la selección de un subconjunto de ellos que satisfaga la suposición anterior, de tal manera que cada patrón en el modelo satisface ciertos requisitos en términos de prevalencia y homogeneidad; según K. Wisaeng (2013), brindó el siguiente resultado aplicado a un dataset orientado al marketing bancario: alcanzó una sensibilidad del 76,10%, una especificidad del 75,00%, 76,08% de exactitud, error absoluto medio de 0.31+, error cuadrático medio de 0.40+ y error absoluto relativo de 70.08+.

C. ID3

Este es un algoritmo de árbol de decisión introducido en 1986 por Ross Quinlan y basado en el algoritmo de Hunt. ID3 utiliza las medidas de ganancia de información para elegir el atributo de división, sólo acepta atributos categóricos en la construcción de un modelo de árbol. No da un resultado preciso cuando hay ruido. Para construir el árbol de decisión, la ganancia de información es calculada para cada atributo y se selecciona el atributo con la ganancia de información más alta para designarlo como un nodo raíz; se etiqueta al atributo como un nodo raíz y los posibles valores del atributo se representan como arcos, a continuación todos los resultados se ponen a prueba para comprobar si están sometidas a la misma clase o no, si todos los casos están cayendo bajo la misma clase, el nodo se representa con el nombre de clase única, de lo contrario se selecciona el atributo de división para clasificar los casos, este algoritmo no es compatible con la poda; según Surjeet Kumar Yadav y Saurabh Pal (2012), brindó el siguiente resultado aplicado a un dataset real orientado al campo educacional

universitario: en precisión alcanzó un 62,22% para instancias correctamente clasificadas y 26,67% para instancias incorrectamente clasificadas, su tiempo de ejecución fue de 0.00 segundos.

D. C4.5 o J48 TREE

Algoritmo sucesor de ID3 desarrollado por Ross Quinlan, también basado en el algoritmo de Hunt, el cual maneja atributos categóricos y continuos para construir un árbol de decisión. Con el fin de manejar los atributos continuos C4.5 divide los valores del atributo en dos particiones basadas en un umbral seleccionado, de tal manera que todos los valores por encima del umbral sean un hijo, y el restante como otro hijo. Este algoritmo también se ocupa de los valores de atributos que faltan. C4.5 usa una relación de ganancia como medida de selección de características para construir el árbol de decisión y utiliza la poda pesimista para eliminar ramas innecesarias con el fin de mejorar la precisión de la clasificación; según Surjeet Kumar Yadav y Saurabh Pal (2012), brindó el siguiente resultado aplicado a un dataset real orientado al campo educacional universitario: en precisión alcanzó un 67,78% para instancias correctamente clasificadas y 32,22% para instancias incorrectamente clasificadas, su tiempo de ejecución fue de 0.03 segundos.

Md. Rafiqul Islam y Md. Ahsan Habib (2015) pusieron en práctica el procedimiento de minería de datos para un análisis prospectivo de sectores de actividad en la banca minorista. Haciendo comparación con el algoritmo J48 del software WEKA se encontró que el modelo de predicción de

minería de datos para los sectores comerciales potenciales para el desembolso de préstamos es muy preciso, por ende la campaña orientada a los objetivos de la empresa en cuestión resultaría a futuro muy eficaz.

E. CART

Algoritmo CART (Classification And Regression Trees) es un algoritmo introducido por Breiman y basado en el algoritmo de Hunt. CART maneja ambos atributos, categóricos y continuos, para construir un árbol de decisión. Este algoritmo se ocupa de los valores faltantes y utiliza el índice Gini como medida para la selección de atributos para construir el árbol de decisión. A diferencia de los algoritmos ID3 y C4.5, CART produce divisiones binarias, por tanto árboles binarios. El índice Gini no utiliza supuestos probabilísticos como ID3 y C4.5. CART también utiliza la poda en base al costo de complejidad para eliminar ramas no fiables del árbol con el fin de mejorar la precisión; según Surjeet Kumar Yadav y Saurabh Pal (2012), brindó el siguiente resultado aplicado a un dataset real orientado al campo educacional universitario: en precisión alcanzó un 62,22% para instancias correctamente clasificadas y 37,78% para instancias incorrectamente clasificadas, su tiempo de ejecución fue de 0.09 segundos.

3.1.2. CUADRO COMPARATIVO

Algoritmo	**Características**					
	Técnica utilizada	***¿Utilización de un conjunto de datos de entrenamiento?***	***Tipos de atributos que acepta***	***¿Trabaja con ruido (Noise)?***	***¿Trabaja con valores faltantes (Missing Values)?***	***Uso***
C4.5 o J48 Tree	**Ganancia de Información**, y para trabajar atributos continuos el uso de un **umbral seleccionado** para realizar las particiones.	Si	Categóricos y Numéricos Continuos y Discretos Binarios	Si	Si	Tipo de dataset: ***Campo Educacional Universitatio*** Resultados de Precisión de Instancias: - Correcta clasificación: 67,78% - Incorrecta claficiación: 32,22% Tiempo de ejecución: 0.03 seg. Algoritmo probado como eficiente para sectores comerciales.
LAD Tree	**Clasificador de variables destino binarias**: Aprendizaje de una expresión lógica capaz de distinguir muestras positivas y negativas.	No	Categóricos yNuméricos Continuos y Discretos Binarios	Si	Si	Tipo de dataset: ***Marketing Bancario*** Resultados: - Sensibilidad: 76,10% - Especificidad: 75,00% - Precisión: 76,08%
J48-Graft Tree	**Injerto**: Adición de nodos a árboles de decisión inferidos.	Si	Categóricos y Numéricos Binarios	Si	Si	Tipo de dataset: ***Marketing Bancario*** Resultados: - Sensibilidad: 76,50% - Especificidad: 78,60% - Precisión: 76,52%
CART (Clasification And Regression Trees)	**Índice Gini**: El algoritmo hace uso del índice Gini como de selección de los atributos. El algoritmo genera árboles binarios.	Si	Categóricos yNuméricos Continuos y Discretos	Si	Si	Tipo de dataset: ***Campo Educacional Universitatio*** Resultados de Precisión de Instancias: - Correcta clasificación: 62,22% - Incorrecta claficiación: 37,78% Tiempo de ejecución: 0.09 seg.
ID3	**Ganancia de Información:** Calculada para cada atributo, siendo seleccionado aquel con la ganancia más alta como nodo raíz y sus posibles valores como arcos.	Si	Categóricos	No	Si	Tipo de dataset: ***Campo Educacional Universitatio*** Resultados de Precisión de Instancias: - Correcta clasificación: 62,22% - Incorrecta claficiación: 26,67% Tiempo de ejecución: 0.00 seg.

Tabla 1: Cuadro comparativo de algoritmos de clasificación.
Fuente propia.

3.1.3. COMPARACIÓN CON RESULTADOS OBTENIDOS EN OTROS ENTORNOS

En este caso la comparación se realizará con una encuesta (survey) que analiza algoritmos de clasificación aplicados a los campos de detección de intrusiones a la red, filtrado de spam e inteligencia artificial. En este caso Sagar S. Nika (2015), autor del paper, obtuvo los siguientes resultados:

Algoritmo	Ventajas	Limitaciones
C4.5	- Construye modelos que son fáciles de interpretar. - Es fácil de implementar. - Puede utilizar tanto valores discretos como continuos. - Es capáz de lidiar con el ruido.	- Pequeñas variaciones realizadas a los datos pueden conducir a diferentes árboles de decisión. - No funciona muy bien cuando el conjunto de datos de entreamiento es pequeño.
ID3	- Produce un resultado más preciso que el algoritmo C4.5. - Su tasa de detección es mayor y el consumo de espacio es menor.	- Requiere un largo tiempo de búsqueda. - A veces puede generar reglas muy largas que son muy difíciles de podar.
K-Nearestneighbor	- Las clases no necesitan ser linealmente separables. - A veces es robusto con respecto a los datos de entrenamiento ruidosos. - Muy adecuado para las clases multimodales.	- El tiempo para encontrar los vecinos cercanos en un gran conjunto de datos puede ser excesivo. - Es sensible a atributos ruidosos o irrelevantes. - El rendimiento del algoritmo depende del número de dimensiones utilizadas.
Naive Bayes	- Es fácil de implementar. - Gran eficiencia computacional y tasa de clasificación. - Predice resultados precisos para la mayoría de los problemas de clasificación y predicción.	- La precisión del algoritmo disminuye si la cantidad de datos es menor. - Para obtener buenos resultados se requiere un número muy grande de registros.
Support vector-machine	- Tiene gran precisión. - Funciona bien si los datos no son linealmente separables.	- El requisito de velocidad y tamaño tanto en el entrenamiento como en las pruebas es mayor. - Requisitos de alta complejidad y memoria extensa en muchos de los casos.
Artificial Neural Network Algorithm	- Es fácil de usar, con pocos parámetros para ajustar. - Una red neuronal aprende y la reprogramación no es necesaria. - Es fácil de implementar. - Es aplicable a una amplia gama de problemas.	- Requiere un elevado tiempo de procesamiento si la red neuronal es grande. - Es complicado saber cuántas neuronas y capas son necesarias. - El aprendizaje puede ser lento.

Tabla 2: Cuadro comparativo de algoritmos de clasificación aplicados a otro entorno. Fuente: Sagar S. Nika.

En su trabajo y a través de su cuadro comparativo podemos apreciar que considera que los mejores algoritmos para tratar datos pertenecientes a los campos antes mencionados serían los árboles de decisión, específicamente el C4.5 e ID3, redes bayesianas y los que operan a través de una red neuronal artificial. Podemos decir que dado un tipo de información en específico, la calidad de los modelos de clasificación generados puede variar para bien o para mal.

3.1.4. CONCLUSIONES

De los algoritmos analizados, los algoritmos C4.5 Tree (J48) y J48-Graft Tree muestran ser los más eficientes para la clasificación de datasets orientados al ámbito comercial online dadas las siguientes razones:

A. El algoritmo C4.5 o J48 Tree trabaja con distintos tipos de atributos, categóricos, numéricos, continuos, etc., característica de vital importancia, ya que datasets orientados a este entorno trabajan con distintos tipos de variables para manejar valores como precios, localizaciones, códigos de trabajadores o clientes, etc.; también es capaz de lidiar con problemas típicos en este tipo de datasets como el ruido (Noise) y valores faltantes (Missing Values). El algoritmo también ha mostrado resultados eficientes en datasets orientados al campo educacional, demostrando que es capaz de clasificar correctamente datos asociados a entidades “persona” y atributos relacionados a ella; así mismo se hace referencia de que este algoritmo trabaja correctamente aplicado en datasets del sector

comercial, por ejemplo en el análisis de la banca minorista o de sectores comerciales potenciales.

B. El algoritmo J48-Graft Tree también es capaz de trabajar con distintos tipos de atributos y lidiar con los problemas típicos de ruido (Noise) y valores faltantes (Missing Values). Además, como su nombre lo indica, este algoritmo genera un árbol de decisión injertado el cual es utilizado en esencia para reducir errores de predicción al momento de generar un modelo. Lo que más llama la atención de este algoritmo es el método "lo simple es lo mejor" utilizado para la selección de un buen árbol, centrándose en el hecho de que los objetos similares tienden a tener más alta probabilidad de pertenencia a la misma clase. Dada la naturaleza del dataset a ser analizado, orientado al ámbito comercial, es vital que los modelos sean de fácil entendimiento o "simples" y que presenten la menor cantidad de errores posibles, ya que estos serán utilizados, muy probablemente, para la toma de decisiones que afecten el rumbo que podría tomar una empresa u organización.

3.2. CLUSTERIZACIÓN

3.2.1. ALGORITMOS

A. K – MEANS

Clustering mediante K-Means es un método de análisis que tiene como objetivo la partición de un conjunto de n observaciones en k agrupaciones, en la que cada observación pertenece a la agrupación con la media más cercana. El algoritmo se llama K-Means, donde k es el número de grupos que se desea, y cada caso es asignado a la agrupación o clúster cuya distancia a su media o centro sea más cercana. La acción del algoritmo se centra en la búsqueda de los centros o medias k, empezando así con un conjunto inicial de medios y la clasificación de los casos en función a las distancias a estos medios; a continuación se calcula nuevamente los centros de las agrupaciones o clúster usando los casos asignados a cada agrupación, y se vuelven a clasificar todos los casos basados en el nuevo conjunto de medios; esto se repetirá hasta que le medios asignados no varíen mucho. Finalmente se calculan los medios finales y se asignan los casos a sus agrupaciones permanentes (Manish Verma, Mauly Srivastava, Neha Chack y Atul Kumar Diswar y Nidhi Gupta, 2012); según su estudio este algoritmo aplicado a un conjunto de datos bancarios relacionados a la información del cliente, el cual se compone de 11 atributos y 600 entradas, brindó el siguiente resultado: formó 2 clústeres, el primero agrupó 254 casos (42%) y el segundo 346 casos (58%), realizó 4 iteraciones, el tiempo que tomó construir el modelo fue de 0.08 segundos y no hubieron instancias que no pertenecieran a ningún clúster.

Según Jiawei Han y Micheline Kamber (2006), existe un buen número de variantes del método K-Means; estos pueden diferir en la selección de los medios iniciales "k", el cálculo de la disimilitud y las estrategias para el cálculo de los centros de los clústeres. Una de las variantes del K-Means es el método K-Modes, el cual extiende el paradigma del algoritmo K-Means con el fin de agrupar datos categóricos mediante las sustitución de las medias de los clústeres por modos (modes), usando una nueva medida de disimilitud para tratar con los objetos categóricos y un método basado en la frecuencia para actualizar los modos de los clústeres. Los métodos K-Means y K-Modes se pueden integrar para agrupar los datos que tengan valores tanto categóricos como numéricos.

B. K-MEDOIDS

El algoritmo K-Means es sensible a valores atípicos, ya que un objeto cuyo valor sea extremadamente grande puede distorsionar sustancialmente la distribución de los datos. El método K-Medoids, en lugar de tomar el valor medio de los objetos en un clúster como un punto de referencia, recoge objetos reales para representar los clústeres, usando un objeto representativo para cada grupo; cada objeto restante se agrupa con el objeto representativo con el cual tenga mayor similitud. El método de partición se realiza a continuación basado en el principio de minimizar la suma de las disimilitudes entre cada objeto y su correspondiente objeto representativo. En general, el algoritmo itera hasta que, eventualmente, cada objeto representativo es en realidad el medoide (medoid), u objeto más céntrico, de su clúster (Jiawei Han y Micheline Kamber, 2006). Según su libro,

viendo más de cerca el algoritmo, los objetos representativos iniciales son elegidos de forma aleatoria. El proceso iterativo de la sustitución de objetos representativos por objetos no representativos continúa mientras se mejora la calidad de la agrupación resultante. Esta calidad se calcula utilizando la función de coste que mide la disimilitud promedio entre el objeto y el objeto representativo de su clúster.

C. CLARA y CLARANS

La técnica de particionamiento del algoritmo K-Medoids funciona de manera eficiente para pequeños conjuntos de datos, pero no escala bien para grandes conjuntos de datos. Para hacer frente a grandes conjuntos de datos se puede utilizar un método basado en el muestreo llamado CLARA (Clustering LARge Applications). La idea detrás de CLARA es la siguiente: en lugar de tomar todo el conjunto de datos en consideración, una pequeña parte de los datos reales se elige como representante de los datos. Los medoides son entonces seleccionados de esta muestra utilizando el algoritmo PAM (Partitioning Around Medoids). Los objetos representativos (medoides) elegidos serán, probablemente, similares a los que han sido elegidos de todo el conjunto de datos. CLARA extrae múltiples muestras del conjunto de datos, se aplica PAM en cada muestra y devuelve su mejor agrupación como salida. La eficacia de CLARA depende del tamaño de la muestra, nótese que PAM hace una búsqueda de los mejores medoides de un conjunto de datos dado, mientras que CLARA busca los mejores medoides en una muestra seleccionada del conjunto de datos. CLARA no puede encontrar el mejor clustering o agrupación si

alguno de los mejores medoides de la muestra no es uno de los mejores medoides del conjunto de datos completos. Un buen agrupamiento basado en el muestro no representa necesariamente una buena agrupación de todo el conjunto de datos si es que la muestra está sesgada (Jiawei Han y Micheline Kamber, 2006).

Jiawei Han y Micheline Kamber (2006) preguntan en su libro: ¿Cómo se puede mejorar la calidad y escalabilidad de CLARA?; un algoritmo de tipo K-Medoids llamado CLARANS (Clustering Large Applications based upon RANdomized Search) fue propuesto, el cual combina la técnica de muestreo de PAM; sin embargo, a diferencia de CLARA, CLARANS no se limita a cualquier muestra en cualquier momento dado; mientras que CLARA tiene una muestra fija en cada etapa de la búsqueda, CLARANS cuenta con una muestra con cierta aleatoriedad en cada etapa de la búsqueda. Conceptualmente el proceso de agrupamiento se puede ver como una búsqueda a través de un gráfico, donde cada nodo es una solución potencial (un conjunto de medoides). Dos nodos son vecinos (conectados por un arco en el gráfico) si sus conjuntos difieren solo en un objeto. A cada nodo se le puede asignar un coste el cual se define por la disimilitud total entre cada objeto y el medoide de su clúster. En cada paso, PAM examina todos los vecinos del nodo actual en su búsqueda de una solución con costo mínimo; el nodo actual es reemplazado por el vecino con el mayor descenso en costes. Debido a que CLARA trabaja sobre una muestra de todo el conjunto de datos, se examina un menor número de vecinos y se restringe la búsqueda a subgrafos que son más pequeños que el gráfico original.

Mientras que CLARA extrae una muestra de nodos al principio de la búsqueda, CLARANS dibuja dinámicamente una muestra aleatoria de vecinos en cada paso de la búsqueda, el número de vecinos a ser incluidos está restringido por un parámetro especificado por el usuario. De esta manera, CLARANS no se limita a la búsqueda de un área localizada; si se encuentra un mejor vecino (es decir tiene un error inferior), CLARANS se mueve al nodo vecino y el proceso comienza de nuevo, de otro modo, la agrupación o clúster actual produce un mínimo local. Si se encuentra un mínimo local, CLARANS comienza con nuevos nodos seleccionados al azar en búsqueda de un nuevo mínimo local; una vez se ha encontrado un número especificado por el usuario de mínimos locales, la salida del algoritmo, como una solución, será el mejor mínimo local, es decir aquel con el costo más bajo.

CLARANS ha demostrado ser experimentalmente más eficaz que PAM y CLARA. Este algoritmo puede ser utilizado para encontrar el número más "natural" de clústeres utilizando un coeficiente silueta (una propiedad los objetos de datos que especifica que tanto un objeto pertenece realmente a su clúster). CLARANS también permite la detección de valores atípicos, sin embargo su complejidad computacional es de O (n^2), donde n representa el número de objetos (Jiawei Han y Micheline Kamber, 2006).

D. HIERARCHICAL CLUSTERING

Hierarchical Clustering o agrupación jerárquica, como su nombre lo indica, construye una jerarquía de clústeres, o en otras palabras, un árbol de

clústeres conocido también como dendograma. Cada nodo clúster contiene hijos clúster, clústeres hermanos particionan los puntos que sean de su padre en común. El proceso básico de la agrupación jerárquica, definido por SC Johnson en 1967, es el siguiente: se comienza con la asignación de cada elemento a un clúster, por lo que si tenemos N elementos ahora tendremos también N clústeres con un elemento cada uno, encontrar los pares más cercanos entre los clústeres y fusionarlos en un solo grupo, teniendo así ahora menos clúster, calcular las distancias entre los nuevos clúster y los antiguos, finalmente repetir los pasos anteriores hasta que todos los elementos se agrupen en el número K de clúster deseado (Manish Verma, Mauly Srivastava, Neha Chack y Atul Kumar Diswar, Nidhi Gupta, 2012), según su estudio este algoritmo aplicado a un conjunto de datos bancarios relacionados a la información del cliente, el cual se compone de 11 atributos y 600 entradas, brindó el siguiente resultado: formó 2 clústeres, el primero agrupó 599 casos (100%), el segundo 1 caso (0%), el tiempo que tomó construir el modelo fue de 1.16 segundos y no hubieron instancian que no pertenecieran a ningún clúster.

Según Jiawei Han y Micheline Kamber (2006), el método de agrupación jerárquica funciona mediante la agrupación de objetos de datos en un árbol de clústeres. Los métodos de agrupación jerárquica pueden clasificarse como de aglomeración o de división, dependiendo de si la descomposición jerárquica está formada de abajo hacia arriba (merging) o de arriba hacia abajo (splitting). La calidad de un método de agrupación jerárquico puro sufre de su incapacidad para realizar ajustes una vez que una decisión de fusión (merge) o división (split) ha sido ejecutada; es decir, si una decisión

de fusión o decisión más tarde resulta haber sido una mala elección, el método no puede hacer "backtrack" o marcha atrás y corregirlo. En su libro Jiawei Han y Micheline Kamber especifican que, en general, hay 2 métodos de agrupación jerárquica:

a) Agrupación Jerárquica Aglomerativa

Esta estrategia de abajo hacia arriba (bottom-up) se inicia mediante la colocación de cada objeto en su propio clúster, y luego se fusionan estos clústeres atómicos en clústeres cada vez más grandes, hasta que todos los objetos estén en un solo grupo o clúster, o hasta que ciertas condiciones de terminación hayan sido satisfechas. La mayoría de métodos de agrupamiento jerárquico pertenecen a esta categoría, difiriendo solo en su definición de similitud inter clúster.

b) Agrupación Jerárquica Divisiva

Esta estrategia de arriba hacia abajo (top-down) hace el proceso contrario a la agrupación jerárquica aglomerativa; ésta subdivide el clúster en clústeres cada vez más pequeños, hasta que cada objeto forme un grupo por sí mismo o hasta que se satisfaga ciertas condiciones de terminación, tales como el número deseado de clústeres o que el diámetro de cada clúster se encuentre dentro de un umbral especificado.

E. BIRCH

BIRCH fue diseñado para agrupar una gran cantidad de datos numéricos mediante la integración de la agrupación jerárquica (en la etapa inicial denominada microclustering) y otros métodos de agrupamiento como el particionamiento iterativo (en una etapa posterior denominada macroclustering). Supera las dos dificultades de los métodos aglomerativos de clusetring: la escalabilidad y la incapacidad para deshacer los que se hizo en un paso anterior. BIRCH introduce dos conceptos "características del clustering o agrupación" y "árbol de características del clustering o agrupación", los cuales son utilizados para resumir las representaciones de los clústeres. Estas estructuras ayudan al método de clusterización a lograr una buena velocidad y escalabilidad en grandes bases de datos y también hacen que sea eficaz para el clustering incremental y dinámico de objetos de datos entrantes. El concepto "características del clustering o agrupación" (Clustering Feature "CF") es esencialmente un resumen de las estadísticas de un clúster determinado: el cero, primer y segundo momento del clúster vistos desde un punto estadístico; las características del clustering son aditivas. Las características del clustering son suficientes para el cálculo de todas las medidas que se necesitan para la toma de decisiones para el agrupamiento en BIRCH; así este algoritmo utiliza el almacenamiento de manera eficiente mediante el empleo de las características del clustering para resumir la información acerca de los clúster de objetos, evitando así la necesidad de almacenar todos los objetos. El segundo concepto "árbol de características del clustering o agrupación" (Clustering Feature Tree "CF tree") es un árbol de altura balanceada que almacena las características del

clustering para una agrupación jerárquica. Por definición un nodo hoja en un árbol tiene descendientes, los nodos no hoja almacenan las sumas de los CFs de sus hijos, y por lo tanto un resumen de la información del clustering de sus descendientes. Un árbol FC tiene dos parámetros, el factor de ramificación "B" y el umbral "T". El factor de ramificación especifica el número máximo de descendientes por nodo no hoja, y el parámetro de umbral especifica el diámetro máximo de sub clústeres almacenados en los nodos hoja del árbol. BIRCH trata de producir los mejores clústeres con los recursos disponibles; dada una cantidad limitada de memoria principal, es importante reducir al mínimo el tiempo requerido para las entradas y salidas "I/O" (Jiawei Han y Micheline Kamber, 2006).

F. CURE

CURE (Clustering Using REpresentatives) es un algoritmo de agrupamiento de datos eficiente para grandes bases de datos, el cual es más robusto para el trato de valores atípicos e identifica clústeres con forma no esféricas y de tamaños variados. Para evitar problemas con clústeres de tamaño y forma no uniforme, CURE emplea un algoritmo de agrupamiento jerárquico que adopta una posición intermedia entre el centroide y todos los puntos extremos. En CURE, un número constante "c" de puntos bien dispersos de un clúster son elegidos y contraídos hacia el centroide del clúster por una fracción "α". Los puntos dispersos, luego de la contracción, son utilizados como representantes del clúster. Las agrupaciones con el par más cercano de representantes serán los clústeres que se fusionaran en cada paso del algoritmo CURE. Esto permite a CURE identificar correctamente

los clústeres, y hace que sea menos sensible a valores atípicos (Wikipedia, 2016).

G. ROCK

ROCK (RObust Clustering using linKs) es un algoritmo de agrupamiento jerárquico que explora el concepto de links o enlaces (el número de vecinos comunes entre dos objetos) para datos con atributos categóricos. Para agrupar datos con atributos booleanos y categóricos, algoritmos de clustering o agrupación tradicionales utilizan funciones de distancia; sin embargo, experimentos muestran que este tipo de medidas de distancia no dan lugar a agrupaciones de alta calidad cuando realizan agrupamiento de data categórica. Por otra parte, la mayoría de algoritmos de agrupamiento o clustering examinan únicamente la similitud entre dos puntos cuando realiza el proceso de clusterización; es decir, en cada paso los puntos que son más similares se fusionan en un solo grupo; este enfoque localizado es propenso a errores, por ejemplo, dos clústeres distintos pueden tener algunos puntos o valores atípicos que estén cerca; por lo tanto, el basarse en la similitud entre dos puntos para tomar decisiones de clustering puede causar que dos clústeres diferentes sean fusionados. ROCK adopta un enfoque más global de agrupación considerando los barrios (neighborhoods) de distintos pares de puntos. Si dos puntos similares también tienen barrios similares, es probable que ambos pertenezcan al mismo grupo, por tanto se pueden fusionar. ROCK primero construye un gráfico escaso a partir de una matriz de similitud de datos dada utilizando un umbral de similitud y el concepto de vecinos compartidos, a

continuación realiza la agrupación jerárquica de aglomeración en el gráfico escaso, una medida de calidad (goodness) es utilizada para evaluar el clustering. El muestreo aleatorio es utilizado para escalar hasta grandes conjuntos de datos. En varios conjuntos de datos de la vida real, tales como conjuntos de datos relacionados al ámbito político (como votaciones al congreso) o data sets diseñados para la experimentación establecidos por UC-Irvine Machine Learning Repository, ROCK ha demostrado que cuenta con la capacidad de derivar clústeres más significativos que otros algoritmos tradicionales de agrupamiento jerárquico (Jiawei Han y Micheline Kamber, 2006).

H. CHAMELEON

Chameleon es un algoritmo de agrupamiento jerárquico que utiliza el modelado dinámico para determinar la similitud entre pares de clústeres, fue derivado basado en las debilidades observadas en dos algoritmos de agrupamiento jerárquico: ROCK y CURE. ROCK y esquemas relacionados hacen hincapié en la interconexión de los clústeres sin tener en cuenta información relativa a la proximidad de los clústeres. CURE y esquemas relacionados consideran la proximidad de los clústeres, ignorando la interconexión de los clústeres. En Chameleon, la similitud de los clústeres se evalúa basada en qué tan bien conectados están los objetos dentro de un clúster y la proximidad de los clústeres. Es decir, dos clústeres son fusionados si su interconectividad es alta y están muy juntos; por lo tanto, Chameleon no depende de un modelo estático proporcionado por el usuario, y puede adaptarse automáticamente a las características internas de los

clústeres que se pueden fusionar. El proceso de fusión facilita el descubrimiento de clústeres naturales y homogéneos, y es aplicable a todo tipo de datos siempre que una función de similitud sea especificada. Chameleon utiliza un enfoque de gráfico de vecino "k" más cercano para construir un gráfico escaso, donde cada vértice del gráfico represente un objeto de datos, y donde exista una arista entre dos vértices (objetos) si uno de los objetos se encuentra entre los objetos "k" más similares del otro objeto. Las aristas son ponderadas para reflejar la similitud entre los objetos. Chameleon utiliza un algoritmo de particionamiento para dividir el gráfico de vecino "k" más cercano en un gran número de, relativamente, pequeños sub-clústeres. A continuación, se utiliza un algoritmo de agrupamiento jerárquico aglomerativo que, repetidamente, fusiona estos sub-clústeres en función de su similitud. Para determinar los pares de sub-clústeres más similares, el algoritmo tiene en cuenta tanto la interconexión, así como la cercanía de los clústeres (Jiawei Han y Micheline Kamber, 2006).

I. DBSCAN

Este algoritmo encuentra todos los clústeres correctamente, independientemente del tamaño, la forma y ubicación de los clústeres respecto a otros. DBSCAN se basa en dos conceptos principales: accesibilidad y conectividad de la densidad, estos dos conceptos dependen de dos parámetros de entrada del algoritmo: el tamaño de la variable épsilon, que vendría a ser el radio de alcance, y los puntos mínimos asociados a un clúster "m" (Manish Verma, Mauly Srivastava, Neha Chack

y Atul Kumar Diswar, Nidhi Gupta, 2012). Para encontrar un clúster, DBSCAN comienza con un punto arbitrario P y recupera la densidad alcanzable a partir de ese punto. Si P es un punto central, el algoritmo produce un clúster. Si P es un punto fronterizo y no existen puntos con densidad accesible o alcanzable DBSCAN visita el siguiente punto en la base de datos. Debido a que utiliza los valores globales EPS (radio épsilon) y MinPts (Mínimo de puntos en un clúster), el algoritmo puede fusionar dos clústeres si estos son muy cercanos uno del otro. Dos clústeres que tengan al menos la densidad del clúster más delgado se pueden separar una de la otra solo si la distancia entre ambos clústeres en mayor al de la variable EPS, por ende una llamada recursiva a DBSCAN puede ser necesaria para detectar clústeres con una densidad mayor a la determinada por MinPts (Martin Ester, Hans-Peter Kriegel, Jiirg Sander y Xiaowei Xu, 1996), según su estudio este algoritmo aplicado a un conjunto de datos bancarios relacionados a la información del cliente, el cual se compone de 11 atributos y 600 entradas, brindó el siguiente resultado: formó 3 clústeres, el primero agrupó 10 casos (40%), el segundo agrupo 6 casos (24%) y el tercero 9 casos (36%), realizó 3 iteraciones, el tiempo que tomo construir el modelo fue de 1.03 segundos y hubieron 575 instancias que no pertenecieron a ningún clúster.

J. EM ALGORITHM

Es un método iterativo cuyo propósito es encontrar una estimación de máxima verosimilitud para un parámetro ɵ de una distribución, donde el modelo depende de las variables latentes no observadas (Manish Verma,

Mauly Srivastava, Neha Chack y Atul Kumar Diswar, Nidhi Gupta, 2012). Para ejecutar este algoritmo se necesita: los datos observados "y", una densidad paramétrica que describa los datos observados "p(y|ɵ)", una descripción de los datos completos "x", una densidad paramétrica de los datos completos "p(x|ɵ)", considerando que el soporte de "x" no depende de ɵ. El algoritmo EM hace una suposición acerca de "x" (los datos completos), luego encuentra el ɵ que maximiza el valor esperado de la verosimilitud logarítmica de "x", una vez que se tiene el nuevo ɵ puede elegir una mejor elección respecto a "x" e itera (José Antonio Camarena Ibarrola, 2016), según el estudio de Manish Verma, Mauly Srivastava, Neha Chack y Atul Kumar Diswar, Nidhi Gupta este algoritmo aplicado a un conjunto de datos bancarios relacionados a la información del cliente, el cual se compone de 11 atributos y 600 entradas, brindó el siguiente resultado: formó 6 clústeres, el primero agrupó 31 casos (5%), el segundo agrupó 97 casos (16%), el tercero agrupó 65 casos (11%), el cuarto agrupó 184 casos (31%), el quinto agrupó 92 casos (15%) y el sexto agrupó 131 casos (22%), el tiempo que demoró construir el modelo fue de 76.84 segundos y no hubieron instancias que no pertenecieran a ningún clúster.

Según Jiawei Han y Micheline Kamber (2006), el algoritmo EM (Expectación – Maximización) es un algoritmo de refinamiento iterativo que puede ser utilizado para la búsqueda de parámetros estimados. Puede ser visto como una extensión del paradigma que propone K-Means, el cual asigna un objeto al clúster con el cual tenga una similitud mayor, basado en la media o centroide del clúster. En lugar de asignar cada objeto en una agrupación especializada, EM asigna cada objeto a un clúster de acuerdo a

un peso que representa la probabilidad de pertenencia, en otras palabras no hay límites estrictos entre los clústeres; por lo tanto los nuevos medios o centroides se calculan basados en las medidas ponderadas.

3.2.2. CUADRO COMPARATIVO

Algoritmo	**Características**				
	Técnica Utilizada	***Tipos de atributos que acepta***	***Ventajas***	***Desventajas***	***Uso***
EM Algorithm	Es un método iterativo cuyo propósito es encontrar una estimación de máxima verosimilitud para un parámetro ɵ de una distribución, donde el modelo depende de las variables no observadas. Puede ser visto como una extensión del paradigma K-Means, solo que en vez de asignar cada objeto a una agrupación especializada, EM asigna cada objeto a un clúster de acuerdo a un peso que representa la probabilidad de pertenenecia.	Categóricos y Numéricos Continuos y Discretos	Simplicidad conceptual. Facilidad de aplicación. La tasa de convergencia de las primera etapas es normalmente buena.	Puede llegar a ser muy lento cuando se llega a un óptimo local. Una alta dimensionalidad puede realentizar dráticamente el algoritmo. Puede requerir muchas iteraciones. Puede no funcionar bien si la falta de información es grande.	Dataset: ***Conjunto de datos bancarios*** - 11 atributos - 600 entradas Resultados: - Clústeres: 6 de 31 casos (5%), 97 casos (16%), 65 casos (11%), 184 casos (31%), 92 casos (15%) y 131 casos (22%). - Tiempo de ejecución: 76.84 seg. - Casos no clasificados: 0.
K-Means	Partición de un conjunto de "n" observaciones en "k" agrupaciones, contando cada una con una media representativa del clúster. El algoritmo se centra en la búsqueda recursiva de nuevas medias y la clasificación de las observaciones en base a distancias respecto a estas.	Categóricos (Mediante la integración con su variación K-Modes) Numéricos Continuos y Discretos	Es computacionalmente más rápido debido a que la data es fraccionada en "k" agrupaciones. Produce clústeres más estrictos que otros algoritmos, especialmente si estos tiene forma globular.	Es sensible a valores atípicos. Dificultad para predecir el valor de "k". No funciona de manera eficiente con un cluster global. Diferentes particiones iniciales pueden resultar en diferentes clústeres finales.	Dataset: ***Conjunto de datos bancarios*** - 11 atributos - 600 entradas Resultados: - Clústeres: 2 de 254 casos (52%) y 346 casos (58%). - Iteraciones: 4 - Tiempo de ejecución: 0.08 seg. - Casos no clasificados: 0
Hierarchical Clustering	**Jerarquía de clústeres**: Construye un dendograma (árbol de clústeres). Asignación de cada elemento a un clúster individual, se encuentran los pares de clústeres más cercanos y se fusionan en un solo grupo, se calcula la distancia entre los nuevos y antiguos clústeres, y se repiten los pasos hasta agruparlos en el número "K" de clústeres deseados. Cuenta con dos métodos: - A. J. Aglomerativa: Cada objeto inicia en su propio clúster, estos se fusionan hasta quedar solo un clúster o se satisfagan ciertas condiciones. - A. J. Divisiva: Se divide el clúster original en subclústeres más pequeños hasta que satisfaga ciertas condiciones de terminación. (Este suele ser menos utilizado)	Categóricos y Numéricos Continuos y Discretos	Produce un ordenamiento de los objetos, lo cual puede ser útil para la visualización de los datos. Clústeres más pequeños son generados, lo cual puede ser útil para el proceso de descubrimiento.	Incapacidad de realizar ajustes una vez que una decisión, fusión o división, ha sido tomada. El uso de diferentes medidas de distancia entre clústeres pueden generar diferentes resultados.	Dataset: ***Conjunto de datos bancarios*** - 11 atributos - 600 entradas Resultados: - Clústeres: 2 de 599 casos (100%) y 1 casos (0%). - Tiempo de ejecución: 1.16 seg. - Casos no clasificados: 0

BIRCH (Algomerativo)	Integración de la argupación jerárquica (en la etapa inicial "microclustering") y otros métodos de agrupamiento como el particionamiento iterativo (en la etapa posterior "macroclustering"). Uso de los conceptos "características del clustering (CC)" y "árbol de CC", utilizados para resumir las representaciones de los clúster, ayudando al algoritmo a lograr buena velocidad y escalar a grandes bases de datos.	Numéricos	Escala a grandes conjuntos de datos. Es capaz de deshacer una acción realizada en una etapa anterior. Buena velocidad. Capaz de realizar clustering incremental (dinámico a objetos entrantes). Es capaz de trabajar con una cantidad limitada de la memoria. Su complejidad computacional es de O(n).	Puesto que cada nodo en un árbol CC puede contener sólo un número limitado de entradas, debido al tamaño, el nodo de un árbol CC no siempre corresponde con lo que el usuario puede considerar un clúster natural. Si los clústeres no son de forma esférica no funciona bien, ya que utiliza la noción de radio o el diámetro para controlar el límite del clúster. Solo trabaja con datos numéricos. Es sensible al orden del registro de datos.	Ha funcionado de manera eficiente en el filtrado de imágenes, así como en el análsis de data en base a su densidad.
CURE (Aglomerativo)	Emplea un algoritmo de agrupamiento jerárquico que adopta una posición intermedia entre el centroide y todos los puntos extremos. Un número "c" de puntos dispersos es seleccionado y contraidos hacia un centroide por una fracción "α", posteriormente los puntos dispersos son utilizados como representates del clúster. Los clústeres con el par de representates más cercanos serán los que se fusionaran en cada paso del algoritmo.	Categóricos y Numéricos Continuos y Discretos	Es eficiente para grandes bases de datos. No es sensible a valores atípicos. Identifica clústeres con formas no esféricas y de tamaños variados.	Solo considera la proximidad de los clústeres, ignorando la interconexión de los mismos. Un poco sensible al ruido.	Dataset: ***BatStateU (Base de datos del cuerpo estudiantil de una universidad)*** - 4 atributos * SRCODE (código de estudiante) * CODE (código compuesto por el nombre del cursos y código del asunto) * COURSE CODE (código del curso) * FG (calificación final) - 529 entradas Resultados: - Se consiguieron los mejores resultados en la aplicación del algoritmo CURE con los siguientes parámetros: * (k=15, c=4 y α=0.7) * (k=8, c=3 y α=0.6) * (k=10, c=4 y α=0.7)
ROCK (Aglomerativo)	**Links** (número de vecinos comunes entre dos objetos): Si dos puntos similares también tienen barrios (conjunto de vecinos) similares es probable que pertenezcan al mismo clúster, por lo tanto pueden fusionarse. ROCK construye un gráfico a partir de una matriz de similitud y el concepto de vecinos compartidos.	Booleanos Categóricos y Numéricos Continuos y Discretos	Escala a grandes conjuntos de datos. Cuenta con un muestreo aleatorio. Trabaja eficientemente con el ruido.	Solo hace hincapié en la interconexión de los clústeres, sin tener en cuenta información relativa a la proximidad de los clústeres.	Eficiente para derivar clústeres más significativos en data sets orientados al ámbito político o diseñados para la experimentación.
Chameleon (Aglomerativo)	**Modelado dinámico**: dos clústeres son fusionados si su interconetividad es alta y están muy juntos uno del otro. El algoritmo utiliza el enfoque del vecino "k" mas cercano para construir un gráfico escaso, donde cada vértice representa un objeto de datos, y las aristas las conexiones con otros objetos de gran similitud, las aristas son ponderadas. Utiliza un algoritmo de particionamiento para dividir el gráfico en pequeños sub-clústeres, y un algoritmo de agrupamiento jerárquico aglomerativo es utilizado para fusionar estos sub-clúster en función a su similitud.	Es aplicable a todo tipo de datos, siempre que una función de similitud sea especificada.	No depende de un modelo estático proporcionado por el usuario. Es capaz de adaptarse automáticamente a características internas de los clústeres que se pueden fusionar. Genera clústeres naturales y homogeneos. Es tolerante a valores atípicos.	El algoritmo no puede ser aplicado en bases de datos de grandes dimensiones. La complejidad del algoritmo en bases de datos de grandes dimensiones es de O(n^2).	Se experimentó en 5 diferentes datasets que contenían puntos en dos dimensiones y en formas geométricas, todos excepto uno de los datasets contenía ruido o valores atípicos. Resultados: - Se comprobó que Chameleon fue capáz de identificar correctamente los grupos auténticos de los 5 conjuntos de datos, a diferencia de otros algoritmos que agrupaban varios de estos en un solo clúster. - Los experimentos ilustraron que Chameleon es muy eficaz en la búsqueda de clústeres con forma, densidad y orientacion arbitraria.

DBSCAN	El algoritmo se basa en dos conceptos: accesibilidad y conectividad de la densidad, y depende de dos parámetros de entrada: el radio de alcance "e", y el mínimo de puntos asociados a un clúster "m". Para encontrar un clúster el algoritmo selecciona un punto arbitrario "P" y recupera su densidad alcanzable; si es central se produce un clúster, si es fronterizo y no existen puntos con densidad alcanzable el algoritmo pasa a otro punto. DBSCAN utiliza llamadas recursivas para detectar clústeres con una densidad mayor.	Categóricos y Numéricos Discretos y Continuos	Encuentra los clústeres correctamente, sin importar el tamaño, forma o ubicación de estos respecto a otros clústeres. Es resistente al ruido.	El algoritmo no funciona bien cuando se trata con clústeres de diferentes densidades o con datos de alta dimensión.	Dataset: ***Conjunto de datos bancarios*** - 11 atributos - 600 entradas Resultados: - Clústeres: 3 de 10 casos (40%), 6 casos (24%) y 9 casos (36%). - Tiempo de ejecución: 1.03 seg. - Casos no clasificados: 575.
K-Medoids	El algoritmo recoge objetos reales para representar los clústeres, siendo los objetos restantes agrupados en relación a su similitud con los objetos representativos. El algoritmo itera hasta que cada objeto representativo se vuelve el medoide (medoid) del clúster.	Categóricos y Numéricos Discretos y Continuos	No es sensible a valores atípicos. Supera la dificultad de la aletoriedad de los centros de agrupamiento.	No escala a grandes conjuntos de datos. Dificultad para predecir el valor de "k". Su complejidad computacional es de O(k(n-k)^2).	Ha mostrado resultados eficientes aplicado a conjuntos de datos a los que se les ha aplicado un modelo de Web Semántica.
CLARA (Clustering LARge Applications)	**Muestreo:** Toma una pequeña parte de los datos reales como representante de los mismos, eligiendo los medoides utilizando el algoritmo PAM. El proceso se hace en varias muestras seleccionadas por el algoritmo, devolviendo la mejor agrupación como salida.	Categóricos y Numéricos Discretos y Continuos	No es sensible a valores atípicos. Escala a grandes conjuntos de datos.	No puede encontrar el mejor clustering si alguno de los mejores medoides de la muestra no es uno de los mejores medoides del conjunto de daos completo. Tiene una muestra fija que puede perjudicar el resultado si esta sesgada.	A lo largo de una serie de experimentos en los que se aplicó el algorimto, CLARANS ha demostrado encontrar agrupamiento de mejor calidad que CLARA. En algunos casos, CLARA ha demostrado tener un menor tiempo de ejecución que CLARANS, sin embargo este último mide su tiempo de ejecución en proporción al número de clústeres "k" deseado.
CLARANS (Clustering Large Applications based upon RANdomized Search)	CLARANS utiliza la técnica de muestreo de PAM, sin embargo no se limita a cualquier muestra en cualquier momento dado, sino que cuenta con una muestra con cierta aleatoriedad en cada etapa de la búsqueda. El proceso de agrupamiento se puede ver como una búsqueda a través de un gráfico, donde cada nodo es un conjunto de medoides (soluciones potenciales) con un coste de disimilitud asignado con sus elementos, y los vecinos (unidos a través de arcos) que se definen si dos conjuntos difieren solo en 1 elemento.	Categóricos y Numéricos Discretos y Continuos	No es sensible a valores atípicos. Escala a grandes conjuntos de datos. Cuenta con diferentes muestras durante cada etapa de la búsqueda.	Su complejidad computacional es de O(n^2).	

Tabla 3: Cuadro comparativo de algoritmos de clusterización.
Fuente Propia.

3.2.3. COMPARACIÓN CON RESULTADOS OBTENIDOS EN OTROS ENTORNOS

En este caso la comparación se realizará con una encuesta (survey) que analiza algoritmos de clusterización aplicados al campo de la bioinformática. En este caso Muhammad Ali Masood y M. N. A. Khan (2015), autores del paper, obtuvieron los siguientes resultados:

Algoritmo/Metodología	Fortalezas	Debilidades
K-Means & K-Medoids	En la aplicación este algoritmo de agrupamiento ha demostrado ser capaz de poder actuar como una línea de base para la evaluación de la progresión del rendimiento de estudiantes en una institución de estudios superiores. Se demostro que el algoritmo se alinea con la cantidad de muestras, clústeres, iteraciones y dimensiones, por lo que es escalable.	El tipo y aplicación de datos bajo ciertos tipos de escenarios en particular determina la selección del algoritmo de agrupamiento. La ténica de agrupamiento no se aplico a multiples tipos de conjuntos de datos, por lo que no se pudo evaluar el potencial real. Utiliza ténicas de medida de similtud entre pares, las cuáles no son muy eficaces y requieren mucho tiempo. K-Medoids tiene graves problemas de consumo de recursos y tiempo.
Clustering basado en densidad	La aplicación de algoritmos basados en densidad mejoró el rendimiento de la agrupación con la ventaja de la reducción de la perdida de objetos. Este tipo de algoritmo también mejora el resultado del agrupamiento de conjuntos de datos con alta dimensionalidad y densidad irregular. Además muestra mejora respecto a la sensibilidad a los parámetros. Se determino que es muy eficaz para el procesamiento y análisis de lugares y eventos utilizando grandes colecciones de imágenes y datos geo-etiquetados.	La calidad de los resultados de agrupamiento están estrechamente ligados a los parámetros "umbral de ruido" y "densidad". El algoritmo no es capaz de lidiar correctamente con el conjunto de datos cuando este ha sufrido modificaciones, eliminación o remodelamiento. Diferentes enfoques de aplicación de este tipo de algoritmos no fueron utilizados, por lo que su eficiencia aplicado a otros tipos de repositorios aún no es clara.

Tabla 4: Cuadro comparativo de algoritmos de clusterización aplicados a otro entorno. Fuente: Muhammad Ali Masood y M. N. A. Khan.

En su trabajo y a través de su cuadro comparativo podemos apreciar que para el análisis de repositorios orientados a este entorno en particular se determinó que el algoritmo K-Means así como los que realizan clustering basado en densidad son los más eficientes para el procesamiento de este tipo de datos; sin embargo, el menciona específicamente que los resultados pueden variar dependiendo del tipo del repositorio de datos ya que durante el desarrollo de su trabajo no probaron la eficiencia de dichos algoritmos en otros entornos.

3.2.4. CONCLUSIONES

De los algoritmos analizados, los algoritmos EM y K-Means muestran ser los más eficientes para la clusterización de datasets orientados al ámbito comercial dadas las siguientes razones:

A. El algoritmo EM basa su análisis en estimaciones de máxima verosimilitud para un determinado parámetro ɵ haciendo suposiciones acerca de los datos completos, siendo un método iterativo con un modelo que depende de las variables no observadas. Dicha técnica puede ser de gran utilidad para el análisis del comportamiento de clientes, ya que no siempre se refleja específicamente aquellos parámetros que se desea analizar, y es en base a otros atributos que se pueden determinar agrupaciones de interés y definir tendencias de mayor verosimilitud de los datos. El algoritmo también trabaja con varios tipos de atributos y cuenta con varias características positivas, entre las que destacan la simplicidad conceptual y la facilidad de aplicación. El algoritmo aplicado a un

conjunto de datos bancarios, relacionado directamente con las transacciones comerciales online, mostró un excelente agrupamiento de los datos, siendo las agrupaciones bastante especializadas, sin embargo su tiempo de ejecución fue algo elevado tardándose más de 75 segundos.

B. El algoritmo K-Means es comúnmente utilizado por investigadores dada su eficiencia al momento de realizar clustering en datasets orientados a distintos entornos. Para un entorno comercial en el que se busca analizar patrones frecuentes, donde los datasets contarán comúnmente con miles de registros asociados a transacciones, el algoritmo K-Means será eficaz y rápido dado que fraccionará la data en k agrupaciones al inicio, haciendo más fácil su procesamiento; además, el algoritmo trabaja con distintos tipos de atributos, de manera que no causará conflicto con este tipo de dataset. Probado en un conjunto de datos bancarios generó pocos clústeres, siendo estos menos especializados que los generados por el algoritmo EM; sin embargo, su tiempo de ejecución fue instantáneo, lo que facilitaría un análisis rápido de enormes bases de datos, manteniendo un nivel de confiabilidad y precisión aceptable.

3.3. ASOCIACIÓN

3.3.1. ALGORITMOS

A. A PRIORI

Es uno de los algoritmos más populares en minería de datos para el aprendizaje del concepto que conocemos como “reglas de asociación”. Está siendo utilizado por muchas personas específicamente para el procesamiento de información relacionada a operaciones de transacción y en aplicaciones de tiempo real mediante la recopilación de artículos comprados por clientes sobre el tiempo, de modo que conjuntos de ítems frecuentes puedan ser generados. Conjuntos de elementos frecuentes se pueden encontrar muy fácilmente debido a su explosión combinatoria, una vez que se obtienen es simplemente fácil generar las reglas de asociación con una confianza mayor o igual a una confianza mínima especificada por el usuario; sin embargo el algoritmo A priori sufre de algunas deficiencias a pesar de ser claro y sencillo, la principal limitación es el costo de tiempo que se emplea para contener el gran número de conjuntos de datos candidatos junto a los muchos conjuntos de elementos frecuentes (Akshita Bhandari, Ashutosh Gupta y Debasis Das, 2014).

A su vez existen variaciones del algoritmo A priori, propuestas con el fin de mejorar la eficiencia del algoritmo original (Jiawei Han y Micheline Kamber, 2006).

a) **Basado en la técnica Hash** (Hashing conjuntos de elementos frecuentes en sus cubos o grupos correspondientes)

Una técnica basada en Hash se puede utilizar para recudir el tamaño de los candidatos para k-itemsets (conjunto de datos) para k > 1.

b) **Reducción de transacciones** (reducir el número de transacciones de scaneo para futuras iteraciones)

Una transacción que no contiene ningún conjunto de elementos "k" frecuente no puede contener ningún conjunto de elementos "k+1". Por tanto una transacción de este tipo puede ser marcada o removida de una futura consideración debido a que escaneos posteriores de la base de datos para conjuntos de elementos "j", donde j > k, no, etc. lo necesitan.

c) **Particionamiento** (Particionamiento de los datos para encontrar conjuntos de elementos candidatos)

Una técnica de particionamiento se puede utilizar de tal manera que solo se necesite dos escaneos a la base de datos para extraer los conjuntos de elementos frecuentes. Se compone de dos fases: En la fase I el algoritmo subdivide las transacciones de "D" en "n" particiones no superpuestas. Si el umbral mínimo de apoyo para las transacciones en "D" es "min_sup", entonces el contador de soporte mínimo para una partición sería "min_sup" * el número de transacciones en esa partición. Para cada partición, todos los conjuntos de elementos frecuentes dentro

de la partición son encontrados, estos se conocen como conjuntos de elementos frecuentes locales. El procedimiento emplea una estructura de datos especial que, para cada conjunto de elementos, registra los TIDs de las transacciones que contienen los elementos del conjunto de elementos, esto permite encontrar todos los conjuntos de elementos frecuentes locales en solo 1 escaneo de la base de datos. Un conjunto de elementos frecuente local puede o no ser frecuente con respecto a toda la base de datos "D", cualquier conjunto de elementos potencialmente frecuente con respecto a "D" debe ocurrir como un conjunto de elementos frecuente en al menos una de las particiones, por lo tantos todos los conjuntos de elementos frecuentes locales son conjuntos de elementos candidatos respecto a "D". En la fase II, un segundo escaneo a "D" se lleva a cabo en el que el apoyo actual de cada candidato se evalúa con el fin de determinar los conjuntos de elementos frecuentes globales. El tamaño de la partición y el número de particiones están establecidos de manera que cada partición pueda caber en la memoria principal, y por tanto solo puede ser leída una vez en cada fase.

d) Muestreo (Minería en un subconjunto de los datos dados)

La idea básica del método de muestreo es recoger una muestra aleatoria "S" del conjunto de datos "D" dado, y luego buscar los conjuntos de elementos frecuentes en "S" en lugar de "D", de esta forma cambiamos cierto grado de precisión por eficiencia. El tamaño de la muestra de "S" es tal que la búsqueda de conjuntos de elementos frecuentes en "S" se puede hacer en la memoria principal, por lo que se

suele requerir únicamente un escaneo a “S”. Debido a que se está en busca de conjuntos de elementos frecuentes en “S” en lugar de “D”, es posible que se vaya a perder algunos conjuntos de elementos frecuentes globales; para disminuir esta posibilidad se utiliza un umbral de apoyo más bajo que el apoyo mínimo para encontrar los conjuntos de elementos frecuentes locales para “S”, denotado como “LS”; el resto de la base de datos es entonces utilizada para calcular las frecuencias reales de cada conjuntos de elementos en “LS”. Un mecanismo se utiliza para determinar si todos los conjuntos de elementos frecuentes globales están incluidos en “LS”, si en “LS” realmente están todos los conjuntos de elementos de “D”, solo se requerirá un escaneo de “D”, de lo contrario un segundo escaneo puede ser realizado con el fin de encontrar los conjuntos de elementos frecuentes que se perdieron. El enfoque del muestreo es especialmente beneficioso cuando la eficiencia es de suma importancia.

e) **Conteo de conjuntos de elementos dinámicos** (Añadiendo conjuntos de elementos candidatos en diferentes puntos durante el escaneo)

La técnica de conteo de conjuntos de elementos dinámicos fue propuesta de manera que la base de datos es dividida en bloques marcados por puntos de inicio. En esta variación los nuevos conjuntos de elementos candidatos se pueden añadir a cualquier punto de inicio, a diferencia de A priori, que determina los nuevos conjuntos de elementos candidatos sólo antes de cada escaneo completo a la base de datos. La técnica es dinámica, en la que se estima el apoyo de todos los conjuntos

de elementos que se han contado hasta el momento, añadiendo nuevos conjuntos de elementos candidatos si todos sus subconjuntos son estimados como frecuentes; el algoritmo resultante requiere un menor número de escaneos a la base de datos que A priori.

3.3.2. COMPARACIÓN CON RESULTADOS OBTENIDOS EN OTROS ENTORNOS

Como es común para este algoritmo, la gran mayoría de trabajos acerca de su aplicación se centran en las reglas de asociación relacionadas a conjuntos de ítems o elementos frecuentes, en este caso se hizo la comparación con su aplicación a una tienda de comestibles común. Pragya Agarwal, Madan Lal Yadav y Nupur Anand (2013) llegaron a la siguiente conclusión:

- Dado que actualmente es el mejor algoritmo para generar reglas de asociación, o por lo menos es el mayormente utilizado, generó con éxito los conjuntos de elementos más frecuentes; sin embargo, ellos señalan que A priori cuenta con varios problemas que podrían solucionarse de usarse otros esquemas, por ejemplo su tiempo de ejecución en grandes bases de datos, número de escaneos a la base de datos, el consumo de memoria y la calidad de reglas halladas.

3.3.3. CONCLUSIONES

El único algoritmo tomado en cuenta en esta sección fue el algoritmo A priori, no solo por ser el mayormente utilizado debido a su gran eficiencia al momento de encontrar reglas de asociación, sino por la efectividad que

presentaría al aplicarse a dataset orientados al comercio electrónico, dadas las siguientes razones:

A. A priori es sumamente utilizado ya que hace una búsqueda eficiente de conjuntos de elementos frecuente en base a su explosión combinatoria; además se ha demostrado que la confianza de estas reglas de asociación generada siempre es mayor o igual a la esperada por los usuarios. Además que las variaciones en los dataset de comercio electrónico, que podrían presentarse, podrán ser solventadas ya que el algoritmo cuenta con distintas variaciones capaces de adaptarse a diferentes situaciones, tales como la técnica Hash, el muestreo, etc. Es demostrado en diferentes trabajos que el algoritmo es frecuentemente utilizado para el procesamiento de información relacionada a operaciones de transacción y en aplicaciones de tiempo real que abarquen a cliente y sus artículos comprados sobre un tiempo determinado.

CAPÍTULO IV: EXPERIMENTACIÓN EN REPOSITORIOS DE DATOS

En este capítulo veremos el análisis experimental realizado a los algoritmos identificados como más eficientes en el capítulo anterior. Los repositorios utilizados para el análisis fueron extraídos de UCI Machine Learning Repository.

REPOSITORIOS DE DATOS

4.1. POPULARIDAD DE NOTICIAS EN LINEA (ONLINE NEWS POPULARITY)

Este conjunto de datos resume un conjunto heterogéneo de características sobre artículos publicados por Mashable (www.mashable.com) en un periodo de 2 años. El conjunto de datos tiene 61 atributos y 39645 instancias.

4.1.1. DICCIONARIO DE DATOS

ID	*Atributo*	*Tipo*	*Descripción*
0	URL	Categórico	Dirección URL del artículo.
1	timedelta	Numérico	Días entre la publicación del artículo y la adquisición del conjunto de datos.
2	n_tokens_title	Numérico	Número de palabras en el título.
3	n_tokens_content	Numérico	Número de palabras en el contenido.
4	n_unique_tokens	Numérico	Tasa de palabras únicas en el contenido.
5	n_non_stop_words	Numérico	Tasa de palabras no comunes en el contenido.
6	n_non_stop_unique_tokens	Numérico	Tasa de palabras únicas no comunes en el contenido.
7	num_hrefs	Numérico	Número de links.
8	num_self_hrefs	Numérico	Número de links a otros artículos publicados por Mashable.
9	num_imgs	Numérico	Número de imágenes.
10	num_videos	Numérico	Número de videos.
11	average_token_length	Numérico	Longitud media de la palabras en el contenido.
12	num_keywords	Numérico	Número de palabras clave en la metadata.
13	data_channel_is_lifestyle	Binario	El canal de datos es de categoría "estilo de vida".
14	data_channel_is_entertaiment	Binario	El canal de datos es de categoría "entretenimiento".
15	data_channel_is_bus	Binario	El canal de datos es de categoría "negocios".

16	data_channel_is_socmed	Binario	El canal de datos es de categoría "medios sociales".
17	data_channel_is_tech	Binario	El canal de datos es de categoría "tecnología".
18	data_channel_is_world	Binario	El canal de datos es de categoría "mundo".
19	kw_min_min	Numérico	Peor palabra clave (min. Acciones)
20	kw_max_min	Numérico	Peor palabra clave (max. Acciones)
21	kw_avg_min	Numérico	Peor palabra clave (promedio de acciones)
22	kw_min_max	Numérico	Mejor palabra clave (min. Acciones)
23	kw_max_max	Numérico	Mejor palabra clave (max. Acciones)
24	kw_avg_max	Numérico	Mejor palabra clave (promedio de acciones)
25	kw_min_avg	Numérico	Promedio de palabras clave (min. Acciones)
26	kw_max_avg	Numérico	Promedio de palabras clave (max. Acciones)
27	kw_avg_avg	Numérico	Promedio de palabras clave (promedio de Acciones)
28	self_reference_min_shares	Numérico	Mínimo de acciones de artículos referenciados en Mashable.
29	self_reference_max_shares	Numérico	Máximo de acciones de artículos referenciados en Mashable.
30	self_reference_avg_shares	Numérico	Promedio de acciones de artículos referenciados en Mashable.
31	weekday_is_monday	Binario	El artículo fue publicado un lunes.
32	weekday_is_tuesday	Binario	El artículo fue publicado un martes.
33	weekday_is_wednesday	Binario	El artículo fue publicado un miercoles.
34	weekday_is_thursday	Binario	El artículo fue publicado un jueves.
35	weekday_is_friday	Binario	El artículo fue publicado un viernes.
36	weekday_is_saturday	Binario	El artículo fue publicado un sábado.
37	weekday_is_sunday	Binario	El artículo fue publicado un domingo.
38	is_weekend	Binario	El artículo fue publicado un fin de semana.
39	LDA_00	Numérico	Cercano al tópico 0 LDA.
40	LDA_01	Numérico	Cercano al tópico 1 LDA.
41	LDA_02	Numérico	Cercano al tópico 2 LDA.
42	LDA_03	Numérico	Cercano al tópico 3 LDA.
43	LDA_04	Numérico	Cercano al tópico 4 LDA.
44	global_subjetivity	Numérico	Subjetividad del texto.
45	global_sentiment_polarity	Numérico	Polaridad del sentimiento del texto.
46	global_rate_positive_words	Numérico	Tasa de palabras positivas en el cotenido.
47	global_rate_negative_words	Numérico	Tasa de palabras negativas en el contenido.
48	rate_positive_words	Numérico	Tasa de palabras positivas entre los tokens no neutrales.
49	rate_negative_words	Numérico	Tasa de palabras negativas entre los tokens no neutrales.
50	avg_positive_polarity	Numérico	Promedio de la polaridad de palabras positivas.
51	min_positive_polarity	Numérico	Mínimo de la polaridad de palabras positivas.
52	max_positive_polarity	Numérico	Máximo de la polaridad de palabras positivas.
53	avg_negative polarity	Numérico	Promedio de la polaridad de palabras negativas.
54	min_negative_polarity	Numérico	Minimo de la polaridad de palabras negativas
55	max_negative_polarity	Numérico	Máximo de la polaridad de palabras negativas.
56	title_subjectivity	Numérico	Subjetividad del título.
57	title_sentiment_polarity	Numérico	Polaridad del título.
58	abs_title_subjectivity	Numérico	Nivel de subjetividad absoluto.
59	abs_title_sentiment_polarity	Numérico	Nivel de polaridad absoluta.
60	shares	Numérico	Número de acciones (objetivo).

Tabla 5: Diccionario de datos del repositorio de datos "Online News Popularity". Fuente propia.

4.1.2. PRE PROCESAMIENTO

El atributo **n_unique_tokens** representa el porcentaje referente a palabras únicas en el contenido (el cual debería ir de 0 a 1), sin embargo existe un outlier o valor atípico que sale de este rango como se muestra en la gráfica a continuación:

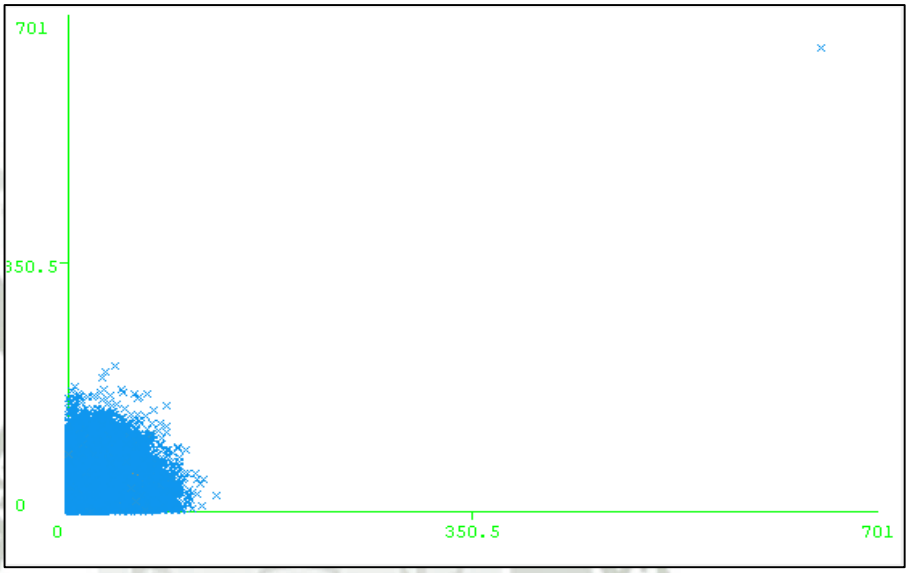

Figura 1: Distribución del atributo n_unique_tokens antes del filtrado de valores. Fuente: WEKA.

Para tratar con ello se utiliza el algoritmo RemoveWithValues de WEKA para quedarnos con el rango deseado de 0 a 1, quedando la nueva distribución de la siguiente manera:

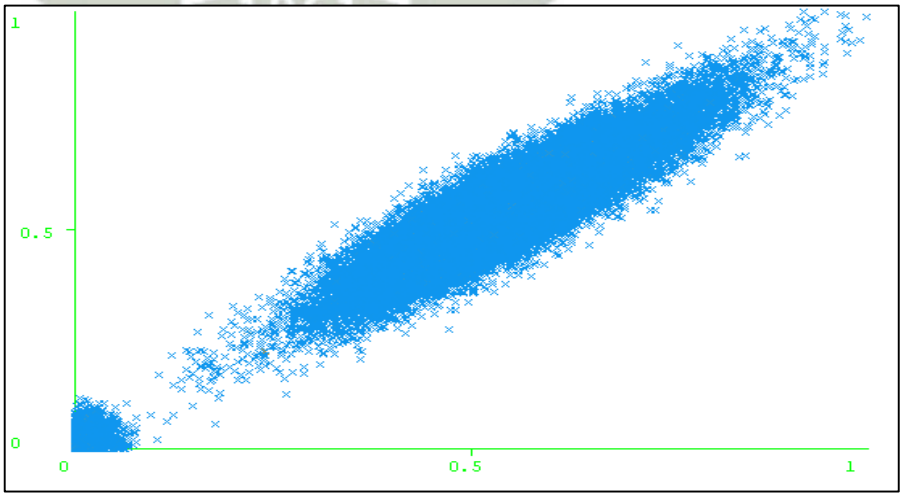

Figura 2: Distribución del atributo n_unique_tokens después del filtrado de valores. Fuente: WEKA.

Atributos **n_non_stop_words** y **n_non_stop_unique_words** presentan problemas similares y la misma solución.

Los atributos **data_channel_is_life**, **data_channel_is_entertainment**, **data_channel_is_bus**, **data_channel_is_socmed**, **data_channel_is_tech** y **data_channel_is_world** cuentan con únicamente dos valores, siendo variables binarias, sin embargo son tratadas como numéricas por WEKA, como se muestra a continuación:

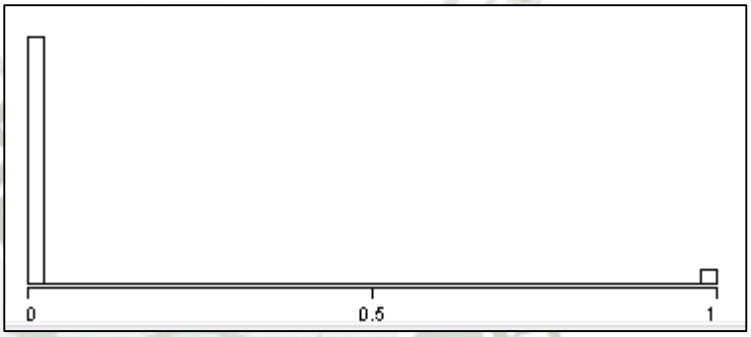

Figura 3: Variable binaria siendo tratada como numérica. Fuente: WEKA.

Se utilizó el algoritmo NumericToNominal para transformar estas variables, las columnas representan 0 o 1, las cuales indican si la noticia pertenece o no al canal correspondiente. El resultado fue el siguiente:

Figura 4: Variable data_channel_is_bus (binaria). Fuente: WEKA.

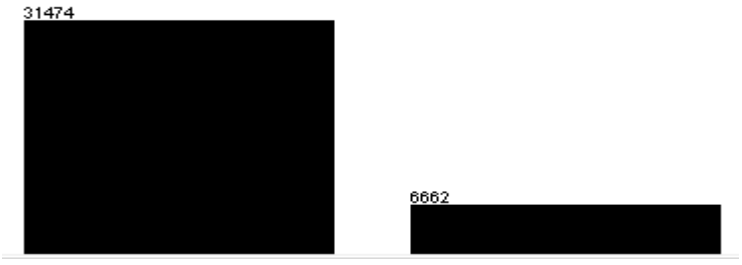

Figura 5: Variable data_channel_is_entertainment (binaria). Fuente: WEKA.

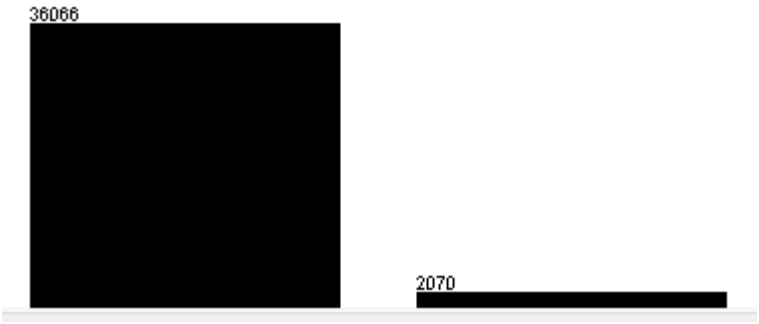

Figura 6: Variable data_channel_is_lifestyle (binaria). Fuente: WEKA.

Figura 7: Variable data_channel_is_socmed (binaria). Fuente: WEKA.

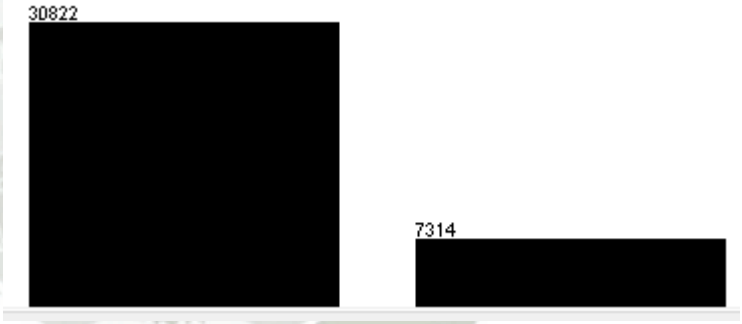

Figura 8: Variable data_channel_is_tech (binaria). Fuente: WEKA.

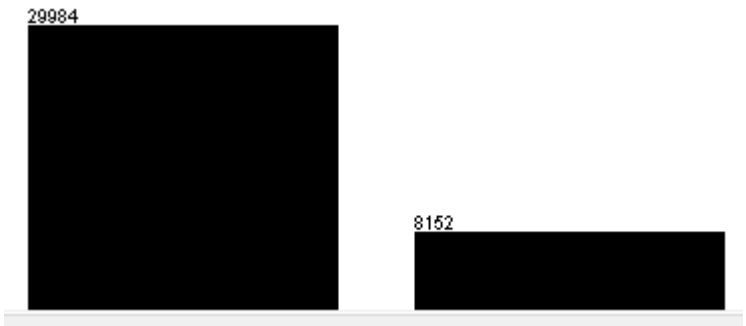

Figura 9: Variable data_channel_is_world (binaria). Fuente: WEKA.

El mismo algoritmo es utilizado para transformar a binarios los atributos **weekday_is_monday**, **weekday_is_tuesday**, **weekday_is_wednesday**, **weekday_is_thursday**, **weekday_is_friday**, **weekday_is_saturday**, **weekday_is_sunday** y **is_weekend**, los cuales representan con un 1 o 0 si el día en que se publicó el artículo fue un lunes, martes, miércoles, jueves, viernes, sábado, domingo o fin de semana respectivamente.

Finalmente se procedió a discretizar el conjunto de datos, ya que algunos algoritmos de los que veremos sólo aceptan atributos categóricos; además, el proceso de aprendizaje frecuentemente es menos eficiente y menos efectivo cuando el conjunto de datos tiene un gran número de variables cuantitativas. El algoritmo utilizado fue **Discretize**, presente en WEKA; todos los atributos fueron discretizados para tener 4 grupos con frecuencias los más cercanas posible, salvo los atributos n_unique_tokens, n_non_stop_words y n_non_stop_unique_tokens, que al representar porcentajes con valores pequeños se decidió discretizarlos únicamente en dos grupos.

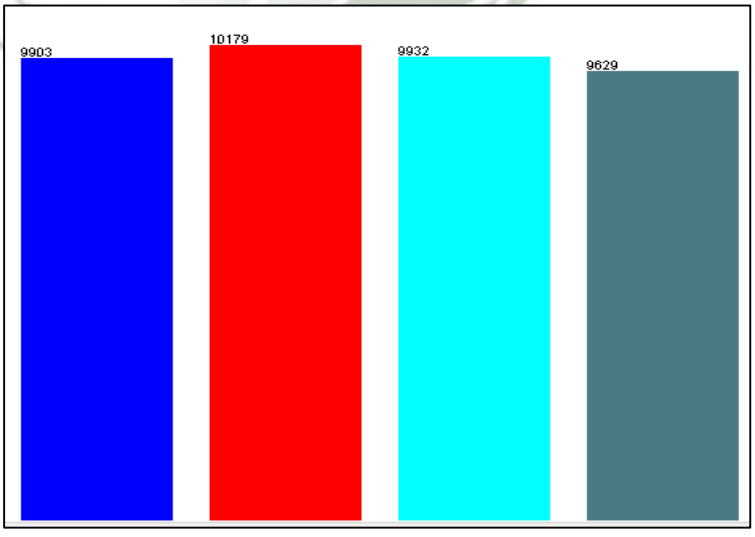

Figura 10: Ejemplo de discretización del atributo "shares".
Fuente: WEKA.

4.1.3. APLICACIÓN DE ALGORITMOS

En este caso el objetivo es predecir el número de acciones "compartir" de las noticias publicadas en el sitio web www.mashable.com en redes sociales, y la relación que tiene esta "popularidad" con sus atributos. El atributo objetivo dentro del conjunto de datos es "**shares**".

A. ALGORITMO J48

a) Algoritmo J48 – Eficiencia del Algoritmo

```
Time taken to build model: 22.91 seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances       32068               80.892  %
Incorrectly Classified Instances      7575               19.108  %
Kappa statistic                          0.7452
Mean absolute error                      0.1332
Root mean squared error                  0.2581
Relative absolute error                 35.5359 %
Root relative squared error             59.612  %
Total Number of Instances            39643

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.866     0.078      0.787     0.866     0.825      0.967    '(-inf-945.5]'
                 0.794     0.066      0.807     0.794     0.801      0.956    '(945.5-1450]'
                 0.772     0.053      0.828     0.772     0.799      0.959    '(1450-2850]'
                 0.804     0.058      0.817     0.804     0.811      0.967    '(2850-inf)'
Weighted Avg.    0.809     0.064      0.81      0.809     0.809      0.962

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 8574  540  388  401 |    a = '(-inf-945.5]'
  977 8086  529  587 |    b = '(945.5-1450]'
  756  765 7664  747 |    c = '(1450-2850]'
  587  626  672 7744 |    d = '(2850-inf)'
```

Figura 11: Resultados del algoritmo J48 en Online News Popularity. Fuente: WEKA.

Podemos ver que un 80.892% de las instancias fueron correctamente clasificadas y que la tasa de verdaderos positivos es moderadamente alta para las 4 clases del atributo "shares".

También podemos ver que la diagonal central de la matriz de confusión tiene los valores más altos, por lo que podemos decir que la clasificación es confiable.

b) Algoritmo J48 – Análisis del Conocimiento

En este caso, dada la gran cantidad de atributos del repositorio de datos, el árbol generado es muy grande, por lo que para comprender cómo analizar la calidad del conocimiento se tomaron algunos fragmentos.

```
num_videos = '(-inf-0.5]'
|    LDA_00 = '(-inf-0.025051]'
|   |    global_sentiment_polarity = '(-inf-0.057762]'
|   |   |    global_rate_positive_words = '(-inf-0.028388]': '(1450-2850]' (2.0)
|   |   |    global_rate_positive_words = '(0.028388-0.039023]': '(945.5-1450]' (3.0/1.0)
|   |   |    global_rate_positive_words = '(0.039023-0.050283]': '(-inf-945.5]' (1.0)
|   |   |    global_rate_positive_words = '(0.050283-inf)': '(1450-2850]' (0.0)
|   |    global_sentiment_polarity = '(0.057762-0.119138]'
|   |   |    weekday_is_wednesday = 0
|   |   |   |    weekday_is_thursday = 0: '(-inf-945.5]' (5.0/1.0)
|   |   |   |    weekday_is_thursday = 1: '(1450-2850]' (3.0)
|   |   |    weekday_is_wednesday = 1: '(945.5-1450]' (2.0/1.0)
|   |    global_sentiment_polarity = '(0.119138-0.177846]'
|   |   |    title_sentiment_polarity = '(-inf-0]': '(945.5-1450]' (3.0/1.0)
|   |   |    title_sentiment_polarity = '(0-0]'
|   |   |   |    min_negative_polarity = '(-inf--0.707143]': '(1450-2850]' (2.0)
|   |   |   |    min_negative_polarity = '(-0.707143--0.525]': '(945.5-1450]' (4.0/2.0)
|   |   |   |    min_negative_polarity = '(-0.525--0.341667]'
|   |   |   |   |    self_reference_min_shares = '(-inf-638.5]': '(-inf-945.5]' (4.0)
|   |   |   |   |    self_reference_min_shares = '(638.5-1250]': '(945.5-1450]' (2.0)
|   |   |   |   |    self_reference_min_shares = '(1250-2750]': '(-inf-945.5]' (0.0)
|   |   |   |   |    self_reference_min_shares = '(2750-inf)': '(-inf-945.5]' (1.0)
|   |   |   |    min_negative_polarity = '(-0.341667-inf)'
|   |   |   |   |    self_reference_max_shares = '(-inf-1050]': '(1450-2850]' (0.0)
|   |   |   |   |    self_reference_max_shares = '(1050-2850]': '(945.5-1450]' (2.0)
|   |   |   |   |    self_reference_max_shares = '(2850-7950]': '(1450-2850]' (2.0)
|   |   |   |   |    self_reference_max_shares = '(7950-inf)': '(1450-2850]' (3.0)
|   |   |    title_sentiment_polarity = '(0-0.250758]': '(945.5-1450]' (6.0/2.0)
|   |   |    title_sentiment_polarity = '(0.250758-inf)': '(1450-2850]' (2.0)
|   |    global_sentiment_polarity = '(0.177846-inf)'
|   |   |    max_negative_polarity = '(-inf--0.129167]': '(-inf-945.5]' (5.0/2.0)
|   |   |    max_negative_polarity = '(-0.129167--0.091667]': '(-inf-945.5]' (6.0/1.0)
|   |   |    max_negative_polarity = '(-0.091667--0.041667]'
|   |   |   |    weekday_is_thursday = 0: '(945.5-1450]' (3.0)
|   |   |   |    weekday_is_thursday = 1: '(2850-inf)' (4.0/2.0)
|   |   |    max_negative_polarity = '(-0.041667-inf)': '(945.5-1450]' (2.0/1.0)
```

Figura 12: Fragmento del árbol de clasificación de ONP generado por J48. Fuente: WEKA.

En este fragmento podemos interpretar, por ejemplo, que cuando el número de videos es inferior a 0.5 (dada la discretización podemos asumir que la noticia no contaba con ningún video), su porcentaje de

pertenencia al primer grupo según el Latent Dirichlet Allocation es inferior al 2.5051%, la polaridad global del texto es inferior al 5.7762% y la tasa de palabras positivas en el contenido es inferior al 2.83888%, en dos ocasiones la cantidad de veces que se compartió una noticia fue inferior a 945.5.

También podemos ver en este fragmento que el algoritmo J48 emplea todos los atributos del repositorio de datos mostrando la relación existente entre ellos y el atributo objetivo **shares**; en otras palabras, las diferentes ramas del árbol nos muestran cómo dados ciertos valores para cada atributo influyen, ya sea de manera positiva o negativa, al número de veces que es compartida una noticia en redes sociales.

Dentro del todo el conocimiento obtenido podemos ver ejemplos de información relevante para la toma decisiones dentro del entorno estudiado, en este caso la popularidad de las noticias en www.mashable.com; por ejemplo:

```
global_rate_positive_words = '(0.039023-0.050283]'
|   n_non_stop_unique_tokens = '(-inf-0.690476]': '(945.5-1450]' (21.0/6.0)
|   n_non_stop_unique_tokens = '(0.690476-inf)'
|   |   num_imgs = '(-inf-0.5]': '(945.5-1450]' (2.0)
|   |   num_imgs = '(0.5-1.5]': '(2850-inf)' (5.0/1.0)
|   |   num_imgs = '(1.5-9.5]': '(2850-inf)' (0.0)
|   |   num_imgs = '(9.5-inf)': '(2850-inf)' (0.0)
```

Figura 13: Primer ejemplo de conocimiento encontrado en "Online News Popularity" por J48. Fuente: WEKA.

En este caso podemos ver que para un ratio de palabras positivas en el contenido de entre 3.9023% a 5.0283% y una tasa de palabras poco comunes o "palabras no vacías" únicas en el contenido mayor a

69.0476%, el número de imágenes juega un papel decisivo en el nivel de popularidad de una noticia; por ejemplo se pudo determinar que cuando una noticia no cuenta con imágenes (discretizado como inferior a 0.5) la noticia no suele ser muy popular, ya que se registró que en dos ocasiones la cantidad de veces se compartió una noticia con estas características fue entre 946 a 1450, mientras que se registró que 5 noticias con una o dos imágenes fueron compartidas en redes sociales más de 2850 veces; por otro lado noticias con más de 3 imágenes nunca son compartidas a este nivel.

Otro ejemplo del tipo de conocimiento obtenido en este repositorio de datos es el siguiente:

```
title_subjectivity = '(-inf-0.011111]'
|   self_reference_max_shares = '(-inf-1050]'
|   |   rate_positive_words = '(-inf-0.599569]': '(945.5-1450]' (1.0)
|   |   rate_positive_words = '(0.599569-0.708234]': '(2850-inf)' (2.0)
```

Figura 14: Segundo ejemplo de conocimiento encontrado en "Online News Popularity" por J48. Fuente: WEKA.

En este caso se tiene en cuenta la subjetividad del título de la noticia, donde nos indica que si es inferior al 1.1111%, el máximo de acciones "compartir" a través de referencias en otros artículos de www.mashable.com es inferior a 1050, y además la tasa de palabras positivas en el contenido en inferior al 59.9569%, se registró que solo en una ocasión una noticia con estas características se compartió entre 946 y 1450 veces; en cambio se registró dos ocasiones en que noticias con una tasa de palabras positivas de entre 59.9569% y 70.8234% fueron compartidas más de 2850 veces.

Las ramas de conocimiento pueden o no ser significativas, 5 de las más relevantes encontradas por el algoritmo fueron las siguientes:

1. Cuando el canal de información al que pertenece la noticia es "tecnología", la noticia contó con 1 o 0 imágenes, la tasa de palabras vacías únicas en el contenido es inferior al 69.0476%, el mínimo de polaridad de palabras positivas es inferior al 4.1667% y la máxima polaridad de palabras negativas estuvo entre -12.9167% y -9.1667%, se registró 40 ocasiones en que noticias con estas características fueron compartidas entre 1450 y 2850 veces en redes sociales; 14 de estas instancias fueron incorrectamente clasificadas. En este caso podemos interpretar que cuando una noticia habla de tecnología y esta suele usar palabras negativas o controversiales respecto al tema suele ser constantemente compartida, sin embargo, no llega al punto de ser muy popular entre personas que navegan a través de redes sociales.
2. En 41 ocasiones cuando la noticia fue publicada un fin de semana, el máximo de acciones con el promedio de palabras clave fue inferior a 3562, el número de links a otros artículos fue inferior a 2 y la tasa de palabras positivas en el contenido fue superior al 80.0621%, ésta fue compartida entre 1450 y 2850 veces; 15 de estas instancias fueron incorrectamente clasificadas. En este caso podemos interpretar que cuando una noticia es publicada entre viernes y domingo (fin de semana), palabras clave comunes de la noticia no suelen ser muy utilizadas por los usuarios y se utiliza

un lenguaje positivo para describir dicha noticia, ésta suele ser constantemente compartida, sin embargo, llega el punto en que ya no suele ser muy popular entre usuarios de redes sociales.

3. Se determinó que, cuando la tasa de palabras no vacías únicas en el contenido fue superior al 69.0476%, el número de palabras clave estuvo entre 5 y 8, la tasa de palabras positivas en el contenido fue inferior al 2.8388%, el mínimo de acciones con la peor palabra clave fue de 0 o 1 y la noticia contó únicamente con 1 o 2 imágenes, noticias con estas características en 37 ocasiones fueron compartidas menos de 945 veces; 7 de estas instancias fueron incorrectamente clasificadas. En este caso podemos interpretar que cuando una noticia no suele tener palabras comúnmente usadas en su contenido, que tiene varias palabras clave y sin embargo algunas de ella nunca son utilizadas, y que cuenta con una mínima cantidad de imágenes, éstas son muy poco populares en redes sociales.
4. Cuando el mínimo de acciones de artículos referenciados es superior a 2750, la noticia contó con al menos 1 video y el promedio de polaridad de palabras positivas en el contenido estuvo entre 30.6249% y 35.8762%, se registró 35 ocasiones en que la noticias fue compartida más de 2850 veces; 8 de estas instancias fueron incorrectamente clasificadas. En este caso podemos interpretar que noticias cuyos enlaces a otras noticias son constantemente utilizados, suelen utilizar más palabras negativas que positivas, y casi no cuentan o simplemente carecen

de material audiovisual, son muy poco populares entre usuarios de redes sociales.

5. Se registró 42 ocasiones en que cuando, el promedio de acciones del promedio de palabras clave de una noticia estuvo entre 2382 y 2870, la noticia contó con 1 o 2 imágenes, el promedio de acciones de artículos referenciados fue inferior a 981, contó con 0 o 1 video, el promedio de acciones con la peor palabra clave fue de 141 y la subjetividad del título fue inferior al 1.1111%, ésta no fue compartida más de 945 veces. En este caso podemos interpretar que cuando una noticia suele ser comúnmente hallada por sus palabras clave pero que cuenta con poco material audiovisual, cuyos enlaces a otras noticias son muy poco utilizados, y además cuentan con muy poca subjetividad en sus títulos, éstas son muy poco populares entre los usuarios de redes sociales.

c) ***Algoritmo J48 – Conclusión***: El algoritmo J48 mostró muy buenos resultados al momento de clasificar todas las instancias teniendo en cuenta el atributo objetivo "**shares**", mostrando correctamente todas las correlaciones que tiene este con el resto de atributos del repositorio de datos; el conocimiento generado ha demostrado ser relevante para la toma de decisiones, por ejemplo para determinar que tanto influye a la popularidad de una noticia que éstas lleven imágenes, videos, o que el texto de la misma este influenciado por el sentimiento del autor, etc., esto

sin tener en cuenta que varias de las ramas no son significantes para el resultado final, ya que cuentan con información irrelevante, tales como la mayoría de atributos binarios como si una noticia fue o no publica un fin de semana. En cuanto a la eficiencia del algoritmo, la tasa de verdaderos positivos no fue excesivamente alta para las clases del atributo, habiendo muchas instancias que fueron incorrectamente clasificadas; sin embargo, el tiempo que le tomó al algoritmo construir el modelo fue muy bueno dada la cantidad de instancias, siendo tan solo poco más de 22 segundos.

B. ALGORITMO J48-GRAFT

a) Algoritmo J48-Graft – Eficiencia del Algoritmo

```
Time taken to build model: 92.93 seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances       31976               80.6599 %
Incorrectly Classified Instances      7667               19.3401 %
Kappa statistic                          0.7421
Mean absolute error                      0.1343
Root mean squared error                  0.2591
Relative absolute error                 35.8099 %
Root relative squared error             59.8414 %
Total Number of Instances            39643

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.87      0.08       0.784     0.87      0.825      0.967    '(-inf-945.5]'
                 0.789     0.066      0.805     0.789     0.797      0.955    '(945.5-1450]'
                 0.769     0.054      0.827     0.769     0.797      0.959    '(1450-2850]'
                 0.799     0.058      0.815     0.799     0.807      0.966    '(2850-inf)'
Weighted Avg.    0.807     0.065      0.808     0.807     0.806      0.962

=== Confusion Matrix ===

    a    b    c    d   <-- classified as
 8617  520  369  397 |    a = '(-inf-945.5]'
 1003 8035  546  595 |    b = '(945.5-1450]'
  765  779 7634  754 |    c = '(1450-2850]'
  613  645  681 7690 |    d = '(2850-inf)'
```

Figura 15: Resultados del algoritmo J48-Graft en "Online News Popularity".
Fuente: WEKA

Podemos ver que un 80.6599% de las instancias fueron correctamente clasificadas y que la tasa de verdaderos positivos es moderadamente alta para las 4 clases del atributo “shares”; cabe destacar que tanto estos valores como la matriz de confusión, la cual también nos brinda un buen nivel de confiabilidad dado que la diagonal central es mayor que el resto de valores de la matriz, son muy similares a los proporcionados por el algoritmo J48, hallándose la principal diferencia en el tiempo que le toma al algoritmo generar el modelo, en este caso 92.93 segundos.

b) Algoritmo J48-Graft – Análisis del Conocimiento

En este caso, 5 de las ramas de conocimiento más relevantes encontradas por el algoritmo fueron las siguientes:

1. Se determinó que en 32 ocasiones las noticias que, contaron con un promedio de acciones del promedio de palabras clave superior a 3600, tuvieron 1 o ningún video, tuvieron 1 o ninguna imagen y el promedio de acciones de la mejor palabra clave es 244572, éstas fueron compartidas menos de 945 veces; 13 de estas instancias fueron incorrectamente clasificadas. En este caso podemos interpretar que noticias cuyas palabras clave fueron muy utilizadas pero que contaron con poco material audiovisual fueron pobremente compartidas por usuarios de redes sociales.
2. Se determinó que noticias que, fueron publicadas un fin de semana, pertenecieron a los canales de información de “mundo” y “entretenimiento”, el mínimo de acciones con la

peor palabra clave fue de 50, el número de links a otros artículos en la plataforma web fue de 4 o 5 y la tasa de palabras no vacías únicas en el contenido en inferior al 69.0476%, en 24 ocasiones fueron compartidas entre 1450 y 2850 veces; 7 de estas instancias fueron incorrectamente clasificadas. En este caso podemos interpretar que noticias publicadas de viernes a domingo (fin de semana) con temática "mundo" o "entretenimiento", contó con palabras clave que jamás fueron utilizadas por usuarios y con constantemente utilizados por los usuarios, además el uso de palabras poco frecuentes fue de menos del 69%, éstas fueron frecuentemente leídas y compartidas, sin embargo este valor no fue excesivo habiendo noticias más populares entre los usuarios.

3. Se registró 37 ocasiones en que noticias que, tuvieron una tasa de palabras no vacías únicas en su contenido inferior al 69.0476%, el número de palabras clave fue inferior a 5, la tasa global de palabras positivas en el contenido fue inferior al 2.8388% el mínimo de acciones con la peor palabra clave fue de 0 y la noticia contó con 1 o 2 imágenes, éstas fueron compartidas menos de 945 veces. En este caso podemos interpretar que noticias que hicieron uso en su contenido de menos del 69% de palabras raramente utilizadas, no contaron con más de 5 palabras clave en su metadata, además hubieron palabras clave que jamás fueron utilizadas por los usuarios, contó con muy pocas palabras de polaridad positiva y escasos

archivos multimedia, éstas fueron muy poco populares entre los usuarios, siendo que ninguna superó las 945 acciones "compartir" en redes sociales.

4. Se encontró que, cuando el canal de información es "tecnología", la noticia contó con 1 o ningún video, la tasa de palabras no vacías únicas en el contenido es inferior al 69.0476%, el mínimo de la polaridad de las palabras positivas es inferior al 4.1667% y la máxima polaridad de las palabras negativas estuvo entre -12.9167% y -9.1667%, 40 noticias fueron compartidas entre 1450 y 2850 veces; 14 de estas instancias fueron incorrectamente clasificadas. En este caso podemos interpretar que noticias que pertenecieron a la temática "tecnología", contaron con poco material de audio-video, el empleo de palabras poco frecuentes en su contenido fue inferior al 69% y su contenido en general tiene una tendencia de polaridad negativa, éstas fueron frecuentemente compartidas, sin embargo, su nivel de popularidad no fue excesivo siendo que ninguna fue compartidas más de 2850 veces.
5. Se registraron 40 veces en que, el número de links de la noticia fue superior a 13, su tasa o porcentaje de pertenencia a la tercera categoría del LDA (Latent Dirichlet Allocation) es inferior al 37.5907%, a la segunda categoría es inferior al 2.8574%, a la cuarta categoría es inferior al 2.5051%, a la categoría 0 es inferior al 2.5051% y la mínima polaridad de

palabras negativas es inferior al -70.7143%, las noticias fueron compartidas más de 2850 veces; 17 de estas instancias fueron incorrectamente clasificadas. En este caso podemos interpretar que noticias que contaron con una gran número de links, pertenecieron en base a distintos porcentajes a diferentes categorías del tema (categorías desconocidas determinadas por el LDA), y que su contenido en general fue de polaridad muy negativa, éstas fueron muy populares entre los usuarios de redes sociales, siendo que todas fueron compartidas más de 2850 veces, el nivel de popularidad más alto de los registros.

c) ***Algoritmo J48-Graft – Conclusión***: El algoritmo J48-Graft, al igual que J48, mostró muy buenos resultados al momento de clasificar todas las instancias teniendo en cuenta el atributo objetivo "**shares**". El conocimiento generado también resultó ser muy relevante para la toma de decisiones, dejando de lado nuevamente aquellas ramas que emplean atributos binarios irrelevantes en su mayoría de casos como si una noticia fue o no publicada un lunes. En cuanto a su eficiencia, aunque el número de instancias clasificadas y sin clasificar en las 4 clases del atributo objetivo son muy similares a las generadas por J48, así como el resto de variables estadísticas y la matriz de confusión, el árbol generado por este algoritmo fue mucho más profundo y contó con un mayor número de hojas; sin embargo, el tiempo

que le tomó al algoritmo la construcción del modelo fue mucho mayor, tomándole al algoritmo más de 90 segundos.

C. ALGORITMOS EM y K-MEANS

a) Algoritmos EM y K-Means – Eficiencia de los Algoritmos

Para la ejecución de los algoritmos se les solicitó a ambos formar 4 clústeres para la experimentación, este valor puede cambiar dependiendo de cada usuario y de la necesidad que se presente.

Cabe destacar que para ambas clusterizaciones no se tomó en cuenta el atributo "url", ya que este contempla valores diferentes en cada una de sus instancias y esto entorpece el proceso de clusterización.

Los resultados de la eficiencia de la clusterización arrojados por los algoritmos se resumen en la siguiente tabla comparativa:

Nombre del Algoritmo	Número de Clústeres	Instancias por clúster	Número de iteraciones	Tiempo tomado para generar el modelo (segundos)	Instancias no clasificadas
Algoritmo K-Means	4	0: 9502 (24%) 1: 10839 (27%) 2: 10912 (28%) 3: 8390 (21%)	7	10.76	0
Algoritmo EM	4	0: 7974 (20%) 1: 13622 (34%) 2: 10851 (27%) 3: 7196 (18%)		133.79	0

Tabla 6: Cuadro de comparación de la eficiencia de los algoritmos K-Means y EM para "Online News Popularity". Fuente propia.

b) Algoritmo EM – Análisis del Conocimiento

```
Number of clusters: 4


                                Cluster
Attribute                            0          1          2          3
                                   (0.2)     (0.34)     (0.27)     (0.18)
=============================================================================
 timedelta
  '(-inf-164.5]'                2086.2786  2904.3562  1971.9106  2972.4546
  '(164.5-338.5]'               2705.9091  2536.6907  2416.5843  2224.8159
  '(338.5-542.5]'               2334.0925  3234.6016   3148.221  1212.0848
  '(542.5-inf)'                  879.4832  4885.8662   3311.616   834.0346
  [total]                       8005.7634 13561.5147 10848.3319  7243.3901
 n_tokens_title
  '(-inf-8.5]'                  1149.9284  2716.5127  2504.3856  1003.1733
  '(8.5-10.5]'                  2751.0768   4676.774   3832.319  2378.8301
  '(10.5-11.5]'                 1455.1676  2370.0891  1820.1749  1318.5684
  '(11.5-inf)'                  2649.5906  3798.1389  2691.4524  2542.8182
  [total]                       8005.7634 13561.5147 10848.3319  7243.3901
 n_tokens_content
  '(-inf-246.5]'                2212.8623  7458.9083     1.0234   259.2061
  '(246.5-409.5]'               2381.5346  5788.3045   254.6051  1479.5558
  '(409.5-716.5]'               1897.0506   313.2655   4557.289  3156.3949
  '(716.5-inf)'                 1514.3158     1.0364  6035.4145  2348.2333
  [total]                       8005.7634 13561.5147 10848.3319  7243.3901
 n_unique_tokens
  '(-inf-0.539226]'             2995.0805   2347.676  9736.3791  4746.8644
  '(0.539226-inf)'              5008.6829 11211.8386  1109.9528  2494.5257
  [total]                       8003.7634 13559.5147 10846.3319  7241.3901
```

Figura 16: Fragmento de los resultados de clusterización del algoritmo EM para "Online News Popularity". Fuente: WEKA.

En el caso del algoritmo EM, este nos muestra todos los atributos del repositorio de datos junto a sus respectivas etiquetas, ya que al discretizarlos estos se volvieron atributos nominales. Como se puede apreciar en la imagen por cada etiqueta de atributo el algoritmo realiza un conteo de frecuencia que es asignado a cada clúster encontrado por el mismo, este valor puede contener decimales; este valor a su vez representa un porcentaje de probabilidad de pertenencia de cada instancia a su respectivo clúster. A través de estos resultados nos es posible determinar tendencias tales como, por ejemplo, que en un clúster que abarca el 34% de la totalidad de instancias (el clúster más grande), la mayor parte de las noticias cuentan con un título el cual se compone

de más de 11 palabras, sin embargo, dentro del mismo, la mayor parte de las noticias no cuentan con más de 246 palabras en su contenido.

El algoritmo no formó clústeres muy homogéneos y el valor "Log likelihood", que podríamos interpretar como el logaritmo de verosimilitud, arrojó un valor de -60.6625, por lo que podemos decir que este no fue un resultado ajustado, en otras palabras, no tiene un buen nivel de confiabilidad.

c) Algoritmo K-Means – Análisis del Conocimiento

Cluster centroids:

Attribute	Full Data (39643)	Cluster# 0 (9502)	1 (10839)	2 (10912)	3 (8390)
timedelta	'(-inf-164.5]'	'(-inf-164.5]'	'(-inf-164.5]'	'(542.5-inf)'	'(338.5-542.5]'
n_tokens_title	'(8.5-10.5]'	'(11.5-inf)'	'(11.5-inf)'	'(8.5-10.5]'	'(8.5-10.5]'
n_tokens_content	'(-inf-246.5]'	'(716.5-inf)'	'(-inf-246.5]'	'(409.5-716.5]'	'(246.5-409.5]'
n_unique_tokens	'(-inf-0.539226]'	'(-inf-0.539226]'	'(0.539226-inf)'	'(-inf-0.539226]'	'(0.539226-inf)'
n_non_stop_words	'(-inf-1]'	'(1-inf)'	'(-inf-1]'	'(1-inf)'	'(-inf-1]'
n_non_stop_unique_tokens	'(-inf-0.690476]'	'(-inf-0.690476]'	'(0.690476-inf)'	'(-inf-0.690476]'	'(0.690476-inf)'
num_hrefs	'(-inf-4.5]'	'(13.5-inf)'	'(-inf-4.5]'	'(4.5-7.5]'	'(7.5-13.5]'
num_self_hrefs	'(-inf-1.5]'	'(4.5-inf)'	'(2.5-4.5]'	'(2.5-4.5]'	'(-inf-1.5]'
num_imgs	'(0.5-1.5]'	'(0.5-1.5]'	'(-inf-0.5]'	'(0.5-1.5]'	'(0.5-1.5]'
num_videos	'(-inf-0.5]'	'(-inf-0.5]'	'(-inf-0.5]'	'(-inf-0.5]'	'(-inf-0.5]'

Figura 17: Fragmento de los resultados de clusterización del algoritmo K-Means para "Online News Popularity". Fuente: WEKA.

En este caso el algoritmo K-Means nos da como resultado los centroides de los clúster finales encontrados por el algoritmo, y es a través de estos que podemos determinar las tendencias; sin embargo, este no detalla el nivel de frecuencia de cada etiqueta del atributo objetivo. En el fragmento de los resultados mostrado en la Fig. 16 podemos ver los centroides de los 4 clústeres solicitados al algoritmo respecto a los primeros 10 atributos del repositorio de datos, así como el centroide de la población general o la totalidad de instancias; por ejemplo, en el tercer clúster, el cual representa la mayor parte de la población agrupando

10912 instancias, han predominado las noticias que cuentan con una cantidad superior a 542 palabras en su contenido, así como aquellas que tienen de 8 a 10 palabras en su título. Cabe mencionar que en este caso el segundo clúster más grande encontrado por el algoritmo, con 10839 instancias agrupadas, encontró la misma tendencia hallada por el algoritmo EM en su mayor clúster. Este algoritmo formó clústeres más homogéneos.

d) Algoritmos EM y K-Means – Conclusión: Ambos algoritmos, tanto EM como K-Means, lograron agrupar todas las instancias del repositorio de datos; sin embargo, el algoritmo K-Means logró formar clústeres más homogéneos con buena calidad de conocimiento, un mayor nivel de confiabilidad y en un tiempo muy inferior al que le tomó al algoritmo EM, siendo este 10 veces menor; y aunque el algoritmo EM muestra un resultado que involucra todas las etiquetas de cada atributo del repositorio de datos, lo que facilita un análisis más amplio tomando en cuenta un mayor número de factores, éste arrojó un logaritmo de verosimilitud muy bajo, lo que no da plena confianza de la totalidad de sus resultados.

D. ALGORITMO A-PRIORI

a) Algoritmo A-priori – Eficiencia del Algoritmo

En este caso se le indicó al algoritmo el número de reglas de asociación que se precisan, en nuestro caso para la experimentación se le solicitó 40; cabe destacar que este algoritmo trabaja únicamente con atributos

nominales, razón por la cual se discretizó toda la información antes de su aplicación.

Las reglas de asociación encontradas por el algoritmo fueron las siguientes:

```
 1.  is_weekend=0 34453 ==>  weekday_is_saturday=0 34453    conf:(1)
 2.  is_weekend=0 34453 ==>  weekday_is_sunday=0 34453    conf:(1)
 3.  weekday_is_sunday=0  is_weekend=0 34453 ==>  weekday_is_saturday=0 34453    conf:(1)
 4.  weekday_is_saturday=0  is_weekend=0 34453 ==>  weekday_is_sunday=0 34453    conf:(1)
 5.  weekday_is_saturday=0  weekday_is_sunday=0 34453 ==>  is_weekend=0 34453    conf:(1)
 6.  is_weekend=0 34453 ==>  weekday_is_saturday=0  weekday_is_sunday=0 34453    conf:(1)
 7.  data_channel_is_lifestyle=0  is_weekend=0 32746 ==>  weekday_is_saturday=0 32746    conf:(1)
 8.  data_channel_is_lifestyle=0  is_weekend=0 32746 ==>  weekday_is_sunday=0 32746    conf:(1)
 9.  data_channel_is_lifestyle=0  weekday_is_sunday=0  is_weekend=0 32746 ==>  weekday_is_saturday=0 32746    conf:(1)
10.  data_channel_is_lifestyle=0  weekday_is_saturday=0  is_weekend=0 32746 ==>  weekday_is_sunday=0 32746    conf:(1)
11.  data_channel_is_lifestyle=0  weekday_is_saturday=0  weekday_is_sunday=0 32746 ==>  is_weekend=0 32746    conf:(1)
12.  data_channel_is_lifestyle=0  is_weekend=0 32746 ==>  weekday_is_saturday=0  weekday_is_sunday=0 32746    conf:(1)
13.  data_channel_is_socmed=0  is_weekend=0 32447 ==>  weekday_is_saturday=0 32447    conf:(1)
14.  data_channel_is_socmed=0  is_weekend=0 32447 ==>  weekday_is_sunday=0 32447    conf:(1)
15.  data_channel_is_socmed=0  weekday_is_sunday=0  is_weekend=0 32447 ==>  weekday_is_saturday=0 32447    conf:(1)
16.  data_channel_is_socmed=0  weekday_is_saturday=0  is_weekend=0 32447 ==>  weekday_is_sunday=0 32447    conf:(1)
17.  data_channel_is_socmed=0  weekday_is_saturday=0  weekday_is_sunday=0 32447 ==>  is_weekend=0 32447    conf:(1)
18.  data_channel_is_socmed=0  is_weekend=0 32447 ==>  weekday_is_saturday=0  weekday_is_sunday=0 32447    conf:(1)
19.  is_weekend=0 34453 ==>  data_channel_is_lifestyle=0 32746    conf:(0.95)
20.  weekday_is_saturday=0  weekday_is_sunday=0 34453 ==>  data_channel_is_lifestyle=0 32746    conf:(0.95)
21.  weekday_is_saturday=0  is_weekend=0 34453 ==>  data_channel_is_lifestyle=0 32746    conf:(0.95)
22.  is_weekend=0 34453 ==>  data_channel_is_lifestyle=0  weekday_is_saturday=0 32746    conf:(0.95)
23.  weekday_is_sunday=0  is_weekend=0 34453 ==>  data_channel_is_lifestyle=0 32746    conf:(0.95)
24.  is_weekend=0 34453 ==>  data_channel_is_lifestyle=0  weekday_is_sunday=0 32746    conf:(0.95)
25.  weekday_is_saturday=0  weekday_is_sunday=0  is_weekend=0 34453 ==>  data_channel_is_lifestyle=0 32746    conf:(0.95)
26.  weekday_is_sunday=0  is_weekend=0 34453 ==>  data_channel_is_lifestyle=0  weekday_is_saturday=0 32746    conf:(0.95)
27.  weekday_is_saturday=0  is_weekend=0 34453 ==>  data_channel_is_lifestyle=0  weekday_is_sunday=0 32746    conf:(0.95)
28.  weekday_is_saturday=0  weekday_is_sunday=0 34453 ==>  data_channel_is_lifestyle=0  is_weekend=0 32746    conf:(0.95)
29.  is_weekend=0 34453 ==>  data_channel_is_lifestyle=0  weekday_is_saturday=0  weekday_is_sunday=0 32746    conf:(0.95)
30.  weekday_is_sunday=0 36906 ==>  data_channel_is_lifestyle=0 35017    conf:(0.95)
31.  weekday_is_saturday=0 37190 ==>  data_channel_is_lifestyle=0 35273    conf:(0.95)
32.  weekday_is_friday=0 33942 ==>  data_channel_is_lifestyle=0 32148    conf:(0.95)
33.  data_channel_is_socmed=0  weekday_is_sunday=0 34720 ==>  data_channel_is_lifestyle=0 32831    conf:(0.95)
34.  data_channel_is_socmed=0  weekday_is_saturday=0 35047 ==>  data_channel_is_lifestyle=0 33130    conf:(0.95)
35.  data_channel_is_socmed=0 37320 ==>  data_channel_is_lifestyle=0 35221    conf:(0.94)
36.  weekday_is_saturday=0 37190 ==>  data_channel_is_socmed=0 35047    conf:(0.94)
37.  is_weekend=0 34453 ==>  data_channel_is_socmed=0 32447    conf:(0.94)
38.  weekday_is_saturday=0  weekday_is_sunday=0 34453 ==>  data_channel_is_socmed=0 32447    conf:(0.94)
39.  weekday_is_saturday=0  is_weekend=0 34453 ==>  data_channel_is_socmed=0 32447    conf:(0.94)
40.  is_weekend=0 34453 ==>  data_channel_is_socmed=0  weekday_is_saturday=0 32447    conf:(0.94)
```

Figura 18: Reglas de asociación encontradas por A-priori en "Online News Popularity". Fuente: WEKA.

Las reglas fueron encontradas en, aproximadamente, 1 minuto.

b) Algoritmo A-priori – Análisis del Conocimiento

En este caso el repositorio de datos contó con dos grupos de atributos binarios, por lo que al momento de generar las reglas el algoritmo asoció a muchos de ellos, y si bien es correcto decir que las reglas son exactas con sus afirmaciones, éstas no nos bridan un gran conocimiento el cual nos permita tomar decisiones importantes; un claro ejemplo de ello es la

regla 1 que nos indica que cuando la noticia no se publicó un fin de semana entonces tampoco se publicó un sábado, información verdadera pero irrelevante; esto sucede con todas las reglas de asociación halladas en este repositorio de datos, las cuales el algoritmo considera las mejores dada las frecuencias asociadas a ellas.

c) ***Algoritmo A-priori – Conclusión:*** Las reglas de asociación halladas por el algoritmo fueron de mucha confiabilidad, siendo en su mayoría de 100%; sin embargo, la relevancia de estas fue ínfima, ninguna contó con el nivel de calidad necesario por cualquier usuario para su utilización en cualquier ámbito requerido tal como, por ejemplo, la toma de decisiones.

4.2. MARKETING BANCARIO (BANK MARKETING)

Este conjunto de datos está relacionado con las campañas de marketing directo de una empresa bancaria portuguesa. El conjunto de datos tiene 17 atributos y 4522 instancias.

4.2.1. DICCIONARIO DE DATOS

ID	*Atributo*	*Tipo*	*Descripción*
0	age	Numérico	Edad.
1	job	Categórico	Tipo de trabajo.
2	marital	Categórico	Estado marital.
3	education	Categórico	Nivel de educación.
4	balance	Numérico	Balance del cliente
5	default	Binario	¿Tiene crédito en mora?
6	housing	Binario	¿Tiene crédito para vivienda?
7	loan	Binario	¿Tiene crédito personal?
8	contact	Binario	Medio de contacto.
9	month	Categórico	Mes del último contacto.
10	day_of_week	Categórico	Día del contacto.
11	duration	Numérico	Duración del último contacto, en segundos.
12	campaign	Numérico	Número de contactos realizados al cliente durante esta campaña.
13	pdays	Numérico	Número de días que pasaron desde que se puso en contacto con el cliente por última vez de una campaña anterior.
14	previous	Numérico	Número de contactos realizados antes de esta campaña y para este cliente.
15	poutcome	Categórico	Resultado de la campaña de Marketing anterior.
16	y	Binario	¿El cliente se suscribio a un depósito a plazo?

Tabla 7: Diccionario de datos del repositorio de datos "Bank Marketing". Fuente propia.

4.2.2. PRE PROCESAMIENTO

Se utilizó el algoritmo **NumericToNominal** para transformar el atributo **"day_of_week"**, con el fin de que a cada día de la semana se le asigne una frecuencia y este no sea tomado como un simple valor numérico.

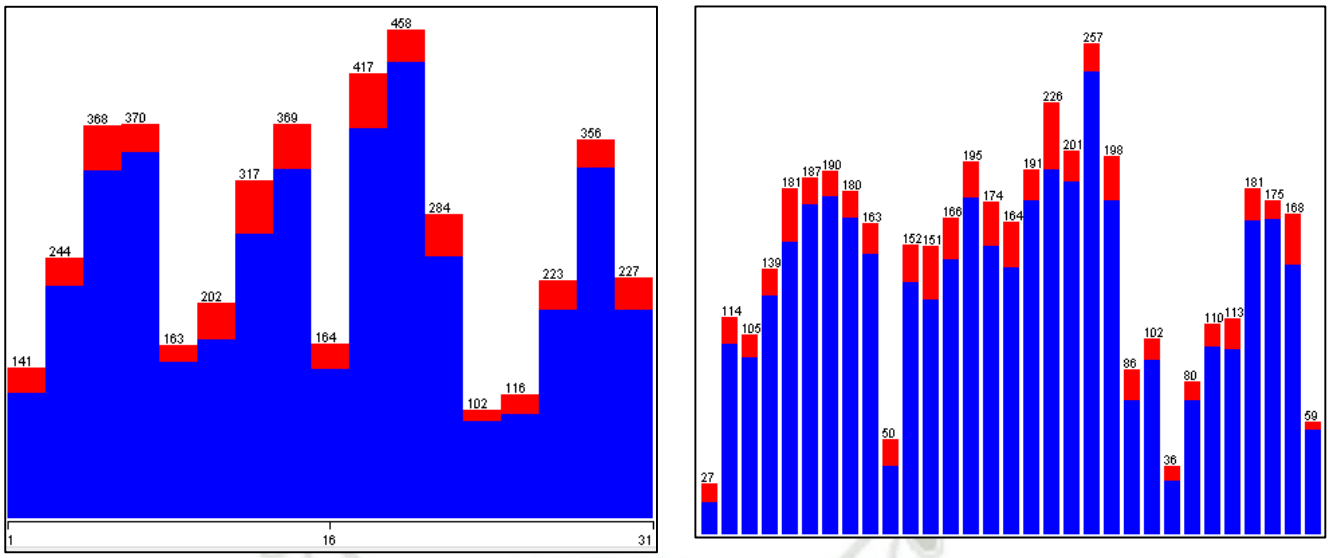

Figura 19: Atributo day_of_week como numérico y nominal. Fuente: WEKA.

Posteriormente se utilizó el algoritmo "Discretize" para discretizar los siguientes atributos numéricos del repositorio de datos: **age**, **balance**, **duration**, **campaign**, **pdays** y **previous**; todos fueron discretizados en 4 clases.

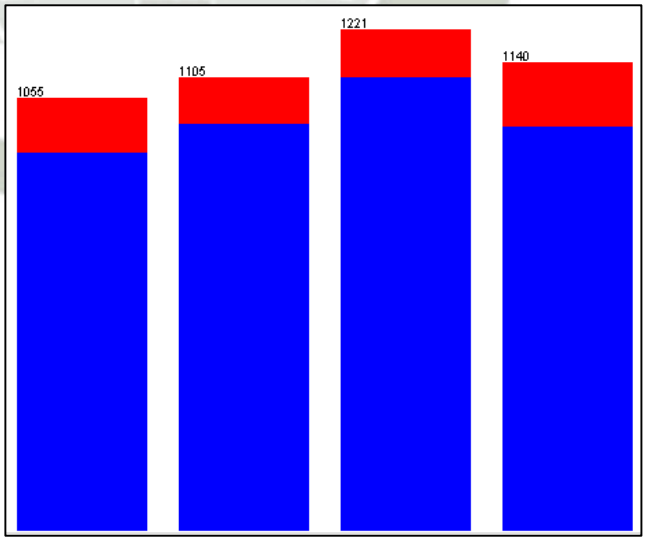

Figura 20: Ejemplo de discretización con el atributo "age" en el repositorio de datos Bank Marketing. Fuente: WEKA.

4.2.3. APLICACIÓN DE ALGORITMOS

En este caso el objetivo de la clasificación es predecir si un cliente se suscribirá a un depósito a plazo. El atributo objetivo dentro del conjunto de datos es "**y**".

A. ALGORITMO J48

a) Algoritmo J48 – Eficiencia del Algoritmo

```
Time taken to build model: 0.11 seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances        4064               89.8916 %
Incorrectly Classified Instances       457               10.1084 %
Kappa statistic                          0.2549
Mean absolute error                      0.1586
Root mean squared error                  0.2816
Relative absolute error                 77.7296 %
Root relative squared error             88.1927 %
Total Number of Instances             4521

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.993     0.823      0.903     0.993     0.946      0.829    no
                 0.177     0.007      0.767     0.177     0.287      0.829    yes
Weighted Avg.    0.899     0.729      0.887     0.899     0.87       0.829

=== Confusion Matrix ===

    a    b   <-- classified as
 3972   28 |    a = no
  429   92 |    b = yes
```

Figura 21: Resultados del algoritmo J48 en "Bank Marketing".
Fuente: WEKA.

Podemos ver que un 89.8916% de las instancias fueron correctamente clasificadas, mientras que un 10.1084% de las instancias fueron incorrectamente clasificadas. La tasa de verdaderos positivos es muy alta para la clase 1 y muy baja para la clase 2.

Podemos ver que la diagonal central de la matriz de confusión no tiene los valores más altos en su totalidad, ya que los falsos negativos superan por mucho a los verdaderos negativos.

b) Algoritmo J48 – Análisis del Conocimiento

Las ramas de conocimiento más relevantes encontradas por el algoritmo fueron las siguientes:

1. Se determinó que, cuando la duración del último contacto con un cliente fue de 104 a 185 segundos y el resultado de la campaña de marketing anterior fue desconocido o inexistente, 931 clientes no se suscribieron a un depósito a plazo; 24 de estas instancias fueron incorrectamente clasificadas. En este caso podemos interpretar que cuando el tiempo de contacto con el cliente es de aproximadamente 2 minutos y no se ha tenido contacto anteriormente con él (un potencial cliente nuevo), más de 900 clientes no se suscribieron; dado el número total de clientes este número no es muy elevado, sin embargo, se podría analizar si el tiempo de contacto es el adecuado.
2. Cuando la duración del último contacto con un cliente fue de 185 a 330 segundos y el resultado de la campaña de marketing anterior fue desconocido o inexistente, 892 clientes no se suscribieron a un depósito a plazo, 53 de estas instancias fueron incorrectamente clasificadas; mientras que si la campaña anterior falló no se suscribieron 133 clientes, 11 de estas instancias incorrectamente clasificadas; finalmente si la

campaña fue satisfactoria 52 clientes se suscribieron; 13 de estas instancias incorrectamente clasificadas. En este caso podemos interpretar que cuando el tiempo de contacto con el cliente es de 2 a 5 minutos aproximadamente el haber entablado contacto con dicho cliente anteriormente es un factor determinante, ya que si este no existió una gran cantidad de clientes no se suscribieron al depósito, mientras que si se tuvo contacto con el mismo en una campaña fallida este número redujo considerablemente, finalmente si fue satisfactoria hubo muy pocos clientes que no aceptaron dicho depósito.

3. Cuando la duración del último contacto fue superior a los 330 segundos y el resultado de la campaña de marketing anterior fue desconocido o inexistente, 917 clientes no se suscribieron a un depósito a plazo, 247 de estas instancias fueron incorrectamente clasificadas; mientras que si la campaña anterior falló 110 clientes no se suscribieron, 34 instancias incorrectamente clasificadas; si el resultado de la campaña de marketing anterior fue clasificado como “otros”, el mes en que se realizó el último contacto influyó, ya que el algoritmo determinó que la mayor parte de los clientes que no se suscribieron al depósito a plazo fueron contactados por última vez en el mes de mayo; finalmente, si el resultado de la campaña de marketing anterior fue satisfactorio, 44 clientes se suscribieron al depósito a plazo, 13 instancias incorrectamente clasificadas. En este caso podemos interpretar que cuando la duración del contacto con un

cliente es muy elevada, superando los 6 minutos en la mayoría de casos, el resultado de una campaña anterior fue determinante siendo, como en casos anteriores, que con un mejor resultado anterior el número de cliente que aceptaron el depósito fue superior, siendo en este caso la inexistencia de una campaña el peor escenario posible. Así mismo, de ser un resultado que no coincide con ninguna de las categorías (por ejemplo una campaña por aniversario o una promoción especial), siendo este clasificado como "otros", el mes en que se realizó dicha campaña fue influyente, ya que en este caso se detectó que evento realizados en el mes de Mayo fueron los que atrajeron a la menor cantidad de clientes.

c) Algoritmo J48 – Conclusión

El algoritmo J48 no mostró los mejores resultados al momento de clasificar todas las instancias, teniendo en cuenta que para el atributo objetivo "y", que representa si un cliente se suscribió o no a un depósito a plazo, a pesar de encontrarse un gran número de verdaderos positivos, la cantidad de falsos negativos fue muy alta en comparación a los verdaderos negativos; respecto a la calidad del conocimiento generado, el algoritmo tomó primordialmente el atributo **duration** (duración en segundos del último contacto con el cliente) como primer nivel de clasificación y **poutcome** (resultado de la campaña de marketing anterior) como segundo nivel, dejando de lado a la mayoría de atributos; aunque esto podría ser debido a que no resultaron ser relevantes para la

clasificación. En cuanto a la eficiencia del algoritmo, la tasa de verdaderos positivos fue muy alta en comparación a los falsos positivos y a los falsos negativos (aun siendo estos más que los verdaderos negativos); al algoritmo le tomó 0.11 segundos generar el modelo.

B. ALGORITMO J48-GRAFT

a) Algoritmo J48-Graft – Eficiencia del Algoritmo

```
Time taken to build model: 0.24 seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances        4064               89.8916 %
Incorrectly Classified Instances       457               10.1084 %
Kappa statistic                          0.2549
Mean absolute error                      0.1586
Root mean squared error                  0.2816
Relative absolute error                 77.7296 %
Root relative squared error             88.1927 %
Total Number of Instances             4521

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.993    0.823      0.903     0.993     0.946      0.829    no
                 0.177    0.007      0.767     0.177     0.287      0.829    yes
Weighted Avg.    0.899    0.729      0.887     0.899     0.87       0.829

=== Confusion Matrix ===

    a    b   <-- classified as
 3972   28 |    a = no
  429   92 |    b = yes
```

Figura 22: Resultados del algoritmo J48-Graft en "Bank Marketing". Fuente: WEKA.

Podemos ver que un 89.8916% de las instancias fueron correctamente clasificadas, mientras que un 10.1084% de las instancias fueron incorrectamente clasificadas. La tasa de verdaderos positivos es muy alta para la clase 1 y muy baja para la clase 2. Podemos ver que la

diagonal central de la matriz de confusión no tiene los valores más altos en su totalidad, ya que los falsos negativos superan por mucho a los verdaderos negativos. Cabe destacar que en este caso en ambos algoritmos, tanto J48 como J48-Graft, obtuvieron valores idénticos.

b) Algoritmo J48-Graft – Análisis del Conocimiento

Las ramas de conocimiento más relevantes encontradas por el algoritmo fueron las siguientes:

1. Se determinó que cuando la duración del último contacto con un cliente fue de 104 a 185 segundos y el resultado de la campaña de marketing anterior fue desconocido o inexistente, 931 clientes no se suscribieron a un depósito a plazo, 24 de estas instancias no fueron correctamente clasificadas; en cambio si la campaña anterior falló 136 clientes no se suscribieron a un depósito a plazo, 8 instancias incorrectamente clasificadas. En este caso podemos interpretar que cuando el tiempo de contacto con un usuario nuevo es de aproximadamente 2 minutos una gran cantidad de ellos declinaran la oferta de un depósito, sin embargo, de existir un contacto de una campaña anterior, aun si ésta no fue exitosa, la probabilidad de que el cliente rechace el depósito reduce considerablemente.
2. Cuando la duración del último contacto con un cliente fue de 104 a 185 segundos, la campaña de marketing anterior fue exitosa, el nivel de educación del cliente es “secundaria”, el número de contactos realizados al cliente en la campaña fue de 1 o 2, el

cliente no contó con crédito para vivienda pero si crédito personal y no se realizaron contactos antes de esta campaña con el cliente 63 clientes no se suscribieron al crédito a plazo; 3 instancias fueron incorrectamente clasificadas. En este caso podemos interpretar que cliente nuevos con los que la empresa nunca había tenido contacto, educación "secundaria" con ingresos moderados (dados sus créditos) y a los cuáles solo se les contactó máximo un par de veces son buenos candidatos para adquirir depósitos a plazo fijo, dado que la cantidad de cliente con éstas características que lo rechazaron fue muy baja.

3. Cuando la duración del último contacto con un cliente fue de 104 a 185 segundos, la campaña de marketing anterior fue exitosa, el nivel de educación del cliente es "terciario" o "universitario" y el medio de contacto fue desconocido, 68 clientes se suscribieron al depósito a plazo; en cambio si el medio de contacto fue teléfono o celular y no hubo contactos realizados a este cliente antes de esta campaña, 294 clientes no se suscribieron, 8 instancias incorrectamente clasificadas; si paso más de un día desde que se puso en contacto con el cliente por última vez 294 clientes no se suscribieron, 8 instancias incorrectamente clasificadas; además si el trabajo del cliente no está asociado al ámbito empresarial y tiene crédito de vivienda, en 189 ocasiones se suscribieron al depósito a plazo; 11 de estas instancias fueron incorrectamente clasificadas. En este caso podemos interpretar que clientes con un nivel de estudios "superior" cuyos contactos

fueron realizados a través de otros medios ajenos al teléfono o celular y los cuales duraron aproximadamente 2 minutos no suelen suscribirse a los depósitos ofrecidos por la empresa; también podemos determinar que clientes con estas características con los cuáles se pone en contacto como máximo dos días después de su último contacto son más propensos a suscribirse. Por otro lado, suelen ser más los clientes con éstas características cuyos trabajos estén relacionados al ámbito empresarial e ingresos superiores los que se suscriben a dicho depósito.

4. Se encontró que cuando la duración del último contacto con un cliente fue de 185 a 330 segundos y el resultado de la campaña de marketing anterior fue desconocido o inexistente, 892 no suscribieron a un depósito a plazo, 53 instancias incorrectamente clasificadas; si la campaña falló 133 clientes no se suscribieron, 21 instancias incorrectamente clasificadas; si la campaña de marketing anterior fue exitosa y se desconoce el medio de contacto con el cliente, 341 no se suscribieron al depósito a plazo, 2 instancias incorrectamente clasificadas; si el medio de contacto con el cliente fue celular o teléfono y no se contactó con el cliente anteriormente 892 cliente no se suscribieron al depósito a plazo, 53 instancias incorrectamente clasificadas; además el algoritmo también determinó que en días 21, 23 o 29 del mes existe un buen número de cliente que no se suscribieron al depósito a plazo. En este caso podemos interpretar que dada

una duración de contacto de aproximadamente 2 minutos el resultado de una campaña anterior con el cliente fue influyente, dado que si la campaña fue inexistente hubo un gran número de personas que no se suscribieron, mientras que si la campaña fue exitosa hubo un menor número de rechazos, adicionalmente si el medio de contacto fue teléfono o celular con cliente nuevos estos fueron más propensos a rechazar el depósito, además de que en general clientes con los que se contactó en los últimos días del mes tiene un índice de rechazo más elevado.

5. Cuando la duración del último contacto con un cliente fue superior a los 330 segundos y el resultado de la campaña de marketing anterior fue desconocido o inexistente, 917 clientes no se suscribieron a un depósito a plazo, 247 instancias fueron incorrectamente clasificadas; si la campaña anterior falló 110 clientes no se suscribieron, 34 instancias incorrectamente clasificadas; si la campaña fue clasificada como “otros” el mes en que se realizó el último contacto intervino, ya que en los meses de mayo y abril hubo un mayor número de clientes que decidieron no suscribirse al depósito a plazo; cuando el resultado de la campaña anterior fue positivo o satisfactorio y el medio de contacto con el usuario fue desconocido, 323 clientes no se suscribieron al depósito a plazo, 56 instancias incorrectamente clasificadas; si el medio de contacto con el usuario fue teléfono o celular y antes de esta campaña nunca se había contactado con el cliente, en 917 ocasiones el cliente no se suscribió al depósito

a plazo, 247 de estas instancias fueron incorrectamente clasificadas. En este caso podemos interpretar que, como en ocasiones anteriores, que dependiendo de si antes se tuvo contacto con un usuarios y las condiciones en se tuvo dicho contacto, éste será más propenso o no a suscribirse al depósito, así mismo, el uso de medios directos como teléfono y celular con clientes nuevos hacen que estos sean menos propensos a suscribirse en una entidad desconocida; además, en campañas espontáneas clasificadas como "otro" realizadas en los meses de mayo y abril fueron las que tuvieron la menor aceptación por parte de los clientes.

c) Algoritmo J48-Graft – Conclusión

El algoritmo J48-Graft, al igual que el algoritmo J48, no mostró los mejores resultados al momento de clasificar todas las instancias, teniendo en cuenta el atributo objetivo "y"; sin embargo, este algoritmo generó un árbol de clasificación más grande, así como un mayor número de hojas; esto permitió obtener un conocimiento más detallado ya que de esta manera se han empleado un mayor número de atributos al momento de la clasificación de las instancias. En cuanto a la eficiencia del algoritmo, la tasa de verdaderos positivos fue muy alta en comparación a los falsos positivos y a los falsos negativos (aun siendo estos más que los verdaderos negativos), al igual que el algoritmo J48 como se mencionó anteriormente; al algoritmo le tomó 0.62 segundos generar el modelo.

C. ALGORITMO EM y K-MEANS

a) Algoritmos EM y K-Means – Eficiencia de los Algoritmos

Para la ejecución de los algoritmos se les solicitó a ambos formar 4 clústeres para la experimentación, este valor puede cambiar dependiendo de cada usuario y de la necesidad que se presente.

Los resultados de la eficiencia de la clusterización arrojados por los algoritmos se resumen en la siguiente tabla comparativa:

Nombre del Algoritmo	Número de Clústeres	Instancias por clúster	Número de iteraciones	Tiempo tomado para generar el modelo (segundos)	Instancias no clasificadas
Algoritmo K-Means	4	0: 1527 (34%) 1: 553 (12%) 2: 1010 (22%) 3: 1431 (32%)	4	0.09	0
Algoritmo EM	4	0: 1063 (24%) 1: 816 (18%) 2: 1575 (35%) 3: 1067 (24%)		2.93	0

Tabla 8: Cuadro de comparación de la eficiencia de los algoritmos K-Means y EM para "Bank Marketing". Fuente propia.

b) Algoritmo EM – Análisis del Conocimiento

```
Number of clusters: 4


                     Cluster
Attribute                 0         1         2         3
                     (0.24)    (0.18)    (0.35)    (0.23)
============================================================
age
  '(-inf-32.5]'    200.0547       181  383.2467  294.6986
  '(32.5-38.5]'    176.1791       227  380.5305  325.2904
  '(38.5-48.5]'    278.8846       207  480.0441  259.0713
  '(48.5-inf)'      425.615       205  329.3909  183.9941
  [total]         1080.7335  819.9999 1573.2122 1063.0544
job
  unemployed       44.3739        22   36.2782   29.3479
  services        152.7106        63  197.3614     7.928
  management       30.5873       187   227.428  527.9847
  blue-collar     300.3663       153  493.5416    3.0922
  self-employed    25.9591        29   70.1268   61.9141
  technician      118.5102       142  226.5757  284.9142
  entrepreneur     46.0065        24   57.6791   44.3144
  admin.          166.6722       109  166.7174   39.6104
  student           6.2403        23    17.219   41.5406
  housemaid        55.9055        19   30.3882   10.7063
  retired         126.6174        50   47.3819   10.0007
  unknown          14.7843         7   10.5147    9.7009
  [total]         1088.7335  827.9999 1581.2122 1071.0544
marital
  married         757.8888  492.9999  998.5873   551.524
  single          170.8564       240   380.205  408.9386
  divorced        150.9883        86    193.42  101.5918
  [total]         1079.7335  818.9999 1572.2122 1062.0544
education
  primary         280.3453        99   296.825    5.8297
  secondary       718.2862  415.9999  883.2767  292.4371
  tertiary         44.0599       267  306.8835  736.0567
  unknown           38.042        38    86.227    28.731
  [total]         1080.7335  819.9999 1573.2122 1063.0544
default
  no              1054.2525  811.9999 1542.2461 1040.5015
  yes               24.4809         6   28.9661    20.553
  [total]         1078.7335  817.9999 1571.2122 1061.0544
```

Figura 23: Fragmento de los resultados de clusterización del algoritmo EM para "Bank Marketing". Fuente: WEKA.

Como se ha visto antes el algoritmo EM nos muestra todos los atributos del repositorio de datos junto a sus respectivas etiquetas, en el caso de

varios atributos antes numéricos el número de estas etiquetas varía según lo solicitado por el usuario gracias a la discretización. En este caso se han determinado tendencias como que en el clúster más grande, por ende aquel que representa la mayor parte de la población, la mayor parte de los clientes cuenta con una edad de entre 38 y 48 años, su trabajo está relacionado al trabajo manual, particularmente a la industria o que son personas casadas.

El algoritmo formó clústeres relativamente homogéneos y el valor "Log likelihood" (logaritmo de verosimilitud) arrojó un valor de -17.6614, por lo que podemos decir que este no fue un resultado muy ajustado pero tampoco muy alejado de ello, en otras palabras, el nivel de confiabilidad no es bueno pero tampoco es pésimo.

c) Algoritmo K-Means – Análisis del Conocimiento

```
Number of iterations: 4
Within cluster sum of squared errors: 28891.0
Missing values globally replaced with mean/mode

Cluster centroids:
                                        Cluster#
Attribute                 Full Data            0            1            2            3
                            (4521)       (1527)       (553)       (1010)       (1431)
=============================================================================================
age                     '(38.5-48.5]' '(48.5-inf)' '(38.5-48.5]' '(32.5-38.5]' '(38.5-48.5]'
job                       management   technician  blue-collar   management  blue-collar
marital                      married      married      married      married      married
education                  secondary    secondary    secondary     tertiary    secondary
default                           no           no           no           no           no
balance           '(443.5-1478.5]' '(1478.5-inf)' '(443.5-1478.5]' '(443.5-1478.5]' '(-inf-68.5]'
housing                          yes           no          yes           no          yes
loan                              no           no           no           no           no
contact                    cellular     cellular     cellular     cellular      unknown
day                              20           18            6           21           20
month                           may          aug          may          aug          may
duration          '(-inf-104.5]' '(330.5-inf)' '(185.5-330.5]' '(185.5-330.5]' '(-inf-104.5]'
campaign            '(-inf-1.5]'   '(1.5-2.5]'   '(-inf-1.5]'   '(-inf-1.5]'   '(-inf-1.5]'
pdays                 '(-inf-0]'    '(-inf-0]' '(282.5-inf)'    '(-inf-0]'    '(-inf-0]'
previous            '(-inf-0.5]'  '(-inf-0.5]'   '(1.5-3.5]'  '(-inf-0.5]'  '(-inf-0.5]'
poutcome                unknown      unknown      failure      unknown      unknown
y                            no           no           no           no           no
```

Figura 24: Resultados de clusterización del algoritmo K-Means para "Bank Marketing". Fuente: WEKA.

El algoritmo K-Means nos da como resultado los centroides de los clúster finales encontrados por el algoritmo. En el resultado mostrado en la Fig. 24 podemos ver los centroides de los 4 clústeres solicitados al algoritmo, así como el centroide de la población general o la totalidad de instancias; por ejemplo, el primer clúster, el cual representa la mayor parte de la población agrupando 1527 instancias, nos indica que han predominado aquellos cliente que cuentan con una edad superior a los 48 años (cabe destacar que el resto de clústeres indicaron que la edad predominante fue de 38 a 48 años), cuenta con una carrera técnica, son personas casadas, su nivel de estudios es "secundario", no tienen crédito en mora, el balance del cliente es superior a los 1478, no cuenta con crédito de vivienda, no cuenta con crédito personal, el medio de contacto con el cliente fue "celular", se le contacto por última vez a mediados del mes de agosto, la duración del último contacto fue superior a los 330 segundos, se realizó de 1 a 2 contactos con el cliente durante la campaña, no se realizaron contactos anteriores con este cliente por ende el resultado de la última campaña fue categorizado como desconocido o inexistente, y finalmente la mayoría no se suscribió al depósito a plazo. Este algoritmo formó clústeres menos homogéneos que EM.

d) ***Algoritmos EM y K-Means – Conclusión:*** Ambos algoritmos, tanto EM como K-Means, lograron agrupar todas las instancias del repositorio de datos "Bank Marketing"; en este caso el algoritmo EM logró formar clústeres más homogéneos, aunque estos no variaron mucho respecto a los formados por K-Means: sin embargo, el algoritmo K-Means logró un

mayor nivel de confiabilidad y en un tiempo muy inferior al que le tomó al algoritmo EM, siendo más de 30 veces menor; en este caso el algoritmo EM muestra un resultado con clústeres más homogéneos, además involucra todas las etiquetas de cada atributo del repositorio de datos facilitando un análisis más amplio, sin embargo, el tiempo que le tomó al algoritmo clusterizar las instancias fue más de 30 veces mayor que el tomado por K-Means, si bien este valor con el número de instancias con las que se experimentó no es un gran cambio significativo, al trabajarse con bases de datos inmensamente mayores el tiempo tomado será un aspecto bastante negativo; cabe destacar que el algoritmo EM también arrojó un logaritmo de verosimilitud muy bajo, lo que no da plena confianza de la totalidad de sus resultados.

D. ALGORITMO A-PRIORI

a) Algoritmo A-priori – Eficiencia del Algoritmo

Para este repositorio de datos "Bank Marketing", se le solicitó al algoritmo encontrar las 10 mejores reglas de asociación.

```
Best rules found:

 1. previous='(-inf-0.5]' 3705 ==> pdays='(-inf-0]' 3705    conf:(1)
 2. pdays='(-inf-0]' 3705 ==> previous='(-inf-0.5]' 3705    conf:(1)
 3. poutcome=unknown 3705 ==> pdays='(-inf-0]' 3705    conf:(1)
 4. pdays='(-inf-0]' 3705 ==> poutcome=unknown 3705    conf:(1)
 5. poutcome=unknown 3705 ==> previous='(-inf-0.5]' 3705    conf:(1)
 6. previous='(-inf-0.5]' 3705 ==> poutcome=unknown 3705    conf:(1)
 7. previous='(-inf-0.5]' poutcome=unknown 3705 ==> pdays='(-inf-0]' 3705    conf:(1)
 8. pdays='(-inf-0]' poutcome=unknown 3705 ==> previous='(-inf-0.5]' 3705    conf:(1)
 9. pdays='(-inf-0]' previous='(-inf-0.5]' 3705 ==> poutcome=unknown 3705    conf:(1)
10. poutcome=unknown 3705 ==> pdays='(-inf-0]' previous='(-inf-0.5]' 3705    conf:(1)
```

Figura 25: Reglas de asociación encontradas por A-priori en "Bank Marketing". Fuente: WEKA.

b) Algoritmo A-priori – Análisis del Conocimiento

Como se vio anteriormente, el algoritmo halla las mejores reglas de asociación en base a las frecuencias relacionadas. En este caso de manera similar a lo ocurrido con el repositorio de datos "Online News Popularity", "Bank Marketing" contó con 3 atributos que constantemente se van a relacionar, estos son previous (número de contactos realizados a un cliente antes de la presente campaña), pdays (número de días que han pasado desde que se puso en contacto con un cliente por última vez) y poutcome (resultado de la campaña de marketing); esto es debido a que si una campaña anterior es desconocida o inexistente no se pudieron haber hecho contactos previos con un cliente, o al menos estos no fueron registrados; así mismo, si no hubo contactos previos con un cliente el número de días desde el último contacto es 0. Por ende las reglas encontradas por el algoritmo en este repositorio de datos resultaron irrelevantes.

c) Algoritmo A-priori – Conclusión: Las reglas de asociación halladas por el algoritmo fueron de mucha confiabilidad, siendo todas del 100%; sin embargo, debido a la relación que 3 de sus atributos guardan entre sí hizo que el conocimiento generado por el algoritmo fuera irrelevante.

4.3. CASO DE ESTUDIO

Con el fin de probar los algoritmos seleccionados y analizados a lo largo de la investigación se utilizará una base de datos asociada a los trámites de compra de insumos médicos realizados por EsSalud durante los años 2015 y 2016; la base de datos cuenta con 39568 instancias.

4.3.1. DICCIONARIO DE DATOS

ID	*Atributo*	*Tipo*	*Descripción*
0	Doc.compras	Categórico	Serie de números que representan la orden de compra a la que pertenece un artículo.
1	Posición	Numérico	Posición del artículo dentro de la orden de compra.
2	Ind.borrado	Binario	Indica si dicho artículo fue borrado de la orden de compra.
3	Ce.suministrad.	Categórico	Código que representa el centro al que le fue suministrado el artículo de darse el caso.
4	Centro	Categórico	Código que representa el centro que realiza la orden de compra.
5	Almacén	Categórico	Código que representa el almacén que realiza la orden de compra.
6	N° necesidad	Categórico	Serie de número y caracteres que representa el documento de necesidad asociado a la orden de compra.
7	Grupo-compras	Categórico	Código que representa quién realizo la orden de compra (200: Lima, 100: Arequipa, 120: Delegado de Lima a Arequipa)
8	Proveedor	Categórico	Código que representa al proveedor del artículo.
9	Cantidad pedido	Numérico	Número que representa la cantidad del producto solicitado.
10	UM de pedido	Categórico	Unidad de medida del artículo.
11	Material	Categórico	Código que representa al artículo.
12	Txt.brv.	Categórico	Descripción del artículo.
13	Fecha entrega	Categórico	Fecha prevista para la entrega del artículo.
14	Valor bruto	Numérico	Costo total de la compra del artículo.
15	Contrato	Categórico	Código que representa el contrato por la compra del artículo.
16	Pos.contrato	Numérico	Posición del artículo dentro del contrato de compra.
17	Grupo artículos	Categórico	Categoría a la que pertenece el artículo.
18	Entregado	Numérico	Cantidad entregada. También representa si se entrego dicho artículo.
19	Sol.pedido	Categórico	Código que representa la solicitud de pedido.
20	Pos.sol.pedido	Categórico	Posición del artículo en la solicitud de pedido.
21	Fecha de pedido	Categórico	Fecha en la que se realizó el pedido.

Tabla 9: Diccionario de datos de la base de datos de adquisición de insumos médicos de EsSalud de los años 2015 y 2016. Fuente propia.

4.3.2. PRE PROCESAMIENTO

Como lo visto en el análisis de repositorios de datos experimentales, existen variables que son tratadas por WEKA como numéricas cuando estas representan categorías o códigos; atributos como ***Posición***, la cual representa la posición de un artículo dentro de una orden de compra.

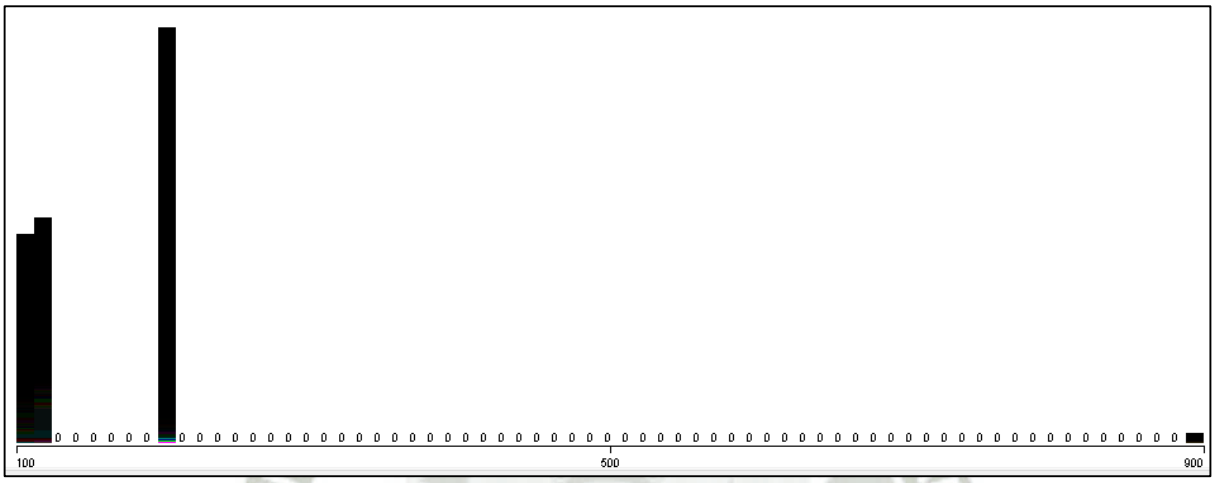

Figura 26: Atributo "Posición" tratado como numérico. Fuente: WEKA.

Para lo cual se aplicó el algoritmo NumericToNominal para transformar los siguientes atributos numéricos a nominales: ***Almacén,*** ***Grupo-compras*** y ***Grupo artículos***, quedando de la siguiente manera:

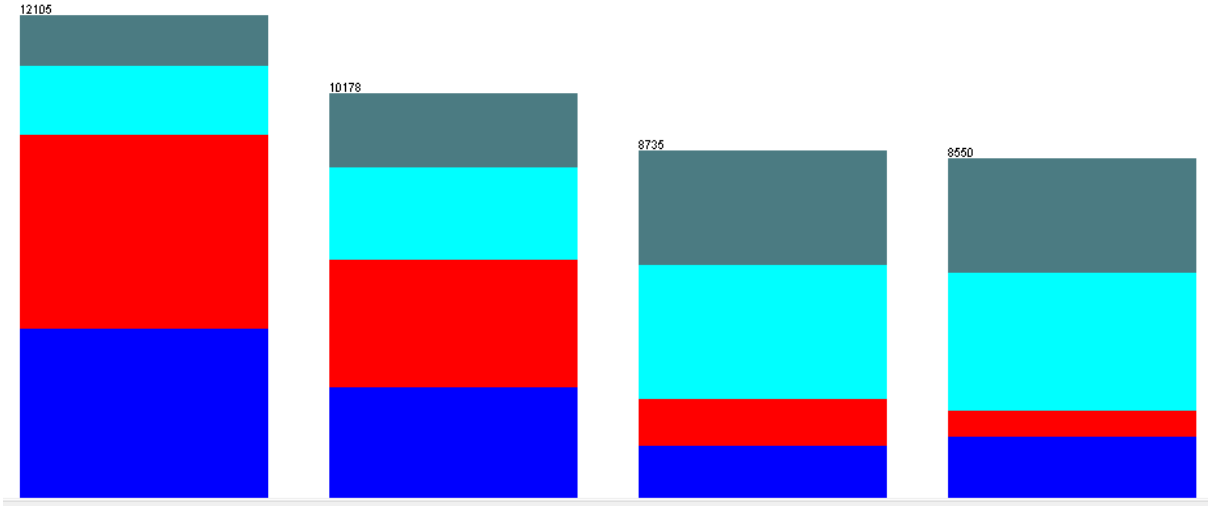

Figura 27: Atributo "Posición" tratado como nominal. Fuente: WEKA.

Finalmente se procedió a discretizar en 4 grupos de frecuencia los siguientes atributos: ***Posición***, ***Material***, ***Pos.contrato***, ***Pos.sol.pedido*** y ***Fecha de pedido***.

4.3.3. APLICACIÓN DE ALGORITMOS

En este caso nuestro objetivo será predecir las tendencias de compra de insumos médicos del Hospital Nacional Carlos Alberto Seguin Escobedo – EsSalud (Arequipa) basándonos en la fecha del año en la que nos encontremos.

A. ALGORITMO J48

```
Time taken to build model: 0.76 seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances       39478               99.7725 %
Incorrectly Classified Instances        90                0.2275 %
Kappa statistic                          0.997
Mean absolute error                      0.002
Root mean squared error                  0.0319
Relative absolute error                  0.5452 %
Root relative squared error              7.3838 %
Total Number of Instances            39568

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.997     0.001      0.996     0.997     0.996      1        '(-inf-42464.5]'
                 0.996     0.001      0.996     0.996     0.996      0.999    '(42464.5-42619.5]'
                 0.999     0          0.999     0.999     0.999      1        '(42619.5-42670.5]'
                 0.999     0          1         0.999     1          1        '(42670.5-inf)'
Weighted Avg.    0.998     0.001      0.998     0.998     0.998      1

=== Confusion Matrix ===

     a     b     c     d   <-- classified as
  9872    31     1     1 |     a = '(-inf-42464.5]'
    37  9850     3     0 |     b = '(42464.5-42619.5]'
     1    11 10892     0 |     c = '(42619.5-42670.5]'
     0     1     4  8864 |     d = '(42670.5-inf)'
```

Figura 28: Resultados del algoritmo J48 en la base de datos de EsSalud.
Fuente: WEKA.

Podemos ver que un 99.7725% de las instancias fueron correctamente clasificadas, mientras que un 0.2275% de las instancias fueron incorrectamente clasificadas. La tasa de verdaderos positivos fue muy alta para las 4 clases del atributo.

También podemos ver que la diagonal central de la matriz es muy superior a sus valores laterales, por lo que podemos decir que la clasificación es confiable.

a) Análisis de conocimiento

Un ejemplo del conocimiento hallado por el algoritmo fue el siguiente:

1. Se determinó que ciertos grupos de insumos médicos suelen ser más frecuentemente solicitados, de tal manera que la fecha de entrega sea fijada entre el 2 de Mayo y el 8 de Setiembre, por ejemplo insumos asociados a la categoría "2010" son extremadamente requeridos durante este periodo, ya que durante los años 2015 y 2016 fueron adquiridos más de 2300 insumos pertenecientes a esta categoría; en cambio, únicamente fueron adquiridos 8 insumos asociados a la categoría "1060" entre dichas fechas.

B. ALGORITMO J48-GRAFT

```
Time taken to build model: 23.44 seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances       39478               99.7725 %
Incorrectly Classified Instances        90                0.2275 %
Kappa statistic                          0.997
Mean absolute error                      0.002
Root mean squared error                  0.0319
Relative absolute error                  0.5452 %
Root relative squared error              7.3838 %
Total Number of Instances            39568

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.997    0.001      0.996     0.997     0.996      1        '(-inf-42464.5]'
                 0.996    0.001      0.996     0.996     0.996      0.999    '(42464.5-42619.5]'
                 0.999    0          0.999     0.999     0.999      1        '(42619.5-42670.5]'
                 0.999    0          1         0.999     1          1        '(42670.5-inf)'
Weighted Avg.    0.998    0.001      0.998     0.998     0.998      1

=== Confusion Matrix ===

     a     b     c     d   <-- classified as
  9872    31     1     1 |     a = '(-inf-42464.5]'
    37  9850     3     0 |     b = '(42464.5-42619.5]'
     1    11 10892     0 |     c = '(42619.5-42670.5]'
     0     1     4  8864 |     d = '(42670.5-inf)'
```

Figura 29: Resultados del algoritmo J48-Graft en la base de datos de EsSalud. Fuente: WEKA.

Podemos ver que el algoritmo J48-Graft mostró los mismos resultados que J48 en cuanto a instancias correcta e incorrectamente clasificadas, tasa de verdaderos positivos para las clases del atributo y en la matriz de confusión; sin embargo, contó con dos diferencias muy importantes: el tiempo de ejecución del algoritmo fue más de 20 veces mayor, pero fue capaz de generar un árbol de clasificación más profundo el cual utilizó de manera más eficiente cada uno de los atributos de la base de datos, algo que el algoritmo J48 no realizó.

a) Análisis de conocimiento

Un ejemplo de conocimiento hallado por el algoritmo fue el siguiente:

1. Se determinó que entre los meses de Abril y Setiembre se adquirió poca cantidad de insumos pertenecientes a la categoría "2040", sin embargo dichos productos suelen ser muy frecuentemente solicitados entre dichas fechas para que su fecha de entrega este programada entre Octubre y Noviembre del mismo año, siendo el proveedor identificado con el código "5126734E7" al que comúnmente le es solicitado este tipo de insumo, siendo también el que comúnmente está asociado a entregas fuera del plazo fijado.

C. ALGORITMO EM Y K-MEANS

Al igual que para las pruebas en los repositorios de datos, se les solicitó a ambos algoritmos formar 4 clústeres para las pruebas. Los resultados de la eficiencia de la clusterización arrojados por los algoritmos se resumen en la siguiente tabla comparativa:

Nombre del Algoritmo	Número de Clústeres	Instancias por clúster	Número de iteraciones	Tiempo tomado para generar el modelo (segundos)	Instancias no clasificadas
Algoritmo K-Means	4	0: 9971 (25%) 1: 9427 (24%) 2: 10509 (27%) 3: 9661 (24%)	7	2.46	0
Algoritmo EM	4	0: 9712 (25%) 1: 5465 (14%) 2: 10829 (27%) 3: 13562 (34%)		59.62	0

Tabla 10: Cuadro comparativo de la eficiencia de los algoritmos K-Means y EM en la base de datos de EsSalud. Fuente: WEKA.

a) Análisis de conocimiento

Como vimos en la etapa de experimentación el algoritmo K-Means nos da como resultado los centroides de los clúster finales encontrados por el algoritmo. En este caso podemos ver que el tercer clúster que representa la mayor parte de la población, con más de 10509 instancias, nos indica cosas como que la mayor cantidad de pedidos son realizados entre el 6 de setiembre y el 27 de octubre, la mayor parte de los insumos solicitados pertenecen a la categoría "1025", el proveedor más solicitado es al que le corresponde al código "472686E7", la solicitud de compra es mayormente hecha desde Lima, etc. Los clústeres formados por este algoritmo fueron muy homogéneos.

Por otro lado el algoritmo EM nos muestra todas las etiquetas de cada atributo, lo que nos da un resultado más detallado pero más complejo de interpretar. En este caso el clúster más grande fue el cuarto, agrupando más de 13000 instancias, dicho clúster nos indica cosas como que la mayor cantidad de solicitudes de compra son realizadas desde Lima, la fecha de entrega más común para los insumos suele ser a fines de Noviembre, la mayor parte de los insumos pertenecen a la categoría "1025", la fecha más común para realizar pedidos es a principios de Setiembre, etc. Cabe destacar que una vez interpretar los resultados son iguales o muy similares entre ambos algoritmos, lo que nos indica que los resultados hallados

tienen un alto nivel de confiabilidad. Los clústeres formados por este algoritmo fueron menos homogéneos que los formados por K-Means.

D. ALGORITMO A-PRIORI

Se le solicitó al algoritmo encontrar 40 reglas de asociación, como se mencionó anteriormente el número de reglas depende de la necesidad del usuario.

```
 1. Ce.suministrad.=NCe 38725 ==> Centro=18H0 38725    conf:(1)
 2. Almacén=1801 38016 ==> Centro=18H0 38016    conf:(1)
 3. Ind.borrado=NL 37936 ==> Centro=18H0 37936    conf:(1)
 4. Ce.suministrad.=NCe Almacén=1801 37177 ==> Centro=18H0 37177    conf:(1)
 5. Ind.borrado=NL Ce.suministrad.=NCe 37131 ==> Centro=18H0 37131    conf:(1)
 6. Ind.borrado=NL Almacén=1801 36416 ==> Centro=18H0 36416    conf:(1)
 7. Ind.borrado=NL Ce.suministrad.=NCe Almacén=1801 35615 ==> Centro=18H0 35615    conf:(1)
 8. Ind.borrado=NL 37936 ==> Ce.suministrad.=NCe 37131    conf:(0.98)
 9. Ind.borrado=NL Centro=18H0 37936 ==> Ce.suministrad.=NCe 37131    conf:(0.98)
10. Ind.borrado=NL 37936 ==> Ce.suministrad.=NCe Centro=18H0 37131    conf:(0.98)
11. Centro=18H0 39568 ==> Ce.suministrad.=NCe 38725    conf:(0.98)
12. Ind.borrado=NL Almacén=1801 36416 ==> Ce.suministrad.=NCe 35615    conf:(0.98)
13. Ind.borrado=NL Centro=18H0 Almacén=1801 36416 ==> Ce.suministrad.=NCe 35615    conf:(0.98)
14. Ind.borrado=NL Almacén=1801 36416 ==> Ce.suministrad.=NCe Centro=18H0 35615    conf:(0.98)
15. Almacén=1801 38016 ==> Ce.suministrad.=NCe 37177    conf:(0.98)
16. Centro=18H0 Almacén=1801 38016 ==> Ce.suministrad.=NCe 37177    conf:(0.98)
17. Almacén=1801 38016 ==> Ce.suministrad.=NCe Centro=18H0 37177    conf:(0.98)
18. Centro=18H0 39568 ==> Almacén=1801 38016    conf:(0.96)
19. Ce.suministrad.=NCe 38725 ==> Almacén=1801 37177    conf:(0.96)
20. Ce.suministrad.=NCe Centro=18H0 38725 ==> Almacén=1801 37177    conf:(0.96)
21. Ce.suministrad.=NCe 38725 ==> Centro=18H0 Almacén=1801 37177    conf:(0.96)
22. Ind.borrado=NL 37936 ==> Almacén=1801 36416    conf:(0.96)
23. Ind.borrado=NL Centro=18H0 37936 ==> Almacén=1801 36416    conf:(0.96)
24. Ind.borrado=NL 37936 ==> Centro=18H0 Almacén=1801 36416    conf:(0.96)
25. Ind.borrado=NL Ce.suministrad.=NCe 37131 ==> Almacén=1801 35615    conf:(0.96)
26. Ind.borrado=NL Ce.suministrad.=NCe Centro=18H0 37131 ==> Almacén=1801 35615    conf:(0.96)
27. Ind.borrado=NL Ce.suministrad.=NCe 37131 ==> Centro=18H0 Almacén=1801 35615    conf:(0.96)
28. Ce.suministrad.=NCe 38725 ==> Ind.borrado=NL 37131    conf:(0.96)
29. Ce.suministrad.=NCe Centro=18H0 38725 ==> Ind.borrado=NL 37131    conf:(0.96)
30. Ce.suministrad.=NCe 38725 ==> Ind.borrado=NL Centro=18H0 37131    conf:(0.96)
31. Centro=18H0 39568 ==> Ind.borrado=NL 37936    conf:(0.96)
32. Ce.suministrad.=NCe Almacén=1801 37177 ==> Ind.borrado=NL 35615    conf:(0.96)
33. Ce.suministrad.=NCe Centro=18H0 Almacén=1801 37177 ==> Ind.borrado=NL 35615    conf:(0.96)
34. Ce.suministrad.=NCe Almacén=1801 37177 ==> Ind.borrado=NL Centro=18H0 35615    conf:(0.96)
35. Almacén=1801 38016 ==> Ind.borrado=NL 36416    conf:(0.96)
36. Centro=18H0 Almacén=1801 38016 ==> Ind.borrado=NL 36416    conf:(0.96)
37. Almacén=1801 38016 ==> Ind.borrado=NL Centro=18H0 36416    conf:(0.96)
38. Centro=18H0 39568 ==> Ce.suministrad.=NCe Almacén=1801 37177    conf:(0.94)
39. Ind.borrado=NL 37936 ==> Ce.suministrad.=NCe Almacén=1801 35615    conf:(0.94)
40. Ind.borrado=NL Centro=18H0 37936 ==> Ce.suministrad.=NCe Almacén=1801 35615    conf:(0.94)
```

Figura 30: Reglas de asociación encontradas por A-priori en la base de datos de EsSalud. Fuente: WEKA.

a) Análisis de conocimiento

Algunas de las reglas más relevantes que podemos interpretar serían las siguientes:

- La regla 16 nos indica que comúnmente cuando una solicitud de compra es generada en base a requerimientos del centro identificado con el código "18H0" y del almacén identificado con código "1801", los insumos médicos adquiridos por el hospital no son delegados a otros centros de la ciudad.
- La regla 24 nos indica que comúnmente órdenes de compra que no han sido alteradas o eliminadas por completo son realizadas desde el centro identificado con código "18H0", además que se realizaron debido a una solicitud de requerimiento del almacén "1801" de dicho centro.

CONCLUSIONES GENERALES

Al terminar la presente investigación, interesante en su pleno desarrollo, se llegó a las siguientes conclusiones en base a los resultados obtenidos:

PRIMERA: Se ha logrado detallar los algoritmos de minería de datos más utilizados por experimentadores y aplicados en organizaciones respecto a un mínimo de 5 características abarcadas dependiendo del tipo de algoritmo analizado, ya sea clasificación, agrupamiento (clusterización) o asociación, permitiendo de esta manera identificar aquellos que operan de manera más eficaz para el procesamiento de data orientada a entornos comerciales.

SEGUNDA: Se puede concluir que tanto la etapa investigativa como experimental han cumplido satisfactoriamente con seleccionar los algoritmos candidatos y probar dichos algoritmos en data experimental y data real respectivamente, permitiendo de esta manera cumplir con el objetivo principal de la investigación y corroborar los resultados a través de un caso de estudio.

TERCERA: A lo largo de la investigación se ha demostrado el gran potencial que tienen las técnicas de minería de datos, ya que éstas no solo representan una gran contribución para la optimización de varios procesos de la empresa, sino una gran ventaja competitiva en el mercado actual. Dicho potencial también contribuye

significativamente en la buena toma de decisiones, dejando de lado métodos tradicionales y poco efectivos como basarse en meras corazonadas.

CUARTA: Se ha comprobado una las premisas de la investigación, y es que como se indicó la minería de datos es un proceso crucial para cualquier organización que requiere una gran cantidad de tiempo y paciencia, no solo es una actividad monótona que requiere de una gran inversión; al analizar los resultados obtenidos se ha demostrado que cualquier tipo de organización que gestione sus datos de manera correcta es capaz crecer y optimizar todos sus procesos.

QUINTA: Al experimentar con algoritmos de minería de datos orientados tanto a la clasificación, agrupamiento y asociación, se determinó que el número de instancias como atributos juegan un papel muy importante al momento de elegir el algoritmo correcto para el procesamiento de los datos, esto debido a características evaluadas y asociadas a cada clase de algoritmo en particular, características tales como, por ejemplo, la calidad del árbol de clasificación, el tiempo de ejecución del algoritmo o la calidad del conocimiento generado; ya que para bases de datos con una cantidad menor de instancias algunos algoritmos generan mejor calidad de conocimiento debido a que en la construcción de los modelos predictivos emplean un mayor número de atributos, permitiéndole al usuario interpretar de manera más detallada el conocimiento deseado, utilizando para ello un tiempo de ejecución aceptable para mostrar dichos resultados; en cambio para repositorios de datos con una gran cantidad de instancias, cuyo tiempo de ejecución para los algoritmos aumentaría exponencialmente, podría ser más recomendable el uso de algoritmos que no muestren un análisis tan detallado, pero con la suficiente calidad para hallar el conocimiento deseado y con un tiempo de ejecución exponencialmente más bajo.

SEXTA: Al emplear los algoritmos de clasificación en la etapa de experimentación se pudo determinar que bases de datos con una gran cantidad de atributos binarios generan también una gran cantidad de ramas de conocimiento innecesarias en los árboles de clasificación, entorpeciendo de esta manera el proceso. Para bases de datos orientadas al ámbito comercial o afines, las cuales comúnmente serán transaccionales, sería recomendable pre-procesar la data de tal manera que únicamente cuente con atributos de datos que sean relevantes para el proceso de análisis tal como sería, por ejemplo, el género de los clientes.

SÉPTIMA: Al emplear los algoritmos de clusterización en la etapa de experimentación se pudo determinar que el tiempo de ejecución entre los dos algoritmos seleccionados para las pruebas es exponencialmente mayor uno respecto al otro, mucho mayor de lo ocurrido anteriormente con los algoritmos de clasificación, siendo el algoritmo EM el que ocupa una mayor cantidad de tiempo para generar su modelo de análisis; sin embargo, el algoritmo EM genera un análisis más detallado respecto a cada etiqueta de clase, caso contrario el K-Means únicamente muestra centroides de los clústeres hallados, pero en un tiempo exponencialmente menor al tomado por EM. Como se mencionó anteriormente el uso de cada uno de estos algoritmos dependerá de los requerimientos de cada usuario.

OCTAVA: Al emplear el algoritmo de asociación A-priori en la etapa de experimentación se pudo determinar que, como en casos anteriores, el uso excesivo de atributos binarios al momento de generación de reglas entorpece dicho proceso, ya que para la generación de las mimas el algoritmo emplea frecuencias de ocurrencia entre uno o más atributos, por lo que si se tiene más de una variable binaria irrelevante como,

por ejemplo, si un artículo fue puesto a la venta un viernes o un sábado y cada uno de estos días cuenta con un atributo booleano propio, el algoritmo los relacionará debido a las numerosas ocurrencias y lo mostrará como una de las mejores reglas de asociación, sin que este nos brinde información realmente relevante.

REFERENCIAS

Advertising Age. (2016). El desafío de administrar la información. Recuperado de: http://argentina.pmfarma.com/articulos/723-el-desafio-de-administrar-la-informacion.html

Ajit Singh. (2005). The EM Algorithm.

Akshita Bhandari, Ashutosh Gupta y Debasis Das. (2014). IMPROVISED APRIORI ALGORITHM USING FREQUENT PATTERN TREE FOR REAL TIME APPLICATIONS.

Braulio José Solano Rojas. (2016). Tareas de la minería de datos: reglas de asociación y secuencias. Recuperado de: http://bsolano.com/ecci/claroline/backends/download.php/UHJlc2VudGFjaW9uZXMvNy5fVGFyZWFzX2RlX2xhX21pbmVy7WFfZGVfZGF0b3MsX3JlZ2xhc19kZV9hc29jaWFjafNuLnBkZg%3D%3D?cidReset=true&cidReq=CI2352

Naveena Devi y O. Sreevani. (2010). Dynamic Modelling Approach For Web Usage Mining Using Open Web Resources.

Carlos Fernández. (2016). Qué es un Data Warehouse? Recuperado de: http://www.dataprix.com/que-es-un-datawarehouse

Carlos Hurtado L. (2016). Arboles de Decisión (I).

Claudia Elena Dinucă, (2011), Using Web Mining In E-Commerce Applications.

George Karypis, Eui-Hong (Sam) y Han Vipin Kumar. (2015). CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling.

Ian H. Witten, Eibe Frank, Mark A. Hall, (2011), Data Mining, Practical Machine Learning Tools and Techniques.

Jiawei Han y Micheline Kamber, (2006), Data Mining Concepts and Techniques.

Ji Wentian, Guo Qingju, Zhong Sheng y Zhou En. (2013). Improved K-medoids Clustering Algorithm under Semantic Web.

José Antonio Camarena Ibarrola. (2016). El Algoritmo E-M.

K. Wisaeng. (2013). A Comparison of Different Classification Techniques for Bank Direct Marketing.

Manish Verma, Mauly Srivastava, Neha Chack, Atul Kumar Diswar y Nidhi Gupta. (2012). A Comparative Study of Various Clustering Algorithms in Data Mining.

Martin Ester, Hans-Peter Kriegel, Jiirg Sander y Xiaowei Xu. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise.

Md. Rafiqul Islam and Md. Ahsan Habib (2015), A Data Mining Approach To Predict Prospective Business Sectors For Lending In Retail Banking Using Decision Tree.

MD Marketing Digital. (2016). Qué es el Marketing Digital? Recuperado de: http://www.mdmarketingdigital.com/que-es-el-marketing-digital.php

Miguel Cárdenas-Montes. (2015). PREPROCESADO DE DATOS PARA MINERIA DE DATOS.

Muhammad Ali Masood, M. N. A. Khan. (2015). Clustering Techniques in Bioinformatics.

Nguyen Thi Linh y Christopher Chua. (2013). Application of CURE Data Clustering Algorithm to Batangas State University Student Database.

Onur Doğan, Hakan Aşan y Ejder Ayçın, (2015), Use Of Data Mining Techniques In Advance Decision Making Processes In A Local Firm.

Peter L. Hammer y Tibérius Bonates. (2006). Logical Analysis of Data: From Combinatorial Optimization to Medical Applications.

Raymond T. Ng y Jiawei Han. (2002). CLARANS: A Method for Clustering Objects for Spatial Data Mining.

Ronald Hochreiter y Christoph Waldhauser (2014), Data Mining Cultural Aspects of Social Media Marketing.

Safia Abbas. (2015), Deposit subscribe Prediction using Data Mining Techniques based Real Marketing Dataset.

Sagar S. Nikam. (2015). A Comparative Study of Classification Techniques in Data Mining Algorithms.

Search Data Managment. (2016). Master Data. Recuperado de: http://searchdatamanagement.techtarget.com/definition/master-data

Pragya Agarwal, Madan Lal Yadav y Nupur Anand. (2013). Study on A priori Algorithm and its Application in Grocery Store.

Suhail Ansari, Ron Kohavi, Llew Mason, and Zijian Zheng Integrating. (2000). E-Commerce and Data Mining Architecture and Challenges

Surjeet Kumar Yadav, Brijesh Bharadwaj, Saurabh Pal. (2012). Data Mining Applications: A comparative Study for Predicting Student's performance.

The National Academies Press. (2016). Characteristics of Scientific and Technical Databases. Recuperado de: https://www.nap.edu/read/9693/chapter/6

Yoichi Hayashi y Satoshi Nakano. (2015). Use of a Recursive-Rule eXtraction algorithm with J48 graft to achieve highly accurate and concise rule extraction from a large breast cancer dataset.

WhatIs. (2016). Reference data. Recuperado de: http://whatis.techtarget.com/definition/reference-data

WhatIs. (2016). Transactional data. Recuperado de: http://whatis.techtarget.com/definition/transactional-data

Wikipedia. (2016). Almacén de datos. Recuperado de: https://es.wikipedia.org/wiki/Almac%C3%A9n_de_datos

Wikipedia. (2016). Minería de datos. Recuperado de: https://es.wikipedia.org/wiki/Miner%C3%ADa_de_datos

Wikipedia. (2016). Reglas de asociación. Recuperado de: https://es.wikipedia.org/wiki/Reglas_de_asociaci%C3%B3n

Zahra Asghari Varzaneh y Nasibeh Emami Chukanlo. (2012). A survey of hierarchical clustering algorithms.