Universidad de las Ciencias Informáticas
Facultad 1

Minería de Datos aplicada al sistema AiresProxy

Trabajo de Diploma para optar por el Título de Ingeniero en Ciencias Informáticas

Autora:

Elisandra Corrales Estrada

Tutores:

Lic. José Alberto Ponce Pérez

Ing. Miguel Angel Chavez Alfonso

La Habana, Cuba, junio de 2012

"Año 54 de la Revolución"

Frase

Alguien dijo alguna vez:

"No hay por qué sufrir, la vida es la manifestación de los sueños, y los sueños se escogen"

Declaración de Autoría

Declaramos ser los únicos autores de la presente tesis y se autoriza a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio.

Para que así conste firmamos la presente a los ____ días del mes de junio del año 2012.

_____

Elisandra Corrales Estrada

Firma del autor

_____ _____

José Alberto Ponce Pérez

Firma del tutor

Miguel Angel Chavez Alfonso

Firma del tutor

Agradecimientos

Mi primer agradecimiento es para mis padres Maricela y Blas, porque han sido mi ejemplo desde que nací, porque no me imagino la vida sin ellos, sin sus consejos y su confianza. Todo lo hago para que estén orgullosos de mí. Este título también les pertenece.

Gracias a mi hermanita Elizabeth por ser el motor impulsor para seguir adelante, porque siempre he querido darle un buen ejemplo y un futuro mejor. Todo lo he hecho pensando en ti.

A mi novio Rodo, gracias por la ayuda en estos cinco años de carrera, por darme tu fuerza, tu espíritu y tu amor para seguir año tras año. Y porque hiciste la tesis como tuya y me ayudaste en todo momento. Te amo.

Agradezco a mis abuelos, mis tíos, primos, a mi familia de forma general, porque todos han influido en mi formación profesional y me brindaron todo su apoyo en estos años de estudio.

A la familia de Rodo también porque me han acogido como parte de la familia y me han ayudado en cuanto han podido. (Aquí incluyo a la familia habanera que nos acogieron en sus casas y nos apoyaron siempre).

Gracias a los tutores por el apoyo brindado para la realización de la tesis, a mis profes de la universidad, a la Revolución, a todos los que lucharon por nuestra Patria y al Comandante Fidel.

Dedicatoria

Dedico este trabajo a mis padres por todo el sacrificio que han hecho para que yo pudiera estar donde estoy, a mi hermana por ser mi inspiración, a Rodo por su infinito amor y a mi familia por estar siempre para mí. Mis éxitos son los suyos.

Resumen

Con la alta disponibilidad de datos producidos a diario en el mundo de la informática, las técnicas tradicionales estadísticas para su procesamiento no suelen aprovechar al máximo la información de valor cognitivo implícita en los mismos. Se hace necesario para ello, implementar nuevas técnicas y herramientas que sirvan de ayuda para solucionar esta problemática. Bajo estas condiciones surge la Minería de Datos como alternativa para la obtención de patrones ocultos en un conjunto de datos.

La Universidad de las Ciencias Informáticas (UCI) cuenta con un servicio de cuotas de navegación por Internet donde se genera un enorme volumen de información que registran los servidores proxy. La Dirección de Redes y Seguridad Informática (DRSI) no aprovecha el conocimiento implícito en los registros de navegación para describir el comportamiento de los usuarios en el uso de las cuotas, por lo tanto, no posee argumentos sólidos para tomar decisiones de gestión y usabilidad en el servicio de navegación de Internet.

En este trabajo se presenta una propuesta a partir de un estudio realizado sobre las técnicas y algoritmos de Minería de Datos, para la selección adecuada del o los algoritmos que puedan aplicarse en el sistema AiresProxy; con el objetivo de obtener patrones, tendencias y relaciones dentro de los datos, que describan el uso por los usuarios a las cuotas de navegación por Internet. Para ello se desarrolla un proceso de descubrimiento de conocimiento, aplicando el algoritmo seleccionado en un caso de estudio; a partir de una muestra de datos generados por el servidor proxy. Posteriormente se realiza el análisis y discusión del proceso, demostrando con los resultados obtenidos la utilidad de la investigación.

Palabras claves: Minería de Datos, descubrimiento de conocimiento, AiresProxy, servidor proxy, cuotas de navegación, técnicas y algoritmos.

Índice

Introducción

El avance de las Tecnologías de la Información y las Comunicaciones (TIC), ha tomado desde hace varios años, un ritmo acelerado. Los cambios en la manera de comunicarnos y compartir información interactivamente forman parte de la vida cotidiana. Internet como elemento esencial de las TIC permite el intercambio de datos de forma fiable y eficiente entre ordenadores [1].

El acceso en Cuba a Internet es actualmente restringido, a causa de las leyes impuestas por el bloqueo norteamericano el país no puede conectarse a los canales internacionales de fibra óptica que pasan muy cerca de las costas cubanas, haciéndolo por vía satélite, lo que es más costoso y limita considerablemente este recurso. La limitación expuesta ha determinado que la conexión se realice de manera organizada para garantizar su uso social.

En la Universidad de las Ciencias Informáticas (UCI), Internet constituye una de las principales fuentes de consulta de conocimientos para la docencia, la investigación y la producción. Como mecanismo de control de acceso se utiliza un servidor proxy, el cual se emplea como intermediario en la comunicación entre un host interno e Internet. Para la navegación los usuarios cuentan con un sistema de cuotas, implementado mediante un análisis que se desarrolla en la Dirección de Redes y Seguridad Informática (DRSI) de la institución a partir de los datos almacenados en los registros generados por el servidor proxy.

Para la aplicación de las reglas implementadas en el sistema de cuotas, la DRSI analiza un elevado volumen de información de forma automatizada utilizando diferentes herramientas que permiten la generación de reportes estadísticos. Entre las herramientas destinadas para el análisis de los datos por la DRSI se encuentra AiresProxy, software desarrollado en la universidad, con el objetivo de brindar reportes sobre la navegación de los usuarios. A partir de los reportes que brinda AiresProxy, los administradores de la DRSI infieren determinados comportamientos de los usuarios lo que permite definir las políticas de navegación. Estos reportes no tienen en cuenta diferentes variables y observaciones que posibiliten asociaciones con un elevado nivel de exactitud. Por otra parte, AiresProxy no realiza análisis de datos de forma

inteligente que describa el comportamiento de los usuarios, constituyendo esto una limitación en términos de competitividad y comercialización.

Los datos tal cual se almacenan no suelen proporcionar beneficios directos. Su valor real reside en la información que se pueda extraer de ellos que ayuden a tomar decisiones o a mejorar la comprensión de los fenómenos que rodean la humanidad.

Una forma muy valiosa de análisis de la información es la Minería de Datos (MD). La MD es una de las etapas de lo que se conoce como el Descubrimiento de Conocimiento en Bases de Datos (Knowledge Discovery in Databases: KDD). Este proceso consta de varias fases e incorpora diferentes técnicas de Aprendizaje Automático, la Estadística, las Bases de Datos, los Sistemas de Toma de Decisiones, las técnicas de Visualización y otras áreas de la informática y de la gestión de información. La MD puede definirse inicialmente como un proceso de descubrimiento de nuevas y significativas relaciones, patrones y tendencias al examinar grandes cantidades de datos [2].

Con el surgimiento de la MD el análisis de los datos se ha encaminado hacia diferentes técnicas y algoritmos que persiguen el descubrimiento automático del conocimiento contenido en la información almacenada de modo ordenado en las bases de datos [3]. Las técnicas de mayor aplicación son las de agrupamiento o clustering, las de reglas de asociación, redes neuronales y técnicas de clasificación.

Como se ha descrito, AiresProxy no realiza análisis inteligente, siendo de gran importancia para la DRSI de la universidad contar con una herramienta que a partir de los registros de navegación, permita obtener patrones, tendencias y relaciones dentro del conjunto de datos para conocer mejor el uso que le dan los usuarios a las cuotas de navegación, aportando al sistema de toma de decisiones de la universidad una visión más amplia.

Por todo lo anterior, se hace necesaria la búsqueda de nuevos modelos, algoritmos y técnicas computacionales que permitan, a través de procesos inteligentes, obtener resultados que sean de interés para los especialistas y directivos del centro y que brinde nuevos conocimientos sobre la información generada.

De acuerdo a la situación problémica planteada se identifica como **problema científico**: ¿Cómo identificar patrones de comportamiento de los usuarios que acceden a Internet, a partir de los datos almacenados en AiresProxy?

Con el fin de resolver el problema planteado se ha trazado el siguiente **objetivo general**: Determinar de las técnicas y algoritmos de minería de datos, cuál o cuáles son aplicables en AiresProxy para obtener patrones que describan el comportamiento de los usuarios en el uso de las cuotas de navegación. Del cual se derivan los siguientes **objetivos específicos**:

- Caracterizar el proceso de descubrimiento de conocimiento en bases de datos, así como las técnicas y algoritmos de minería de datos más utilizados a nivel mundial que puedan ser empleados para la extracción de conocimiento en el sistema AiresProxy
- Identificar las técnicas y los algoritmos de minería de datos aplicables a los registros del sistema AiresProxy.
- Describir las funcionalidades a implementar con el uso de las técnicas y algoritmos de minería de datos seleccionados.
- Diseñar la estructura del sistema de base de datos de AiresProxy necesaria para la implementación de las técnicas y algoritmos de minería de datos propuestos.
- Validar la solución propuesta.

El cumplimiento y ejecución de los objetivos propuestos lleva consigo un estudio que sirva de apoyo para el cumplimiento de lo antes mencionado, es por ello que se define como **objeto de estudio** el proceso de descubrimiento de conocimiento en bases de datos y como **campo de acción** las técnicas y algoritmos de minería de datos.

Para lograr los objetivos específicos se formularon las siguientes **tareas de investigación**:

1. Caracterización del proceso de descubrimiento de conocimiento en bases de datos.

2. Caracterización de las etapas por las que transita el proceso de descubrimiento de conocimiento, enfatizando en la minería de datos.

3. Precisión de las tendencias actuales del uso de los algoritmos y técnicas de minería de datos más utilizados a nivel mundial.

4. Selección de las técnicas y algoritmos de minería de datos a utilizar en nuevas versiones del sistema AiresProxy.

5. Caracterización del sistema de generación de reportes AiresProxy enfatizando en las funcionalidades que este brinda.

6. Desarrollo de una propuesta de nuevas funcionalidades al sistema AiresProxy en las que se empleen el uso de los algoritmos de minería de datos seleccionados.

7. Diseño de la estructura de la base de datos adecuada para la aplicación de los algoritmos de minería de datos seleccionados.

8. Validación de la propuesta.

Se emplean diversos **métodos científicos**, los cuales posibilitan dirigir el proceso de investigación de forma óptima, de modo que permita alcanzar su propósito de la manera más eficiente.

Los métodos teóricos utilizados en esta investigación son: el Histórico-Lógico y el Analítico-Sintético.

Histórico-lógico: este método se utiliza para estudiar las tendencias de las técnicas y algoritmos de MD que se utilizan para el descubrimiento de conocimiento, así como los principales conceptos en esta área y el estado actual. Permite además conocer las herramientas y las metodologías para el desarrollo de proyectos de MD.

Analítico-Sintético: permite realizar un análisis sobre las técnicas de MD existentes para la extracción de conocimiento, sintetizando los algoritmos aplicables para el desarrollo de funcionalidades que incluyan análisis inteligente en el sistema AiresProxy.

Los métodos empíricos empleados son: la entrevista y el experimento.

Entrevista: se realizaron entrevistas para identificar las necesidades de la Dirección de Redes de la universidad. Por otra parte para tener en cuenta los criterios de los miembros de AiresProxy en base a la elección del algoritmo adecuado para la propuesta.

Experimento: este método se utiliza para realizar un proceso de minado de datos, experimentando algunos datos que nos puedan mostrar la realidad del uso que le dan los usuarios a las cuotas de navegación.

Justificación de la investigación: Se agrega un valor al producto AiresProxy, que utilizando técnicas y algoritmos de MD se ubicaría entre los mejores software de análisis de registros de servidores proxy, convirtiéndose así en un producto competitivo y comerciable. AiresProxy con la inclusión de funcionalidades con análisis inteligente, aporta nuevos conocimientos al sistema de toma de decisiones de los directivos de la universidad; permitiendo realizar análisis exploratorios dentro de los conjuntos de datos, buscando relaciones y patrones que describan el comportamiento de los usuarios en el uso de las cuotas de navegación, logrando un mejor aprovechamiento de las cuotas en función de las necesidades del país y optimizando su uso en base a la productividad y al desarrollo tecnológico.

Capítulo 1. Fundamentación Teórica

En el presente capítulo se investiga sobre el descubrimiento de conocimiento en bases de datos, los aspectos principales de un proceso de MD y los modelos, técnicas y algoritmos para realizar dicho proceso. Se realiza un estudio de las metodologías más utilizadas en proyectos de explotación de datos y se exponen las principales herramientas para apoyar la MD, optando por aquella que pueda servir para la experimentación del proceso de extracción de conocimiento. Se analiza el sistema AiresProxy enfatizando en las funcionalidades que brinda y el sistema de base de datos que utiliza, para obtener una mejor comprensión de su funcionamiento.

1.1. El proceso de descubrir conocimiento en bases de datos

En la actualidad la cantidad de datos que se almacena en las Bases de Datos (BD) supera las habilidades humanas para el análisis de estos sin la necesidad de usar técnicas automatizadas. Con el importante progreso de la informática y las tecnologías relacionadas con esta y su expansión por todas las esferas, el almacenamiento de los datos continúa creciendo considerablemente.

Los datos son la materia prima que se convierte en información cuando un usuario los analiza y les atribuye algún significado especial. Se puede hacer referencia a estos como conocimiento, desde el momento en que se elabora o encuentra un modelo para interpretar esta información y con este modelo se obtiene un valor agregado.

Existe un proceso definido para el análisis de datos con el fin de encontrar patrones en grandes cantidades de datos llamado Descubrimiento de Conocimiento en Bases de Datos (*Knowledge Discovery in Databases*: KDD).

KDD es el proceso completo de extracción de información, que se encarga desde la preparación de los datos hasta la interpretación de los resultados obtenidos y se define como: "…el proceso no trivial de identificar patrones válidos, novedosos, potencialmente útiles, y finalmente comprensibles." [4]

El objetivo fundamental del KDD es encontrar datos novedosos que sean útiles a los usuarios, luego de encontrar patrones relevantes e identificarlos como posible conocimiento se debe presentar la información de forma clara y comprensible. Para lograrlo el KDD se traza diferentes metas [5]:

- Procesar automáticamente grandes cantidades de datos crudos.
- Identificar los patrones más significativos y relevantes.
- Presentarlos como conocimiento apropiado para satisfacer las necesidades del usuario.

El volumen de datos que requieren procesamiento y análisis en las BD exceden las capacidades humanas y la dificultad de transformar los datos con precisión en conocimiento, por lo que va más allá de los límites de las bases de datos tradicionales. Por consiguiente, la utilización plena de los datos almacenados depende del uso de técnicas de descubrimiento de conocimientos.

1.1.1. Etapas en el proceso de KDD

El KDD comienza con la recopilación e integración de la información a partir de unos datos iniciales de que se dispone [2]. Para ello se estudia qué datos se necesitan, dónde se pueden encontrar y cómo conseguirlos. Además se hace una integración de múltiples fuentes heterogéneas de datos en una única fuente. Todo esto ocurre en la fase de Integración y Recopilación.

Luego se pasa a la fase de Selección, Limpieza y Transformación donde se eliminan o corrigen los datos incorrectos, que al ser obtenidos de diferentes fuentes presentaron valores erróneos o faltantes y se decide qué tratamiento darle a los datos incompletos.

Estas dos primeras etapas se conocen como Preparación de Datos y determinan que las fases sucesivas sean capaces de extraer conocimiento válido y útil a partir de la información original.

Una vez se tienen los datos adecuados se procede a la fase de la Minería de Datos, proceso en el que se seleccionarán las técnicas adecuadas y se aplican algoritmos para lograr los objetivos pretendidos.

En la fase de Evaluación e Interpretación se evalúan los patrones y se analizan, y si es necesario se vuelve a las fases anteriores para una nueva iteración. Finalmente en la fase de difusión se hace uso del nuevo conocimiento y se hace partícipe de él a todos los posibles usuarios [2]. En el libro *Minería de Datos, Técnicas y herramientas,* se plantea la siguiente figura que explica las fases descritas.

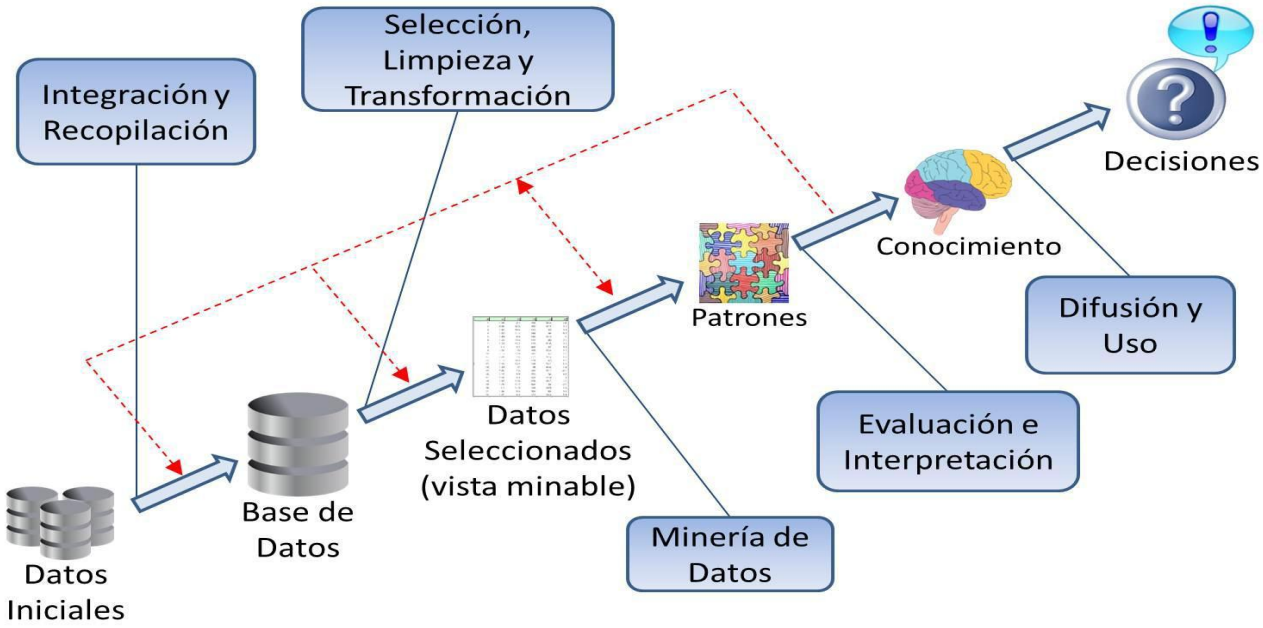


Figura 1: Fases de un proceso KDD [2].

La Figura revela que la salida de una fase constituye la entrada de la siguiente, no obstante, esta salida puede hacer que se repita el proceso, quizás con otros datos, algoritmos, metas y estrategias. Este carácter iterativo e interactivo del proceso KDD es muy importante para extraer conocimiento de alta calidad.

Frecuentemente, se incluye una fase previa a las descritas, en la que se analizan las necesidades de la organización y definición del problema, y se establecen los objetivos de minería de datos; es decir: se definen y priorizan los objetivos del negocio [3], esta fase es conocida como Entendimiento del Dominio. La preparación de datos consume del 60 al 90 por ciento del tiempo necesario para extraer datos y aporta del 75 al 90 por ciento del éxito de un proyecto de minería [4]. (Ver Anexo1)

Una de las premisas del KDD es que el conocimiento es descubierto usando técnicas de aprendizaje inteligente que van examinando los datos a través de procesos automatizados. Para que una técnica sea considerada útil para el descubrimiento del conocimiento, éste debe ser interesante, es decir, debe tener un valor potencial para el usuario [4].

En el proceso de KDD es de gran importancia la identificación de los datos. Para ello hay que imaginar qué datos se necesitan y encontrarlos. Una vez que se tienen los datos, se deben seleccionar aquellos que sean útiles para los requerimientos propuestos. Se preparan, poniéndolos en un formato adecuado. Luego se procede a la minería de datos, proceso en el que se seleccionarán las herramientas y técnicas necesarias para lograr los objetivos perseguidos. Y tras este proceso llega el análisis de los resultados, con lo que se obtiene el conocimiento.

1.2. Minería de Datos

Con el transcurso de los últimos años, el volumen y variedad de información que se encuentra informatizada en las bases de datos y otras fuentes han crecido espectacularmente. Estos datos se originan tanto por transacciones internas, que tienen su origen en las actividades administrativas y comerciales de una organización, así como por fuentes externas. Los datos pueden proceder de fuentes diversas y pertenecer a diferentes dominios, por lo que resulta inminente la necesidad de analizar los mismos para la obtención de información.

El descubrimiento de la información oculta se hace posible gracias a la Minería de Datos, una de las etapas dentro del proceso KDD que aplica sofisticadas técnicas para encontrar patrones y relaciones dentro de los datos permitiendo la creación de modelos [5].De modo que el objetivo

principal de la MD es encontrar patrones de comportamiento en los datos y a partir de ellos generar conocimiento.

Se define como Minería de Datos al proceso de extraer conocimiento útil y comprensible, previamente desconocido, desde grandes cantidades de datos almacenados en distintos formatos. En esencia, la tarea fundamental de la Minería de Datos es encontrar modelos inteligentes. Por sus características es imprescindible que sea realizado como un proceso automático o semiautomático [3].

Se puede decir entonces, que la MD es el proceso para descubrir conocimiento (reglas, patrones) a partir de grandes volúmenes de datos, apoyados en técnicas o herramientas (automáticas o semiautomáticas), de tal manera que su uso ayude a tomar decisiones más seguras que reporten algún tipo de beneficio a las organizaciones.

1.2.1. La minería de datos y su relación con otras disciplinas

La MD es un campo multidisciplinario que ha evolucionado y se ha desarrollado en paralelo o como prolongación de otras tecnologías. Con esto se puede decir que a partir de los avances en otras disciplinas, aparecen los avances en la MD.

Destacan entre las disciplinas que se encuentran más unidas a la MD las siguientes:

Estadística: a menudo se le considera como el área que dio lugar a la minería de datos, pues ha proporcionado muchos de los conceptos, algoritmos y técnicas de análisis para calcular información [3].

Bases de Datos: es el área de donde se nutre el minado de datos. Los conceptos como los Almacenes de Datos o *Data Warehouse* y el procesamiento analítico en línea (OLAP (*On-Line Analytical Processing*)), tienen una gran relación con la minería de datos, donde el acceso eficiente a los datos es muy relevante [3]. Los almacenes de datos proporcionan acceso a datos para análisis complejos, revelación de conocimientos y toma de decisiones. Por otra parte, OLAP permite realizar análisis de datos complejos del almacén de datos [6]. Este análisis suele implicar,

generalmente, la lectura de grandes cantidades de datos para llegar a extraer algún tipo de información útil.

Aprendizaje Automático: es el área de la Inteligencia Artificial capaz de desarrollar métodos computacionales que implementan varias formas de aprendizaje y que aporta algoritmos que mejoran automáticamente a través de la experiencia, centrándose fundamentalmente en la inducción y siendo aplicables a datos tanto numéricos como simbólicos [7]. El aprendizaje automático y la minería de datos se solapan en gran medida, en cuanto a los problemas que tratan y los algoritmos que utilizan.

Visualización de Datos: la visualización de datos es fundamental para comprender los datos de una forma clara y lógica.

Las tecnologías que más destacan son los métodos estadísticos y el aprendizaje automático. Los métodos estadísticos porque forman parte del nacimiento de la MD, el aprendizaje automático para la obtención de reglas de aprendizaje y modelos de los datos, necesita la ayuda de la estadística; constituyendo ambas las áreas más importantes en el surgimiento de la MD.

La amplia integración de otras disciplinas con la MD ha contribuido al progreso de numerosas herramientas de alcance mundial, es por eso que el avance de las diferentes áreas es fundamental en su evolución y desarrollo.

1.3. Metodologías de desarrollo para proyectos de Minería de Datos

Un proyecto de KDD centra su mayor esfuerzo en las fases de Entendimiento del Dominio y la Preparación de los Datos como se mostró anteriormente, lo que unido al hecho de que no se puede arribar a conclusiones por adelantado y el proceso de desarrollo de proyectos de Minería de Datos suele ser engorroso y difícil, normalmente provoca retrasos y desviaciones en la planificación inicial. Por lo que con el objetivo de guiar y organizar el trabajo se han desarrollado diversas metodologías, las cuales con un buen empleo facilitan:

- Hacer una buena planificación y dirección del proyecto.

- Realizar un seguimiento de calidad del proyecto, para lograr la satisfacción del cliente.
- Desarrollar nuevos proyectos de MD con características similares.

Existen diversas metodologías de desarrollo para proyectos de MD tales como: CRISP-DM (*CRoss Industry Standard Process for Data Mining*), SEMMA (*Sample, Explore, Modify, Model, Asses*), Metodología de las cinco A's (*Asses, Access, Analyze, Act, Automate*), Modelo de proceso de Minería de Datos de *Two Crows*, CRITIKAL (*Client-Server Rule Induction Technology for Industrial Knowledge Acquisition from Large Databases*) y Metodología SQL Server- 2005.

Entre las más conocidas y empleadas según un estudio realizado por el *SAS Institute* están: CRISP-DM y SEMMA [8], esto se refleja en el Anexo 2. Las metodologías SEMMA y CRISP-DM comparten la misma esencia, estructurando el proyecto de explotación de datos en fases que se encuentran interrelacionadas entre sí, convirtiéndolo en un proceso iterativo e interactivo [9].

La metodología SEMMA se centra específicamente en las características técnicas del desarrollo del proceso, mientras que la metodología CRISP-DM, mantiene una perspectiva más amplia respecto a los objetivos empresariales del proyecto. Esta diferencia se establece ya desde la primera fase del proyecto de explotación de datos donde la metodología SEMMA comienza realizando un muestreo de datos, mientras que la metodología CRISP-DM comienza realizando un análisis del problema inicial para su transformación en un problema técnico, consultar Anexo 3. Desde ese punto de vista más global se puede considerar que la metodología CRISP-DM está más cercana al concepto real de proyecto, pudiendo ser integrada con una Metodología de Gestión de Proyectos específica que completaría las tareas administrativas y técnicas [9].

Otra diferencia significativa entre la metodología SEMMA y la metodología CRISP-DM radica en su relación con herramientas comerciales. La metodología SEMMA sólo es abierta en sus aspectos generales ya que está muy ligada a los productos SAS[1] donde se encuentra implementada. Por su parte CRISP-DM ha sido diseñada como una metodología neutra respecto

[1] SAS es la compañía líder de software de Business Analytics y servicios y el mayor proveedor independiente del mercado de Business Intelligence [8].

a la herramienta que se utilice para el desarrollo del proyecto de explotación de datos siendo su distribución libre y gratuita [9].

1.3.1. CRISP-DM

La metodología CRISP-DM es suficientemente amplia y flexible, y la más usada gracias a su fácil adaptación a proyectos de MD.

Las ventajas encontradas en esta metodología son:

- El proyecto de MD es visto de forma global y estrechamente relacionado al negocio en cuestión.
- Fue diseñada de forma neutra a la herramienta que se utilice para el desarrollo del proyecto, brindando la facilidad de uso con cualquiera de ellas.
- Es una metodología de distribución libre.
- Muchas de las metodologías que se pueden encontrar en la actualidad se basan en este estándar.
- Es la que cuenta con mayor aceptación por parte de los desarrolladores de procesos de extracción de conocimientos a partir de datos.

La metodología CRISP-DM estructura el ciclo de vida de un proyecto de MD en seis fases, cuya sucesión no es rígida, y se puede mover entre ellas siempre que se requiera [5]:

- Análisis del problema: fase inicial enfocada a entender los objetivos y requerimientos desde una perspectiva de negocio; para luego definirlos en términos de un problema de Minería de Datos y diseñar un plan para satisfacerlos.

- Comprensión de los datos: se hace una recolección y exploración inicial de los datos para familiarizarse con ellos e identificar problemas de calidad. Además, se trata de descubrir o estimar las relaciones más evidentes para formular las primeras hipótesis sobre información oculta en ellos.

- Preparación de los datos: esta fase cubre todas las actividades necesarias para construir la colección de datos que finalmente será minada a partir del grupo inicial. Incluye la colección, exploración, limpieza, transformación y construcción de datos.

- Modelado: durante esta fase se aplican varias técnicas de modelado. Comúnmente existen varias técnicas para resolver un problema de Minería de Datos del mismo tipo. Incluye la evaluación desde el punto de vista de precisión de los modelos.

- Evaluación: al llegar a esta fase se tendrán los modelos de mayor calidad desde la perspectiva de la precisión. Se impone una evaluación de los modelos y de los pasos que se siguieron para su construcción, a fin de determinar si responden apropiadamente a los objetivos de negocio que se determinaron en la primera fase. Es de vital importancia analizar si alguna regla del negocio no fue tomada en cuenta con el suficiente peso.

- Despliegue: en dependencia de los requerimientos y objetivos, la fase de despliegue puede ser tan simple como generar un reporte, o tan compleja como emprender un proceso de KDD de mayor envergadura. En ocasiones, son los clientes y no los desarrolladores quienes implementan esta fase; deben comprender cómo desarrollarla. Se hace imprescindible documentar y presentar los resultados de manera que todos los puedan entender.

Las fases del proyecto de minería de acuerdo a lo establecido por la metodología CRISP-DM interactúan entre ellas de forma iterativa durante el desarrollo del proyecto, formando una secuencia cíclica (Ver Anexo 4).

Se selecciona la metodología CRISP-DM a utilizar en la presente investigación, dadas las características y ventajas expuestas, además de que se obtienen proyectos de minería de alta calidad. Es una metodología que brinda una facilidad de uso con cualquier herramienta que se utilice para el proceso de minería. Se opta además por esta metodología porque permite acercarse a las necesidades reales del proyecto e identificar qué es lo que se quiere encontrar, adentrándose a los requerimientos del negocio en cuestión.

1.4. Modelos, técnicas y algoritmos de MD

El conocimiento obtenido con el proceso de minería, se puede representar a través de patrones o reglas inferidos a partir del análisis del mismo, o en forma más compacta a través de resúmenes. Estos resúmenes o relaciones constituyen el modelo de los datos analizados.

Para llegar a estos modelos existen diferentes técnicas de MD, entre las más utilizadas están: los árboles de decisión, clasificación, redes neuronales, agrupamiento o clustering y obtención de reglas de asociación.

1.4.1. Modelos de MD

Los modelos constituyen la forma de representar el conocimiento obtenido a partir de los datos analizados, y su construcción está determinada por la técnica de minería de datos escogida y el algoritmo seleccionado para realizarlo [3]. Dependiendo de las características de cada modelo de minería estos pueden clasificarse en predictivos o descriptivos.

Los modelos predictivos pretenden estimar valores futuros o desconocidos de variables de interés, que se denominan variables objetivo o dependientes, usando otras variables o campos, a las que se refiere como variables independientes o predictivas. Por ejemplo, un modelo predictivo sería aquel que permite estimar la demanda de un nuevo producto en función del gasto en publicidad [3].

Los modelos descriptivos, en cambio, identifican patrones que explican o resumen los datos, es decir, sirven para explorar las propiedades de los datos examinados, no para predecir nuevos datos. Por ejemplo, una agencia de viaje desea identificar grupos de personas con los mismos gustos, con el objeto de organizar diferentes ofertas para cada grupo y poder así remitirles esta información; para ello analiza los viajes que han realizado sus clientes e infiere un modelo descriptivo que caracteriza estos grupos [3].

1.4.2. Técnicas y algoritmos de MD

Las técnicas de MD son el resultado de un largo proceso de investigación y desarrollo de productos. Estas técnicas se dividen en dos grupos, las predictivas o supervisadas y las descriptivas o no supervisadas.

Las técnicas predictivas especifican el modelo para los datos en base a un conocimiento teórico previo. El modelo supuesto para los datos debe constatarse después del proceso de minería de datos antes de aceptarlo como válido. Se puede incluir entre estas técnicas todos los tipos de regresión, series temporales, análisis de la varianza y covarianza, análisis discriminante, árboles de decisión y redes neuronales. Pero, tanto los árboles de decisión, como las redes neuronales y el análisis discriminante son a su vez técnicas de clasificación que pueden extraer perfiles de comportamiento o clases, siendo el objetivo construir un modelo que permita clasificar cualquier nuevo dato [10]. (Ver Anexo 5).

En las técnicas descriptivas no se asigna ningún papel predeterminado a las variables. No se supone la existencia de variables dependientes ni independientes y tampoco se supone la existencia de un modelo previo para los datos. Los modelos se crean automáticamente partiendo del reconocimiento de patrones. En este grupo se incluyen las técnicas de *clustering* y segmentación (que también son técnicas de clasificación en cierto modo), las técnicas de asociación y dependencia, las técnicas de análisis exploratorio de datos y las técnicas de la reducción de la dimensión (factorial, componentes principales, correspondencias, etc.) [10]. (Ver Anexo 5).

Las predicciones se utilizan para prever el comportamiento futuro de algún tipo de entidad mientras que una descripción puede ayudar a su comprensión.

1.4.2.1. Clasificación

Las técnicas de clasificación permiten clasificar a un individuo nuevo dentro de un conjunto predefinido de clases. Existen diferentes enfoques dentro de las técnicas de clasificación. Todos ellos, sin embargo, construyen un modelo de clasificación a partir de un conjunto de individuos de

entrenamiento de los que se conocen ciertos atributos, incluyendo la clase a la que pertenecen. Ante un nuevo individuo sin clasificar, el modelo de clasificación determinará su clase haciendo uso del valor conocido de sus atributos [4].

1.4.2.2. Árboles de decisión

Los árboles de decisión pertenecen al conjunto de técnicas de clasificación que se basan en la construcción iterativa de un árbol. Estos permiten clasificar los datos en grupos basados en los valores de las variables. El mecanismo de base consiste en elegir un atributo como raíz y desarrollar el árbol según las variables más significativas.

Dado un conjunto de ejemplos de entrenamiento, se construye una partición del espacio de entrada y se asigna a cada región un determinado modelo. Luego, dado un nuevo dato, a partir de los valores de las variables de entrada se determina una región y el predictor del modelo construido le asigna un valor a la variable de salida [11].

Para clasificar individuos, los nodos de los árboles de decisión almacenan una condición sobre un determinado atributo del individuo a clasificar, mientras que las ramas son las que determinan a qué nodo del nivel inmediatamente inferior descender dependiendo de que se cumpla o no dicha condición. Este proceso se realiza hasta alcanzar un nodo hoja, que almacena la clase en la que es clasificado el individuo.

Los árboles de decisión son capaces de extraer una estructura que representa, en cierta medida, el concepto o el patrón de comportamiento que hay asociado a la muestra sobre la que se ha inducido. Una de las grandes ventajas de los árboles de decisión es que, en su forma más general, las opciones posibles a partir de una determinada condición son excluyentes. Esto permite analizar una situación y, siguiendo el árbol de decisión apropiadamente, llegar a una sola acción o decisión a tomar.

El proceso de creación de un árbol de decisión se debe ver como un proceso recursivo, primero se hace la selección de un atributo para ubicar en el nodo raíz y luego hacer una rama para cada

posible valor del atributo. Esto divide el conjunto en dos subconjuntos y se repite el proceso recursivamente para cada una de las ramas.

Los algoritmos de aprendizaje de árboles de decisión más habituales se llaman algoritmos de partición o algoritmos de “divide y vencerás”. Básicamente, el algoritmo va construyendo el árbol añadiendo particiones y los hijos resultantes de cada partición. Lógicamente, en cada partición, los ejemplos se van dividiendo entre los hijos. Finalmente, se llega a la situación en la que todos los ejemplos que caen en los nodos inferiores son de la misma clase y esa rama ya no continúa creciendo.

En la siguiente figura se puede observar un algoritmo básico para generar un árbol de decisión a partir de un conjunto de ejemplos, utilizando la técnica de “partición”.

```
ALGORITMO Partición (N:nodo, E:conjunto de ejemplos)
  SI todos los ejemplos E son de la misma clase c ENTONCES
      Asignar la clase c al nodo N.
      SALIR // Esta rama es pura, ya no hay que seguir partiendo. N es hoja.
  SI NO
      Particiones := generar posibles particiones.
      MejorPartición := seleccionar la mejor partición según el criterio de partición.
    PARA CADA condición i de la MejorPartición HACER
      Añadir un nodo hijo i a N y asignar los ejemplos consistentes a cada hijo (Ei).
      Partición(i, Ei).  // Realizar el mismo procedimiento global con cada hijo.
    FIN-PARA
  FIN-SI
FIN-ALGORITMO

Para generar un modelo con un conjunto de ejemplos E, se invoca con la llamada
Partición(R,E), donde R es un nodo raíz de un árbol por empezar.
```

Figura 2. Algoritmo Partición.

Una característica importante de estos algoritmos es que una vez elegida la partición ya no se puede cambiar, aunque más tarde se pudiera comprobar que ha sido una mala elección. Por tanto, uno de los aspectos más importantes a considerar en estos sistemas es el denominado criterio de partición.

Dentro de los algoritmos utilizados en la construcción de árboles de decisión se encuentra el ID3, a este le siguieron otros muy similares como C4.5 y C5.0. Existen otros algoritmos como, por ejemplo, CART, que genera como resultado un árbol binario que tiene, a lo sumo, dos hijos por nodo. Un algoritmo similar a CART es el denominado CHAID. La idea subyacente tras este algoritmo es la misma que la que se emplea en el algoritmo CART, con la diferencia de que el árbol no tiene por qué ser necesariamente binario, es decir, cada nodo puede tener un número de hijos mayor que dos. A continuación se exponen los aspectos más importantes de los algoritmos más utilizados en la técnica de árboles de decisión:

ID3

El algoritmo ID3 es uno de los primeros algoritmos propuestos para la construcción de árboles de decisión a partir de ejemplos de partida. A continuación se puede observar su funcionamiento:

1. Seleccionar el atributo Ai que maximice la ganancia, es decir el que tenga menor entropía[2].

2. Crear un nodo para ese atributo, con tantos sucesores como valores tenga.

3. Introducir los ejemplos en los sucesores según el valor que tenga el atributo Ai.

4. Por cada sucesor:

   SI sólo hay ejemplos de una clase ck.

   Entonces etiquetarlo con ck.

   SI NO, llamar al id3 con una tabla formada por los ejemplos de ese nodo, eliminando la columna del atributo Ai.

ID3 intenta encontrar el árbol más sencillo que separa mejor los ejemplos. Para ello utiliza la entropía para elegir o tomar decisiones.

[2] La entropía puede ser considerada como una medida de la incertidumbre y de la información necesaria para, en cualquier proceso, poder acotar, reducir o eliminar la incertidumbre.

Algoritmo C4.5

El algoritmo C4.5 fue desarrollado por JR Quinlan en 1993, como una extensión (mejora) del algoritmo ID3 que desarrolló en 1986. Se basa en la utilización del criterio ratio de ganancia. De esta manera se consigue evitar que las variables con mayor número de posibles valores salgan beneficiadas en la selección. Además el algoritmo C4.5 incorpora una poda del árbol de clasificación una vez que éste ha sido inducido. La poda está basada en la aplicación de un test de hipótesis que trata de responder a la pregunta de si merece la pena expandir o no una determinada rama [12].

```
Función C4.5
        (R: conjunto de atributos no clasificadores,
        C: atributo clasificador,
        S: conjunto de entrenamiento) devuelve un árbol de decisión;

Comienzo
        Si S está vacío,
                Devolver un único nodo con Valor Falla;
        Si todos los registros de S tienen el mismo valor para el atributo clasificador,
                Devolver un único nodo con dicho valor;
        Si R está vacío, entonces
                Devolver un único nodo con el valor más frecuente del atributo clasificador en los
                registros de S [Nota: habrá errores, es decir, registros que no estarán bien
                clasificados en este caso];
        Si R no está vacío, entonces
                D ← atributo con mayor Proporción de Ganancia(D,S) entre los atributos de R;
                Sean {dj| j=1,2, .., m} los valores del atributo D;
                Sean {Sj| j=1,2, .., m} los subconjuntos de S correspondientes a los valores de dj
                respectivamente;
                Devolver un árbol con la raíz nombrada como D y con los arcos nombrados d1,
                d2,.., dm que van respectivamente a los árboles
                C4.5(R-{D}, C, S1), C4.5(R-{D}, C, S2), .., C4.5(R-{D}, C, Sm);
Fin
```

Figura 3. Algoritmo C4.5.

El C4.5 construye un árbol de decisión mediante el algoritmo "divide y vencerás" y evalúa la información en cada caso utilizando los criterios de entropía, ganancia o proporción de ganancia.

Algoritmo CART

CART es muy simple, esta simplicidad no solo produce un algoritmo de clasificación muy rápido para clasificar nuevas observaciones, también conduce a un modelo mucho más simple para explicar por qué las observaciones se clasifican en un determinado grupo.

CHAID (Chi-square Automatic Interaction Detector)

Este algoritmo no realiza una fase de post-poda para evitar el sobre entrenamiento, sino que es en la misma fase de construcción del árbol cuando se decide parar. Sólo es capaz de tratar con variables predictoras discretas.

El CHAID analiza todos los valores de cada variable predictora potencial a través del Chi-cuadrado, el cual refleja cuan similares o asociadas están las variables. A partir de aquí, selecciona el predictor más significativo para formar la primera partición en el árbol de decisión, de tal forma que cada nodo está conformado por aquellas categorías similares de la variable seleccionada. El proceso continúa sucesivamente hasta que el árbol queda completado [13].

Algoritmo J48

Este algoritmo es una implementación del C4.5. Permite establecer ciertos parámetros, como obligar a realizar divisiones binarias sobre variables discretas, o cambiar el método de post-poda que utiliza el C4.5 por un método basado en la reducción de error. J48 admite tipos de atributos tantos simbólicos como numéricos.

Los árboles de decisión en ocasiones suelen ser demasiado grandes, por lo que resultan difíciles de entender ya que cada nodo debe ser interpretado dentro del contexto fijado por las ramas anteriores. Cada prueba tiene sentido, solamente, si se analiza junto con los resultados de las pruebas previas. Cada prueba en el árbol tiene un contexto único que es crucial a la hora de entenderla y puede ser muy difícil comprender un árbol en el cual el contexto cambia demasiado seguido al recorrerlo.

1.4.2.3. Redes bayesianas

La clasificación Bayesiana se basa en el teorema de Bayes, y los clasificadores Bayesianos han demostrado una alta exactitud y velocidad cuando se han aplicado a grandes bases de datos [4]. Las redes bayesianas modelan un fenómeno mediante un conjunto de variables y las relaciones de dependencia entre ellas. Dado este modelo, se puede hacer inferencia bayesiana; es decir, estimar la probabilidad posterior de las variables no conocidas, en base a las variables conocidas. Estos modelos pueden tener diversas aplicaciones, para clasificación, predicción, diagnóstico, etc. Además, pueden dar información interesante en cuanto a cómo se relacionan las variables del dominio, las cuales pueden ser interpretadas en ocasiones como relaciones de causa-efecto.

Las redes bayesianas se utilizan para clasificar, pero no dan reglas de cómo obtienen esta información. Otro inconveniente se centra en que las redes bayesianas trabajan con variables discretas. Además resulta de gran coste computacional al tener que calcular las funciones de distribución de todas las variables, y lo mismo ocurre en las redes neuronales cuando tienen que calcular los distintos pesos en cada nodo.

Naive Bayes

Naive Bayes es un algoritmo de clasificación y predicción que construye modelos que predicen la probabilidad de posibles resultados. Naive Bayes utiliza datos históricos para encontrar asociaciones y relaciones y hacer predicciones. Se utiliza para clasificar una nueva instancia de un documento D dentro de un conjunto finito C de clases.

Básicamente, este algoritmo busca correlaciones entre atributos. Cuando no se tiene muy claro qué atributo se puede predecir en función de otros, una técnica muy habitual es tratar de utilizar este algoritmo tratando de predecir el valor de los atributos en función de todos los atributos.

1.4.2.4. Redes de Neuronas Artificiales

Redes de neuronas artificiales es una técnica que modela computacionalmente el aprendizaje humano llevado a cabo a través de las neuronas del cerebro. Las redes de neuronas constituyen

una nueva forma de analizar la información con una diferencia fundamental respecto a las técnicas tradicionales: son capaces de detectar y aprender complejos patrones y características dentro de los datos [4]. Una red neuronal se compone de unidades llamadas neuronas conectadas entre sí. Cada neurona recibe una serie de entradas y proporciona una salida, que será la entrada de la siguiente neurona a la que está conectada. Las conexiones entre las neuronas tienen un peso que se usa para ponderar el valor que cada neurona transmite. Al recibir las entradas, cada neurona calcula la suma ponderada de sus entradas y, en la mayoría de los casos, aplica una función de activación para transformar el valor obtenido en una salida válida.

Las redes de neuronas se organizan en capas, todas las redes neuronales poseen una capa de entrada y una capa de salida y pueden tener las capas intermedias (u ocultas) que se desee, dando lugar a redes de diferentes arquitecturas. (Ver Anexo 6).

Inicialmente la red se ha de entrenar para calcular el peso de cada conexión. Para ello, se suministra a la red cada individuo del conjunto de entrenamiento (para los cuales la salida es conocida) y se ajustan los pesos de acuerdo a las diferencias encontradas entre la salida obtenida y la esperada. De esta forma, si las diferencias son grandes se modifica el modelo de forma significativa y, según van siendo menores, se converge a un modelo final estable.

El Perceptron Simple es una de las redes de neuronas más sencillas que se emplea en MD para resolver problemas de clasificación. En este tipo de red, que tiene dos capas, la entrada X (atributos conocidos) se transforma en una salida Y (variable respuesta, es decir, clase a la que se asigna el individuo), mediante una función no lineal. En el Perceptron, como el resto de redes neuronales, el objetivo de la red durante la fase de entrenamiento es aprender cuáles son los valores más adecuados para los pesos. Para realizar este proceso de aprendizaje existen múltiples algoritmos, pero el más empleado, al menos en MD, es el algoritmo Backpropagation. La principal limitación del Perceptron Simple es que sólo permite resolver problemas de clasificación binarios (dos clases). Este inconveniente se resuelve gracias al Perceptron Multicapa, que es una red neuronal formada por múltiples capas que no son linealmente separables.

Las Redes Neuronales deben ser aplicadas en situaciones en las que las técnicas tradicionales han fallado en dar resultados satisfactorios, o cuando una mejora en el modelado puede significar una diferencia en la eficiencia de la operación de un sistema, lo que lleva como consecuencia una mejora en la relación costo-beneficio.

Se debe considerar el uso de las Redes Neuronales cuando:

- El número de variables o la diversidad de los datos sean muy grandes.
- Las relaciones entre las variables sean vagamente entendibles.
- La relación entre estas variables sean difíciles de describir adecuadamente mediante los métodos convencionales.

Una de las desventajas de las Redes Neuronales es que se deben entrenar para cada problema. Además es necesario realizar múltiples pruebas para determinar la arquitectura adecuada. El entrenamiento es largo y puede consumir varias horas. Las Redes Neuronales presentan un aspecto complejo para un observador externo que desee realizar cambios. Para adicionar nuevo conocimiento, es necesario cambiar las interacciones entre muchas unidades para que su efecto unificado sintetice este conocimiento.

1.4.2.5. Agrupamiento o *Clustering*

Se puede definir el agrupamiento (*clustering* en inglés) como el proceso de dividir un conjunto de datos en grupos mutuamente exclusivos de tal manera que los miembros dentro de cada grupo están lo más cerca posible, mientras que diferentes grupos están lo más lejos posible. Definimos una distancia para medir la cercanía o lejanía en términos de todas las variables disponibles. [14].

A diferencia de las técnicas de clasificación, el *clustering* es un proceso de aprendizaje no supervisado ya que las clases no están predefinidas sino que deben ser descubiertas dentro de los datos.

Algoritmo K-medias

Entre los algoritmos de *clustering* está el K-medias, el cual es hasta ahora el más utilizado en aplicaciones científicas e industriales [15]. Para su implementación, primero se determina la

cantidad de clústeres o grupos que se quiere obtener y se seleccionan los ‘n’ elementos, o centroides de cada clúster aleatoriamente. Luego cada instancia se asigna al grupo más cercano teniendo en cuenta una medida de similitud dada. A continuación para cada clúster se obtienen los centroides de sus instancias, repitiéndose este procedimiento hasta que los centroides de los clúster se hayan estabilizado. K-medias es un algoritmo ávido cuyo objetivo es minimizar el error cuadrado entre la media[3] del grupo y sus elementos, en otras palabras, encontrar grupos donde en cada uno de ellos estén elementos semejantes y que los elementos de grupos diferentes no lo sean [16].

El pseudocódigo del algoritmo K-medias es el siguiente [4]:

1. Elegir k ejemplos que actúan como semillas (k número de clusters).
2. Para cada ejemplo, añadir ejemplo a la clase más similar.
3. Calcular el centroide de cada clase, que pasan a ser las nuevas semillas.
4. Si no se llega a un criterio de convergencia (por ejemplo, dos iteraciones no cambian las clasificaciones de los ejemplos), volver a 2.

Algoritmo Cobweb

El algoritmo de K-medias se encuentra con un problema cuando los atributos no son numéricos, ya que en ese caso la distancia entre ejemplares no está tan clara. Para resolver este problema Michalski presenta la noción de clustering conceptual, que utiliza para justificar la necesidad de un clustering cualitativo frente al clustering cuantitativo, basado en la vecindad entre los elementos de la población [4].

Este algoritmo es el Cobweb, el cual se caracteriza por utilizar aprendizaje incremental, esto es, realiza las agrupaciones instancia a instancia. Durante la ejecución del algoritmo se forma un árbol donde las hojas representan los segmentos y el nodo raíz engloba por completo el conjunto de datos de entrada.

[3] Una media o promedio es una medida de tendencia central de un conjunto de datos.

Al principio, el árbol consiste en un único nodo raíz. Las instancias se van añadiendo una a una y el árbol se va actualizando en cada paso. La actualización consiste en encontrar el mejor sitio donde incluir la nueva instancia, operación que puede necesitar de la reestructuración de todo el árbol o simplemente la inclusión de la instancia en un nodo que ya existía.

La clave para saber cómo y dónde se debe actualizar el árbol la proporciona una medida denominada utilidad de categoría, que mide la calidad general de una partición de instancias en un segmento, dando mayor valor a las clases que presentan una alta similitud entre sus miembros y una baja similitud con el resto de las clases. La reestructuración que mayor utilidad de categoría proporcione es la que se adopta en ese paso.

1.4.2.6. Reglas de Asociación

Las técnicas de asociación pretenden encontrar reglas que muestran la relación que existe entre distintas variables de los registros de una BD.

El descubrimiento de reglas de asociación busca relaciones o asociaciones entre conjuntos de ítems, donde un ítem es un par atributo-valor. Una regla de asociación está compuesta por dos conjuntos de ítems llamados premisa y conclusión, los cuales se unen mediante una flecha que va desde la premisa hacia la conclusión, X implica Y, siendo X y Y conjuntos de ítems. La conclusión siempre contiene un par atributo-valor.

Según la definición original de Agrawal [17] el problema de minería de reglas de asociación se define como:

Sea $I = \{i_1, i_2, \ldots, i_n\}$ un conjunto de n atributos binarios llamados ítems.

Sea $D = \{t_1, t_2, \ldots, t_m\}$ un conjunto de transacciones almacenadas en una base de datos.

Cada transacción en D tiene un ID (identificador) único y contiene un subconjunto de ítems de I. Una regla se define como una implicación de la forma:

$$X \Rightarrow Y$$

Donde:

$X, Y \subseteq I$ y $X \cap Y = \emptyset$

Los conjuntos de ítems X y Y se denominan respectivamente "antecedente" (o parte izquierda) y "consecuente" (o parte derecha) de la regla.

El tamaño de un conjunto de ítems está dado por su cardinalidad, un conjunto de ítems de cardinalidad k se denomina k-itemset.

Soporte y confianza

Los algoritmos de aprendizaje de reglas de asociación se basan en la búsqueda de reglas que cumplan unos requisitos mínimos de confianza, soporte o cobertura. Estas son las medidas más usadas en la literatura para evaluar la calidad de una regla de asociación.

Definición 1. (Soporte de un conjunto de ítems): El soporte de un conjunto de ítems X (sop(X)) en un conjunto de transacciones T, es la fracción de transacciones de T que contienen los ítems de X [18]. Sea el mínimo soporte (minSup) un umbral previamente establecido, un conjunto de ítems X se denomina frecuente si sop(X) ≥ minSup.

Definición 2. (Soporte de una regla): El soporte de una regla $X \Rightarrow Y$ en un conjunto de transacciones T, es la fracción de transacciones de T que contienen los ítems de **X ∪ Y** [17].

Definición 3. (Confianza de una regla): La confianza de una regla $X \Rightarrow Y$ es la fracción de transacciones de T que conteniendo a X, también contienen a Y [17].

A partir de las definiciones se concluye que el soporte de un conjunto de ítems X en una base de datos D se define como la proporción de transacciones en la base de datos que contiene dicho conjunto de ítems:

$$\mathrm{sop}(X) = \frac{|X|}{|D|}$$

Por otra parte la confianza es la probabilidad condicional de que un registro que contenga X también contenga Y. La confianza de una regla se plantea como:

$$\text{conf}(X \Rightarrow Y) = \frac{\text{sop}(X \cup Y)}{\text{sop}(X)} = \frac{|X \cup Y|}{|X|}$$

Las reglas de asociación permiten encontrar relaciones y asociaciones dentro del conjunto de datos, esta es una tarea descriptiva y constituye el eje fundamental de esta investigación. Por lo que se decide utilizar esta técnica en la búsqueda de reglas, patrones y tendencias que describan el comportamiento de los usuarios en el uso de las cuotas de navegación. Esta técnica busca por toda la base de datos, realizando una clasificación en cada barrido, obteniendo las características presentes en las transacciones realizadas, las cuales pueden tener atributos de diferentes tipos.

Algoritmos

Los algoritmos de asociación permiten la búsqueda automática de reglas que relacionan conjuntos entre sí. Son algoritmos no supervisados ya que no existen relaciones conocidas a priori con las que se puedan comprobar la validez de los resultados, sino que se evalúa si esas reglas son estadísticamente significativas mediante los valores del soporte y la confianza.

Partition

Este algoritmo recorre la base de datos sólo dos veces. Cada partición es minada independientemente para encontrar todos los conjuntos de ítems frecuentes en la partición y luego se mezclan éstos para generar el conjunto de los conjuntos de ítems candidatos. En la segunda pasada se cuenta la ocurrencia de cada candidato, aquellos cuyo soporte es mayor que el mínimo soporte especificado se retienen como conjuntos frecuentes.

Este algoritmo emplea el mecanismo de intersección entre conjuntos para determinar el soporte de dichos conjuntos, en este caso cada ítem en una partición mantiene la lista de los identificadores de las transacciones que contienen a dicho ítem. El costo en memoria es mayor que en el Apriori, pues requiere almacenar para cada ítem el conjunto de transacciones que lo contiene.

Eclat

Al igual que el Partition, este algoritmo reduce la cantidad de operaciones de entrada y salida, aunque esta vez atravesando la base de datos sólo una vez. Se basa en realizar un agrupamiento entre los ítems para aproximarse al conjunto de ítems frecuentes y luego emplean algoritmos eficientes para generar los ítems frecuentes contenidos en cada grupo. Este algoritmo presenta problemas de costo alto de memoria.

Para el agrupamiento propone dos métodos que son empleados después de descubrir los conjuntos frecuentes de dos elementos: el primero, por clases de equivalencia: esta técnica agrupa los itemsets que tienen el primer ítem igual. El segundo, por la búsqueda de cliques maximales: se genera un grafo de equivalencia cuyos nodos son los ítems, y los arcos conectan los ítems de los 2-itemsets frecuentes, se agrupan los ítems por aquellos que forman cliques maximales. Este algoritmo al basarse en el agrupamiento entre los ítems, interfiere en el tiempo de ejecución.

FP-Growth

La idea básica del algoritmo FP-Growth puede ser descrita como un esquema de eliminación recursiva: en un primer paso de pre-procesamiento se borran todos los ítems de las transacciones que no son frecuentes individualmente o no aparecen en el mínimo soporte de transacciones, luego se seleccionan todas las transacciones que contienen al menos un ítem frecuente, se realiza esto de manera recursiva hasta obtener una base de datos reducida. Al retorno, se remueven los ítems procesados de la base datos de transacciones en la memoria y se empieza otra vez, y así con el siguiente ítem frecuente. Los ítems en cada transacción son almacenados y luego se ordena descendentemente su frecuencia en la base de datos.

Después de que se han borrado todos los ítems infrecuentes de la base de datos de transacciones, se pasa al árbol FP. Un árbol FP es básicamente de prefijos para las transacciones, esto es: cada camino representa el grupo de transacciones que comparten el mismo prefijo, cada nodo corresponde a un ítem. Todos los nodos que referencian al mismo ítem

son referenciados juntos en una lista, de modo que todas las transacciones que contienen un ítem específico pueden encontrarse fácilmente y contarse al atravesar la lista.

Algoritmo Apriori

El algoritmo Apriori explora todas las transacciones cada vez que comprueba los k-itemsets candidatos. Este algoritmo encuentra las asociaciones más frecuentes e itera sobre la base de datos hasta que las asociaciones obtenidas no tienen el soporte mínimo, es un método simple pero robusto, y presenta una salida intuitiva. Los requisitos que cumple este algoritmo son los mencionados a continuación:

- No se necesita decir los atributos de los lados derecho (consecuente) e izquierdo (antecedente) de las reglas pues se generan de manera automática.
- Existen variedades para tratar todo tipo de datos.
- Especificar mínimo soporte.
- Especificar máximo número de reglas.

Apriori busca iterativamente conjuntos frecuentes con cardinalidad 1 hasta k, y después usa los conjuntos frecuentes para generar las reglas de asociación. En el paso clave del descubrimiento de conjuntos frecuentes, se basa en el principio Apriori, que expresa que cualquier subconjunto de un conjunto de artículos frecuente, debe ser frecuente, esto permite definir el principio de poda en Apriori: Si existe algún conjunto infrecuente, entonces no hay necesidad de generar sus superconjuntos.

El algoritmo Apriori se basa en una propiedad del mismo nombre que indica que si un conjunto de ítems de tamaño *n* no es frecuente, tampoco lo será ningún conjunto de ítems de tamaño n+1 que lo contenga. Usando esa propiedad, el algoritmo Apriori realiza una serie de iteraciones, de tal forma que, en el paso i-ésimo del algoritmo, se construye el conjunto L_i. La notación empleada para definir el algoritmo Apriori se presenta a continuación.

k-itemset	*Itemset* que tiene *k* ítems
L_k	Conjunto de *large k-itemsets* (aquellos que tienen un soporte mínimo). Cada miembro de este conjunto tiene dos elementos: i) *itemset* y ii) contador de soporte.
C_k	Conjunto de large k-itemsets candidatos (itemsets potenciales). Cada miembro de este conjunto tiene dos elementos: i) itemset y ii) contador de soporte.
C_t	Conjunto de *k-itemsets* candidatos cuando los identificadores de las transacciones generadas se guardan junto con los *itemsets* candidatos.

En la siguiente figura se muestra el -pseudo-código del algoritmo Apriori:

```
1. L1 = {large 1-itemsets}
2. Para (k=2; Lk-1 ≠ Ø; k++) hacer
   2.1. Ck = gen-apriori(Lk-1); // Nuevos candidatos
   2.2. Para todo registro t ∈ D hacer
      2.2.1. Ct = subconjunto (Ck,t); // Candidatos en t
      2.2.2. Para todo candidato c ∈ Ct hacer
         2.2.2.1. c.contador++;
   2.3. Lk = {c ∈ Ck | c.contador ≥ soporte_minimo }
3. Respuesta = ∪k Lk;
```

Figura 4. Algoritmo Apriori.

Selección del algoritmo

Para la selección del algoritmo a utilizar se determinaron criterios para la selección del algoritmo, estos fueron propuestos por los desarrolladores de AiresProxy y responden a las necesidades del proyecto.

Algoritmo	Consultas al conjunto de datos	Rendimiento
Apriori	Busca todos los conjuntos frecuentes unitarios contando sus ocurrencias directamente en la base de datos, por lo tanto se realizan varias pasadas en la BD.	Este algoritmo reduce el número de ítems que contienen subconjuntos no frecuentes, lo cual mejora el rendimiento.
Partition	Requiere de menos pasadas que el Apriori para el cálculo de los ítems frecuente a partir de la BD.	El costo en memoria es grande, pues requiere almacenar para cada ítem el conjunto de transacciones que lo contiene; además, el cálculo del soporte de un candidato obtenido por la unión de dos conjuntos frecuentes obliga a interceptar los dos conjuntos.
Eclat	Reduce la cantidad de operaciones de entrada/salida atravesando la BD sólo una vez.	El realizar tareas de agrupamiento requiere de pasos adicionales en su funcionamiento y por otro lado presenta problemas de costo alto de memoria similares al

		Partition.
FP-Growth	No requiere de la generación de candidatos, por lo tanto, precisa de pocos accesos a la BD.	Genera un árbol FP-Tree de una BD proyectada si el árbol inicial no se puede alojar completamente en la memoria principal. Cuando trabaja en conjuntos de datos muy dispersos con miles de transacciones, no aprovecha la compactación que brinda el FP-tree, resultando muy costosos los recorridos sobre el mismo.

El algoritmo seleccionado es el Apriori ya que es un método robusto para encontrar descripciones dentro de un conjunto de datos y reduce el número de ítems que contienen subconjuntos infrecuentes, lo que posibilita mejora en su rendimiento cuando se compara con otros algoritmos. Es además un algoritmo eficiente cuando trabaja con grandes cantidades de datos.

1.5. Herramientas para realizar análisis de datos

En la actualidad existe una gran cantidad de herramientas tanto libres como comerciales para el desarrollo de proyectos de MD. Estas herramientas son utilizadas para resolver problemas del mundo real en la ingeniería, la ciencia, los negocios y otros entornos. Son utilizadas para resolver situaciones donde el volumen de datos es muy grande por la cantidad de variables que se manipulan, o la extracción de conocimiento se hace compleja.

A continuación se expondrán algunas de las herramientas más completas y más utilizadas para la aplicación de la MD.

1.5.1. SPSS Clementine

SPSS Clementine[4] es una herramienta visual comercializada por SPSS que constituye uno de los sistemas más populares en el mercado. Esta herramienta fue comprada en julio del 2009 por la compañía IBM, por lo que pasó a conocerse como IBM SPSS Modeller.

Es un potente software que combina modernas técnicas de modelación con poderosas herramientas de acceso, manipulación y exploración de datos en una interfaz simple. Posibilita de forma rápida desarrollar y desplegar modelos que apoyen la toma de decisiones. Entre sus características a destacar está el hecho de que a diferencia de otras herramientas que se centran en el modelado, ella apoya el ciclo completo de KDD y está diseñada bajo la metodología CRISP-DM.

SPSS Clementine permite analizar, grandes volúmenes de datos, almacenados en grandes bases de datos transaccionales y/o registros de programas y cuenta con los métodos de Redes Neuronales de mayor uso: Kohonen, Prune y Radial Basis. Además incluye árboles de decisión, agrupamiento, reglas de asociación, regresión lineal y regresión logística, entre otras. Posee además un potente soporte gráfico que permite al usuario tener una visión global de todo el proceso, incluidos gráficos estadísticos en 3D y animados; así como visualizadores y navegadores [19].

Entre las técnicas y los algoritmos que implementa se encuentran:

- Árboles de Clasificación y Regresión
- Redes Neuronales
- C 5.0
- CHAID
- Regresión Lineal
- K- Medias
- Mapas Auto organizados o redes de Kohonen

[4] Sitio Web Oficial http://www.spss.com

- Apriori

SPSS Clementine es un software privativo y con grandes requerimientos de hardware, siendo estas sus principales desventajas.

1.5.2. SAS Enterprise Miner

Es una herramienta proporcionada por SAS Institute que agiliza el análisis de MD creando modelos descriptivos y predictivos de alta precisión basados en el análisis de una gran cantidad de datos. SAS Enterprise Miner [5] posee una arquitectura cliente/servidor, basado en Java, puede ser desarrollado tanto en la plataforma de Windows como en Linux.

Este software tiene una interfaz de usuario muy fácil de usar. Soporta el proceso de MD para crear modelos descriptivos y predictivos, sobre la base del análisis de las grandes cantidades de datos que tienen las empresas. Su diseño está inspirado en la metodología SEMMA [20].

Presenta la implementación de algoritmos que proveen modelos predictivos y descriptivos, tales como árboles de decisión, redes neuronales, asociación, agrupamiento, entre otros, además de tener incluido un potente visualizador gráfico para representar los resultados mediante gráficos en dos o tres dimensiones; así como un generador automático de reportes que resume los resultados en un informe HTML [19].

1.5.3. Yale

Yale es una herramienta creada en la universidad alemana de Dortmund para el descubrimiento del conocimiento y la minería de datos. Puesto que Yale está escrito enteramente en Java, funciona en las plataformas o sistemas operativos más conocidos, se retroalimenta de las librerías de funciones de Weka en su entorno de aprendizaje. Es un software de código abierto GNU y con licencia GPL[6] [21].

5 Sitio Web Oficial http://www.sas.com/
6 *GNU Public License*. http://www.gnu.org/copyleft/gpl.html

1.5.4. RapidMiner

RapidMiner[7] es la última versión de Yale, la cual incluye características como las de implicar nuevos formatos de entrada de datos con operadores para Microsoft Excel y SPSS. Es una herramienta flexible para aprender y explorar la MD. La interfaz gráfica de usuario tiene como objetivo simplificar el uso para las tareas complejas de esta área. Desde la perspectiva de la visualización ofrece representaciones de datos en dispersión en 2D y 3D.

1.5.5. Weka

Weka[8] es un software desarrollado bajo la licencia GPL de código abierto e incluye una interfaz gráfica compuesta por diversos entornos, desarrollada por un grupo de investigadores de la Universidad de Waikato de Nueva Zelanda. Se destaca por la cantidad de algoritmos que presenta así como por la eficacia de los mismos. Aunque la herramienta está implementada en Java no presenta problemas de portabilidad mientras que el sistema disponga de la máquina virtual adecuada, convirtiéndolo en un sistema multiplataforma.

Entre sus principales características se encuentra el poseer una interfaz gráfica de usuario compuesta de cuatro entornos que permiten diferentes funcionalidades y formas de análisis [4].

- Simple CLI: la interfaz "Command-Line Interfaz" es simplemente una ventana de comandos java para ejecutar las clases de Weka. La primera distribución de Weka no disponía de interfaz gráfica y las clases de sus paquetes se podían ejecutar desde la línea de comandos pasando los argumentos adecuados.

- Explorer: es la opción que permite llevar a cabo la ejecución de los algoritmos de análisis implementados sobre los ficheros de entrada, una ejecución independiente por cada prueba.

7 Sitio Oficial http://www.rapid-i.com/
8 Sitio Oficial http://www.cs.waikato.ac.nz/ml/weka/

- Experimenter: esta opción permite definir experimentos más complejos, con objeto de ejecutar uno o varios algoritmos sobre uno o varios conjuntos de datos de entrada, y comparar estadísticamente los resultados.

- KnowledgeFlow: esta opción es una novedad de Weka que permite llevar a cabo las mismas acciones del "Explorer", con una configuración totalmente gráfica, desde que se cargan los datos, se aplican algoritmos de tratamiento y análisis, hasta el tipo de evaluación deseada.

Weka emplea el formato ARFF (Atribute-Relation File Format) como soporte de datos, en el que cada uno de los ficheros consta de una lista de instancias con los mismos atributos. Los tipos de datos que permite son los numéricos, cadenas de caracteres, nominal y fecha.

Weka contiene diferentes tareas:

- Preprocesamiento: multitud de herramientas para el preprocesamiento de los datos, por ejemplo discretización de variables.

- Clasificación: algoritmos de clasificación, distribuidos por paquetes, por ejemplo ID3 y C4.5.

- Agrupamiento: diferentes algoritmos de segmentación como el simple K-medias.

- Asociación: algoritmos para encontrar relaciones de asociación entre variables como el Apriori.

- Selección de atributos: una vez cargados los datos, la herramienta es capaz de buscar las mejores variables del modelo.

- Visualización: herramienta de visualización de datos en los ejes cartesianos, con muchas posibilidades.

Como las principales ventajas de Weka están:

- Está disponible libremente bajo la licencia pública general de GNU.

- Es portable porque está completamente implementado en Java y puede correr en casi cualquier plataforma.
- Contiene una extensa colección de técnicas para preprocesamiento de datos y modelado.
- Es fácil de utilizar por un principiante gracias a su interfaz gráfica de usuario.

1.5.6. Elección de la Herramienta de MD a utilizar

Luego de realizarse un estudio de las herramientas de MD más utilizadas en el mundo, se puede observar que SPSS Clementine y SAS Enterprise Miner constituyen aplicaciones líderes en el mercado, basadas en las metodologías CRISP-DM y SEMMA, respectivamente; estas presentan la desventaja de que son herramientas comerciales, y su adquisición puede ser altamente costosa. Por otro lado, tanto Yale (RapidMiner) como Weka son aplicaciones de código abierto y de libre distribución, no comprometen su uso con una metodología en particular y son multiplataformas.

De acuerdo a las características de las aplicaciones antes descritas, se decidió para el desarrollo de esta investigación emplear una herramienta de código abierto, libre distribución e independiente de cualquier metodología, y en este sentido la comparación favorece a Weka y Yale (RapidMiner).

Se opta por la herramienta de análisis de datos Weka, que presenta a su favor muchas ventajas y que por su trayectoria ha demostrado su eficiencia en los procesos de MD.

Razones para seleccionar Weka:

Weka es un software específicamente diseñado y utilizado para investigación y fines educativos. Por esta razón, los elementos que brinda de salida, no están orientados exclusivamente hacia la obtención de herramientas para el dominio de aplicación, sino también hacia la obtención de información acerca del proceso de minería y de la calidad de los resultados obtenidos. Esta característica es importante para un proyecto de investigación como este.

Gracias a que se trata de una herramienta bajo el esquema de licenciamiento público, su uso es totalmente gratis, lo cual facilita su aprovechamiento para este proyecto de investigación.

Adicionalmente a ser una herramienta de uso libre, su código fuente (desarrollado en un lenguaje ampliamente difundido como JAVA) es abierto, lo que significa que no solo se puede hacer uso de los algoritmos implementados, sino también puede analizarse la implementación realizada de cada uno de ellos.

1.6. AiresProxy

AiresProxy es un producto que se desarrolló en la Universidad de las Ciencias Informáticas para obtener determinados reportes de los usuarios que navegan por internet a través de un servidor proxy, como los consumos asociados a los dominios, IP y fechas en que se realizaron las conexiones a internet. Cuenta con dos módulos principalmente, el primero un módulo desarrollado en c++ y qt4 para la lectura y procesamiento del log de navegación generado por Squid. Luego que los procesa inserta los datos en una base de datos. El otro módulo es una interfaz web para ver los reportes.

1.6.1. Funcionalidades

El sistema brinda diferentes funcionalidades desarrolladas con técnicas de programación tradicionales, para ello posee varias vistas que permiten generar los reportes. En cada una de las vistas el sistema muestra el rango de fecha en el que se realiza la monitorización así como los consumos totales reales y los calculados por las reglas de la UCI. Las principales funcionalidades que realiza el sistema son:

Calcular consumo de la Cuota UCI:

Se realiza un cálculo del consumo de la página analizada según las reglas establecidas por la UCI. Las reglas se aplican en dependencia del tipo de página analizada y los horarios de la visita. Existen cuatro tipos de páginas, las de interés, las de ocio, las nacionales y las irrelevantes. Las páginas de interés tienen un gasto de 0 bytes en el consumo de la cuota de los usuarios. En el

horario de 2 de la mañana a 7 de la mañana cualquier página independientemente de su tipo solo gasta la mitad de su gasto real. En los horarios picos (8:00 am – 6: 00 pm) las páginas de ocio gastan 1.5 del gasto real. Las páginas nacionales siempre gastan la mitad de su transferencia real.

Realizar reporte Dominios:

Para realizar el reporte general el usuario debe escoger un rango de fecha válido, inmediatamente el sistema genera el reporte con el dominio, consumo total, el gasto que le propicia el dominio en su cuota UCI y la categoría al cual pertenece dicho dominio (Ver Anexo 7).

Realizar reporte Dominios por IP:

Para realizar el reporte por IP el usuario debe escoger un rango de fecha válido, inmediatamente el sistema genera el reporte por IP, donde muestra el consumo total, el gasto que le propicia el dominio en su cuota UCI y el IP desde donde se accedió (Ver Anexo 8).

Realizar reporte Dominios por fecha:

Para realizar el reporte por fecha el usuario debe escoger un rango de fecha válido, inmediatamente el sistema genera el reporte por fecha, donde muestra el consumo total, el gasto que le propicia el dominio en su cuota UCI y la fecha de cuando se accedió (Ver Anexo 9).

Realizar reporte de Direcciones:

Para realizar el reporte por URL el usuario debe escoger un rango de fecha válido, inmediatamente el sistema genera el reporte con la URL, la fecha, el IP, el gasto que le propicia el dominio en su cuota UCI y el consumo real (Ver Anexo 10).

Reporte total por fecha:

El usuario debe escoger un rango de fecha válido, seguidamente el sistema genera un reporte donde aparecen las fechas con su consumo total en megas (Ver Anexo 11).

Reporte total por IP:

El usuario debe escoger un rango de fecha válido, seguidamente el sistema genera un reporte donde aparecen los IP con su consumo total en megas (Ver Anexo 12).

1.6.2. Sistema de base de datos MongoDB

MongoDB es una BD documental, surgida del movimiento *NoSQL* y que intenta incrementar la escalabilidad del sistema.

En MongoDB, cada registro o conjunto de datos se denomina documento. Los documentos se pueden agrupar en colecciones, las cuales se podría decir que son el equivalente a las tablas en una base de datos relacional, sólo que las colecciones pueden almacenar documentos con muy diferentes formatos en lugar de estar sometidos a un esquema fijo. Se pueden crear índices para algunos atributos de los documentos, de modo que MongoDB mantendrá una estructura interna eficiente para el acceso a la información por los contenidos de estos atributos.

La principal ventaja que tiene MongoDB ante las BD relacionales es la velocidad de consulta. Esto se logra gracias a que los documentos son almacenados en formato BSON, que es una versión modificada del JSON.

MongoDB es el sistema de BD utilizado actualmente por AiresProxy y es el que se decide a utilizar en la investigación para seguir una misma línea y que no se afecte la implementación del sistema.

1.7. Conclusiones

A partir de los resultados del estudio realizado en este capítulo se llegaron a las siguientes conclusiones:

- El proceso de descubrimiento de conocimiento está compuesto por varias fases donde la Preparación de Datos constituye un papel fundamental para las próximas etapas del proceso.

- La MD es un poderoso campo para la obtención de conocimiento a partir de grandes volúmenes de datos. Aplica diferentes modelos, técnicas y algoritmos para resolver los problemas que se deseen.

- Entre las técnicas que se estudiaron se decide utilizar las reglas de asociación con la cual se podrá obtener relaciones dentro del conjunto de datos, permitiendo conocer el comportamiento de los usuarios en el uso de las cuotas de navegación.

- El algoritmo Apriori es el seleccionado para la obtención de las reglas de asociación.

- Las herramientas disponibles permiten automatizar gran parte de la tarea de encontrar los patrones de comportamiento ocultos en los datos. Dentro de estas herramientas, Weka es la que se selecciona para apoyar el proceso de MD y experimentar los datos.

- Para guiar el minado de los datos se selecciona la metodología CRISP-DM.

- AiresProxy es un software que realiza reportes sobre la navegación de los usuarios, aplicando técnicas de programación tradicionales. El sistema no realiza ningún análisis inteligente sobre los datos.

Capítulo 2. Solución propuesta

En este capítulo se plantea cómo funciona el algoritmo Apriori, propuesto para la obtención de reglas dentro de la base de datos de AiresProxy. Para ello se propone la nueva estructura de la BD y el diseño de la misma necesaria para la aplicación del algoritmo y se describen las funcionalidades que se pueden implementar en AiresProxy con el uso de la técnica y el algoritmo seleccionados.

2.1. Funcionamiento del algoritmo Apriori

El Apriori para el descubrimiento de agrupaciones frecuentes utiliza múltiples lecturas de la base de datos. En la primera pasada, se cuenta el soporte de los objetos por separado y se determinan aquellos objetos con soporte mayor que el requerido. En cada pasada posterior de la base de datos se empieza con una semilla de agrupaciones que demostraron en la pasada anterior tener el soporte mínimo pedido y se utilizan para generar el conjunto de agrupaciones candidatas del siguiente nivel. A continuación, la base de datos se vuelve a leer para establecer los soportes de las agrupaciones candidatas, se eliminan todas aquellas con soportes inferiores al mínimo solicitado y el resto se convierten en la semilla de la siguiente pasada. El proceso continúa hasta que no se obtienen más agrupaciones.

A continuación se muestra un ejemplo que ilustra el proceso que realiza el algoritmo Apriori para encontrar conjuntos frecuentes donde el valor del mínimo soporte es 0.5.

C_k : atributos candidatos de tamaño k
L_k : atributos frecuentes de tamaño k
D: Base de datos de las transacciones.
soporte: número de apariciones que debe ser dividido entre la cantidad de transacciones para obtener el valor del soporte.

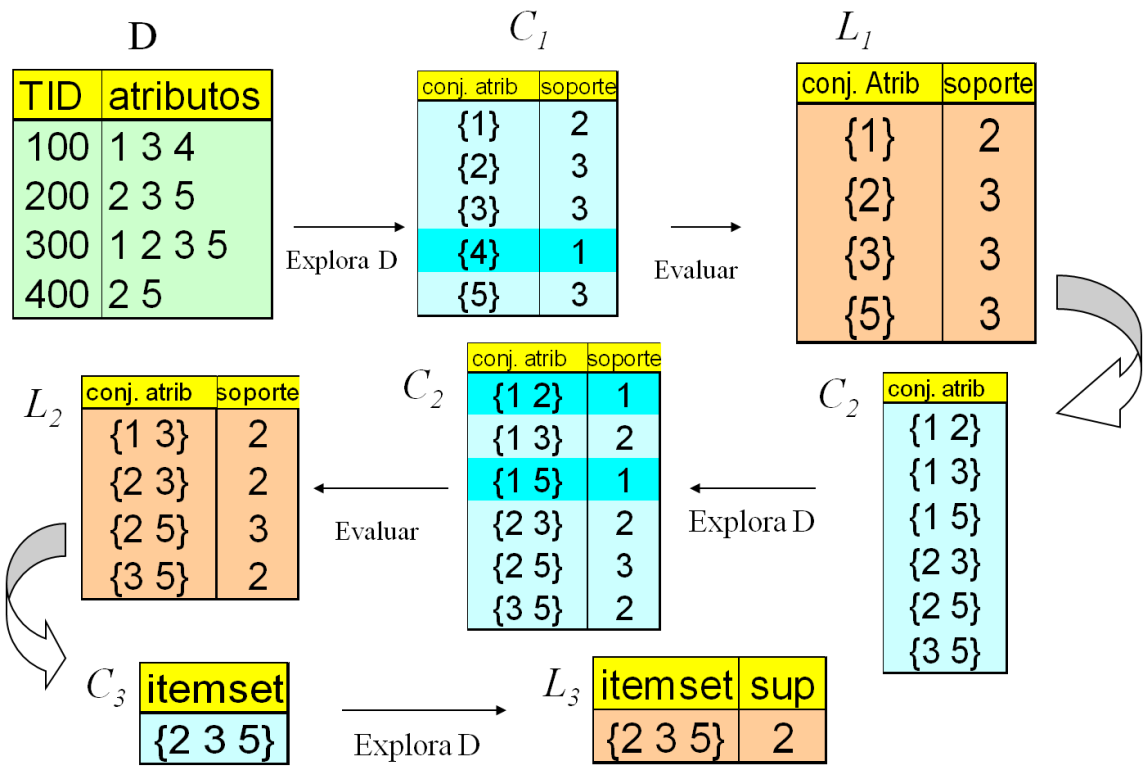


Figura 5. Funcionamiento del algoritmo Apriori.

2.2. Descripción de la solución propuesta

2.2.1. Estructura del sistema de base de datos a utilizar

La estructura de la BD del sistema AiresProxy que se propone utilizar para la inserción de técnicas y algoritmos de MD, incluye los datos que se obtienen el servidor proxy y otros generados a partir de atributos conocidos. A continuación se muestra la descripción de los datos.

Dato generado por Squid	Nombre del dato en la base de datos MongoDB	Observación
timestamp	ts	Dato que guarda la fecha y la hora en que se realiza la conexión. Se adiciona como un float.
transfer size	tr	Número de bytes trasferidos al

		cliente. Se adiciona como un float.
client address	ca	IP del host desde el cual se está haciendo la conexión. Se adiciona como un string.
URI	u	URI de la solicitud del cliente. Se adiciona como un string.
client identity	ci	Nombre del usuario que realizó la conexión. Se adiciona como un string.
-	tc	Dato calculado por la cuota de la UCI, determinado por las reglas de cálculo de la UCI puede ser 0, 1, 0.5 o 1.5. Se almacena 0 en caso de que se haya accedido a una página de interés. Se almacena 1 en caso de que se haya accedido a una página de irrelevante en horario de 8 am a 2 am. Se almacena 0.5 en caso de que se haya accedido a una página nacional o en horario 2 am a 7 am. Se almacena 1.5 en caso de que se haya accedido a una página de ocio en horario 8 am a 8 pm. Para calcular el consumo calculado en los reportes se multiplica este

		parámetro por los bytes trasferidos (transfer size). Se adiciona como un float.
-	cl	(Clasificación o categoría de la página) Se clasifica el tipo de página accedida. Los datos guardados en la BD son los siguientes: irrelevante, ocio, interés y nacional.
-	facultad	Dato referente a la facultad desde donde se realiza una conexión, se determina según el IP cual es la subred a la que pertenece. Se adiciona como un string.
-	area	Dato referente al lugar desde donde se realiza una conexión, se determina según el IP cual es la subred a la que pertenece. Los lugares pueden ser: laboratorio docente, laboratorio de producción, residencia y otros.
-	d	Dato que guarda el dominio. Se adiciona como un string.
	rango_ip	Dato que representa un rango de IP. Se adiciona como un string.
	id_ip, id_d, id_cip, id_cd	Identificadores de registro para una colección. Se adicionan como un tipo de dato float.

result/status codes	rs	Consta de dos fichas separadas por una barra, la primera se refiere al código identificador, clasifica el protocolo y el resultado de una transacción, por ejemplo: TCP_HIT o TCP_DENIED. La segunda indica la respuesta HTTP de código de estado, ejemplo: 200, 304 o 404. Se adiciona como un string.
request method	rm	Método de solicitud. Dado que los clientes pueden utilizar Squid ICP o HTTP, el método más común de solicitud HTTP es GET. Se adiciona como un string.
response time	rt	Indica la cantidad de tiempo que se tardó en procesar la solicitud. Se adiciona como un float.
peering code/peerhost	p	La información consta de dos fichas que describen la interconexión. La primera indica el nombre del host y la segunda la dirección del host. Se adiciona como un string.
content type	ct	Tipo de contenido de la respuesta HTTP. Se adiciona como un string.
HTTP request headers	hq	Squid codifica las cabeceras de petición HTTP y los imprime entre un par de corchetes. Se adiciona

		como un string.
HTTP response headers	hp	Squid codifica los encabezados de respuesta HTTP y los imprime entre un par de corchetes. Se adiciona como un string.

La estructura actual de la BD de AiresProxy incluye dos colecciones, estas son:

proxy.domains: guarda los dominios y su categoría. Esta colección se crea a partir del atributo "uri" que identifica la URI de la solicitud del cliente. Aquí se registra el dominio, la clasificación de la página accedida y el identificador de la transacción.

proxy.logs: guarda la información de la visita. En esta colección se guardan los datos que genera el servidor proxy y la referencia a la colección proxy.domains.

Con el objetivo de obtener mejores resultados al explorar el conjunto de datos se proponen nuevos campos, esto impone un cambio en la estructura actual, en este sentido se plantea para el sistema gestor de BD MongoDB mantener las dos colecciones anteriores y añadir una nueva colección:

proxy.ip: esta colección se crea a partir del atributo "ip" que identifica el IP del host desde el cual se está haciendo la petición, y es utilizado para obtener la facultad, el área desde donde se realiza la petición y el rango IP al que pertenece. Se registra también el identificador de la transacción.

proxy.logs	**proxy.ip**	**proxy.domains**
u: (string)	facultad: (string)	d: (string)
ts: (float)	area: (string)	cl: (string)
ca: (string)	rango_ip: (string)	id_cd: (float)
ci: (string)	id_cip: (float)	

tc: (float)		
rs: (string)		
rm: (string)		
tr: (float)		
p: (string)		
ct: (string)		
hq: (string)		
hp: (string)		
rt: (float)		
id_ip (float)		
id_d (float)		

La estructura centra los atributos que produce el servidor proxy en una única colección, separando los nuevos campos propuestos en colecciones diferentes. Con este diseño se consigue una mejor organización al extender la descripción de nuevos datos, facilitando la generación o modificación de los elementos en caso de ser necesario.

Los campos categoría, dominio y cuota, el sistema AiresProxy actualmente los determina. Para los atributos que pertenecen a la colección “proxy.ip”, se propone que los registros cubran todos los rangos de IP con su identificador correspondiente de manera predeterminada. Esto permite mayor velocidad al realizar peticiones a la base de datos y disminuye el incremento de almacenamiento.

Diseño propuesto de la BD:

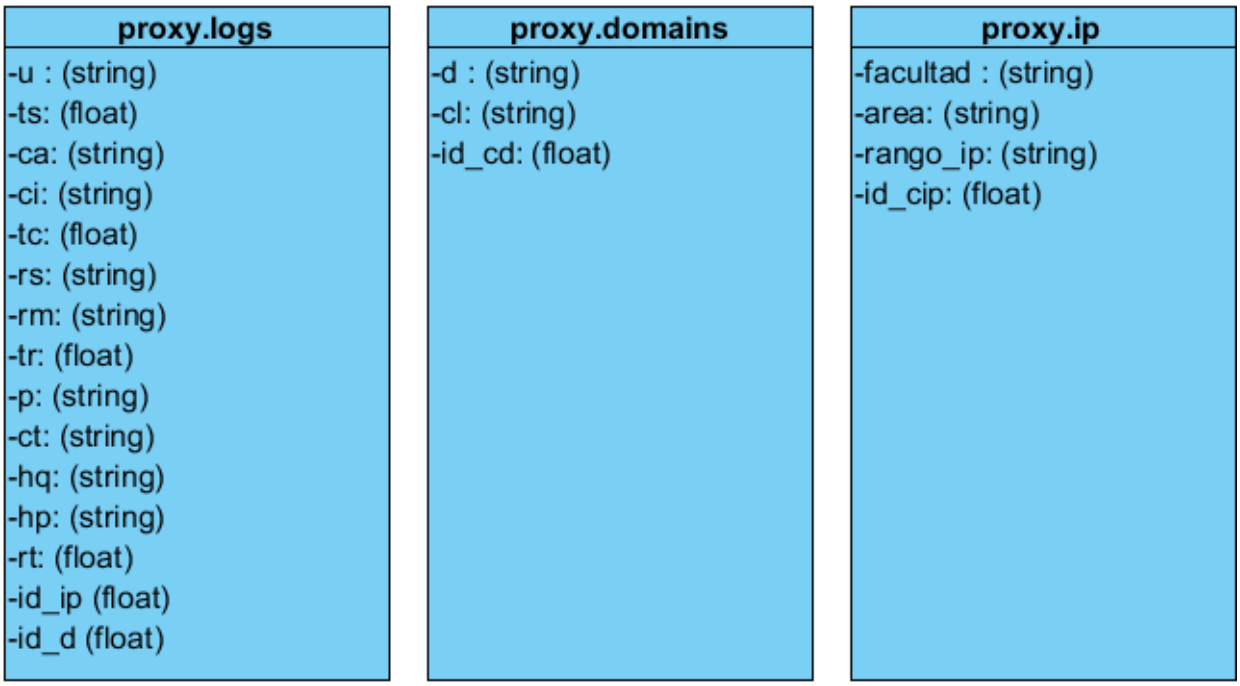


Figura 6. Diseño de la BD.

2.2.2. Funcionalidades propuestas

Aplicando la técnica de Reglas de Asociación y el algoritmo Apriori sobre el conjunto de datos que implementa MongoDB se pueden obtener relaciones diversas.

A partir de las necesidades identificadas por la dirección de redes y el equipo de desarrollo de AiresProxy, para buscar una mejor descripción de los usuarios en el uso de las cuotas que permita definir nuevas políticas de navegación, las funcionalidades propuestas para el sistema AiresProxy son las siguientes:

Relación de los usuarios que tienden a conectarse a sitios clasificados en un determinado horario: se realiza un filtrado escogiendo los atributos usuarios, clasificación y horario. Permite conocer los usuarios que más se conectan a un tipo de página en un determinado horario. Por

ejemplo se puede saber los usuarios que se conectan más a sitios de ocio en horario de 8 am a 12 m.

De un usuario conocer la categoría de los sitios a los que tiende a conectarse: se escoge el usuario especificado con el campo clasificación para conocer de un usuario a que sitios frecuentemente tiende a conectarse. Con este reporte se puede determinar por ejemplo si el usuario X se conecta más a sitios irrelevantes que a sitios de interés.

De un usuario conocer la categoría de los sitios a los que tiende a conectarse, desde las diferentes áreas de la universidad: se realiza un filtrado escogiendo los registros de un usuario especificado con los campos clasificación y área. Un ejemplo claro es saber si el usuario X se conecta más a sitios de ocio desde la residencia o si se conecta más a sitios nacionales desde el laboratorio de producción.

De un usuario conocer la categoría de los sitios a los que tiende a conectarse, desde las diferentes áreas de la universidad en un determinado horario: el usuario y el horario especificado con el campo clasificación y área, permite identificar, por ejemplo, si un usuario X se conecta más a sitios de ocio desde la residencia en horario de la noche o si se conecta más a sitios nacionales desde el laboratorio de producción en el horario de la mañana.

De un usuario conocer los dominios a los que tiende a conectarse: se realiza un filtrado escogiendo los registros del usuario especificado con el campo dominio. Con esto se pueden determinar los dominios a los que tiende conectarse el usuario X.

De un usuario conocer los dominios a los que tiende a conectarse en un determinado horario: se realiza un filtrado escogiendo los registros del usuario y el horario especificados, con el campo dominio. Con esto se pueden identificar los dominios a los que tiende conectarse el usuario X en un horario determinado.

De un usuario conocer los dominios a los que tiende a conectarse, desde las diferentes áreas de la universidad: se realiza un filtrado escogiendo los registros del usuario, con el campo

dominio y área. Con esto se pueden conocer los dominios a los que tiende conectarse el usuario X desde un área de la universidad.

De un usuario conocer los dominios a los que tiende a conectarse, desde las diferentes áreas de la universidad en un determinado horario: se realiza un filtrado escogiendo los registros del usuario y el horario especificado, con el campo dominio y área. Con esto se pueden conocer los dominios a los que tiende conectarse el usuario X desde un área de la universidad en dependencia del horario establecido.

Dado un dominio conocer el horario en que se realizan más accesos: se realiza un filtrado escogiendo los registros del dominio especificado con los horarios correspondientes. Un ejemplo es saber en qué horario acceden más los usuarios al dominio facebook.com.

Dado un dominio conocer el horario en que se realizan más accesos desde las diferentes áreas de la universidad: se realiza un filtrado escogiendo los registros del dominio especificado con los horarios y áreas correspondientes. Se puede determinar en qué horario acceden más los usuarios al dominio facebook.com y desde qué área.

De una facultad conocer la tendencia de acceso a los sitios según su categoría: se realiza un filtrado escogiendo los registros de una facultad y la clasificación de los sitios. Conocer cuáles son los tipos de sitios a los que se acceden con mayor frecuencia desde la facultad X.

Conocer los dominios que más demoran en procesar la solicitud: se realiza un filtrado escogiendo las transacciones del campo dominio y tiempo de respuesta. Por ejemplo, se identifica que el dominio twitter.com es uno de los dominios que más demora en procesar la solicitud.

Las funcionalidades propuestas tienen como propósito obtener una descripción más amplia sobre el comportamiento de los usuarios en el uso de las cuotas de navegación, realizando análisis exploratorios, buscando tendencias y relaciones dentro de los atributos. El conocimiento que se obtiene a partir de los reportes planteados, contribuye a la toma de decisiones más efectivas por parte de la DRSI de la universidad.

2.3. Conclusiones

Al finalizar este capítulo se han llegado a las siguientes conclusiones:

- Con el algoritmo Apriori es posible obtener reglas de asociación que pueden ser utilizadas en AiresProxy para la implementación de las nuevas funcionalidades.
- La nueva estructura de la BD incorpora nuevos atributos con lo que se extiende la descripción de los datos.
- Las funcionalidades propuestas brindan información implícita y valiosa sobre el uso que le dan los usuarios a las cuotas de navegación, estas pueden ser usadas por la DRSI para implementar un sistema de toma de decisiones más eficaz.

Capítulo 3. Experimentación con WEKA

En este capítulo se realiza una descripción de los pasos y actividades propuestos por la metodología CRISP-DM para realizar el proceso KDD, así como una descripción de los resultados de cada una de las fases con el uso de la herramienta Weka en su versión 3.7.5, la cual cuenta con varios entornos de trabajo, entre ellos está el Explorer que se utilizó para dar solución al objetivo de la investigación.

Los datos escogidos para realizar el proyecto pertenecen a la base de datos MongoDB que utiliza el sistema AiresProxy y se agregaron otros que corresponden a la propuesta de esta tesis para enriquecer el aporte de la misma.

Los datos de entrada a la herramienta, sobre los que operan las técnicas de MD, deben estar codificados en un formato específico, denominado Attribute-Relation File Format (extensión "arff"). La herramienta permite cargar los datos en tres soportes: fichero de texto, acceso a una base de datos y acceso a través de internet sobre una dirección URL de un servidor web. En esta investigación se trabaja con ficheros de texto. Los datos deben estar dispuestos en el fichero de la forma siguiente: cada instancia en una fila y con los atributos separados por comas. El formato de un fichero arff sigue la estructura siguiente:

% comentarios

@relation NOMBRE_RELACION
@attribute r1 real
@attribute r2 real ...
...
@attribute i1 integer
@attribute i2 integer
…
@attribute s1 {v1_s1, v2_s1,…vn_s1}
@attribute s2 {v1_s1, v2_s1,…vn_s1}
…

@data

DATOS

Por tanto, los atributos pueden ser principalmente de dos tipos: numéricos de tipo real o entero (indicado con las palabra real o integer tras el nombre del atributo), y simbólicos, en cuyo caso se especifican los valores posibles que puede tomar entre llaves.

3.1. Análisis del problema

3.1.1. Comprensión del negocio

En esta primera fase se analizan los factores determinantes en el proyecto por lo que se deben conocer los objetivos y requisitos del negocio, y a partir de estos definir el problema que permita alcanzar los objetivos del proyecto y producir resultados que cumplan con las expectativas de los clientes.

Como se describió en la situación problémica presentada, para los administradores de redes resulta importante conocer el comportamiento sobre el uso de las cuotas de acceso a Internet y en función del conocimiento obtenido, implementar políticas de navegación que aporten a la producción de la universidad. Con esta idea se implementó el sistema AiresProxy que permite mostrar varios reportes pero sin el uso de técnicas de MD. En la estructura propuesta para la BD de AiresProxy se encuentran recogidos datos como usuario, facultad, área desde donde se realizó la conexión, clasificación de la página accedida, entre otros. Sobre estos datos no se ha hecho ningún estudio que permita conocer información interesante sobre el uso de la cuota de acceso a Internet.

3.2. Comprensión de los datos

3.2.1. Recopilar los datos iniciales

Los datos de interés para realizar el proceso de MD se tomaron de la BD de AiresProxy a partir de los registros de navegación por Internet archivados en el servidor proxy y otros datos propuestos. Los administradores de red de la UCI tienen asignados segmentos de IP para cada

facultad y área desde donde se realiza la conexión, esta característica se utilizó para obtener estas variables a partir del campo IP. Otro dato que se obtiene es la clasificación o categoría de la página (ocio, irrelevante, de interés y nacional) y el dominio al que pertenece.

3.2.2. Descripción de los datos

A continuación se muestran los datos utilizados para el desarrollo del caso de estudio, se tiene en cuenta los campos de la base de datos de AiresProxy.

Tabla de atributos de la BD de MongoDB.

Nombre del atributo	Descripción	Tipo de dato
u	URI de la solicitud del cliente.	Cadena
ca	IP de la PC que realiza la petición.	Cadena
ci	Usuario que realiza la petición.	Cadena
tc	Cuota utilizada al acceder a una página.	Numérico
ts	Secuencia de caracteres que denotan la hora y fecha.	Numérico
rt	Tiempo que se tardó en procesar la solicitud.	Numérico
tr	Número de bytes trasferidos al cliente.	Numérico
rs	Código identificador y resultado de una transacción	Cadena
rm	Método de solicitud	Cadena
p	Información de interconexión	Cadena
ct	Tipo de contenido	Cadena
hq	Cabeceras de petición HTTP	Cadena
hp	Cabeceras de respuesta HTTP	Cadena
id_ip	Identificador relacionado con la colección que contiene el atributo	Numérico

	facultad.	
id_d	Identificador relacionado con la colección que contiene el atributo dominio.	Numérico

Tabla de los atributos que se proponen integrar a la BD de AiresProxy:

Nombre del atributo	Descripción	Tipo de dato
facultad	Facultad desde donde se realiza la petición.	Cadena
area	Lugar desde donde se realiza la petición.	Cadena
rango_ip	Rango de IP al que pertenece la PC que realizó la petición.	Cadena
id_cip	Identificador de registro	Numérico

Tabla de atributos integrada a AiresProxy:

Nombre del atributo	Descripción	Tipo de dato
d	Se refiere al dominio visitado.	Cadena
cl	Clasificación de la página	Cadena
id_cd	Identificador de registro	Numérico

3.2.3. Explorar los datos

La exploración de los datos es necesaria para comprender los mismos y poder tomar las decisiones apropiadas antes de generar los modelos. Para ello se podrán realizar consultas,

reportes y vistas para consolidar el objetivo final del proyecto de minería. Los resultados más significativos del reporte exploratorio de los datos por campos son:

- El caso de estudio utiliza 100 registros, donde están involucrados 5 usuarios y se comportan como muestra la gráfica.

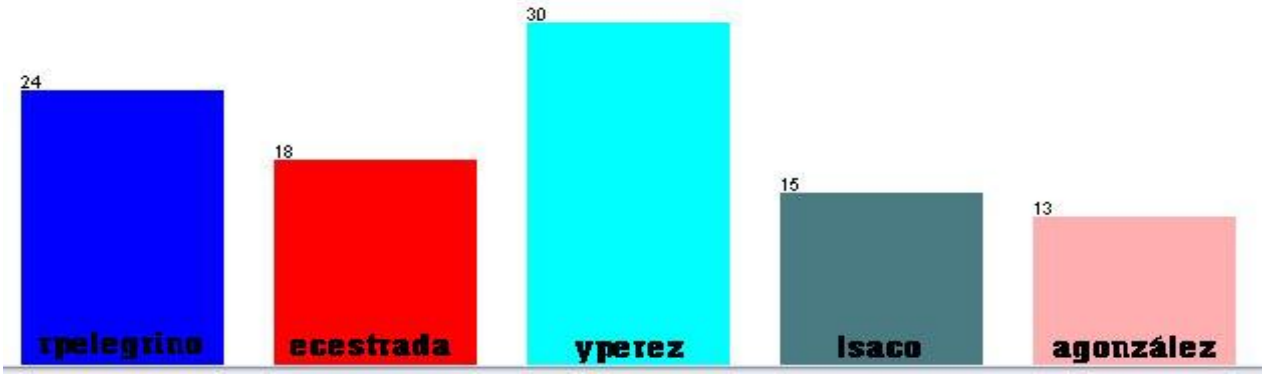


Figura 7. *Relación de usuarios por cantidad de registros*

- El 30% de los accesos a internet se realizan a sitios de ocio.
- El 30% de los accesos a internet se realizan a sitios nacionales.
- El 16% de los accesos a internet se realizan a sitios irrelevantes.
- El 24% de los accesos a internet se realizan a sitios de interés.
- El 37% de los accesos a internet se realizan desde la facultad 1.
- El 34% de los accesos a internet se realizan desde la facultad 2.
- El 29% de los accesos a internet se realizan desde la facultad 6.
- El 15% de los accesos a internet se realizan en el horario de la mañana y el almuerzo.
- El 23% de los accesos a internet se realizan en el horario de tarde.
- El 11% de los accesos a internet se realizan en el horario de comida.
- El 26% de los accesos a internet se realizan en el horario de la noche.
- El 10% de los accesos a internet se realizan en el horario de la madrugada.
- El 50% de los accesos a internet se realizan en el horario de la mañana.

Los dominios utilizados presentan el siguiente comportamiento:

	Dominio	Por ciento
1	youtube.com	14 %
2	facebook.com	9 %
3	twitter.com	4 %
4	taringa.net	3 %
5	wikipedia.com	12 %
6	patriagrande.com.ve	9 %
7	juventudrebelde.cu	3 %
8	ecured.cu	9 %
9	cubadebate.cu	3 %
10	ain.cu	7 %
11	prensalatina.cu	11 %
12	proverbia.net	7 %
13	mascotas.com	9 %

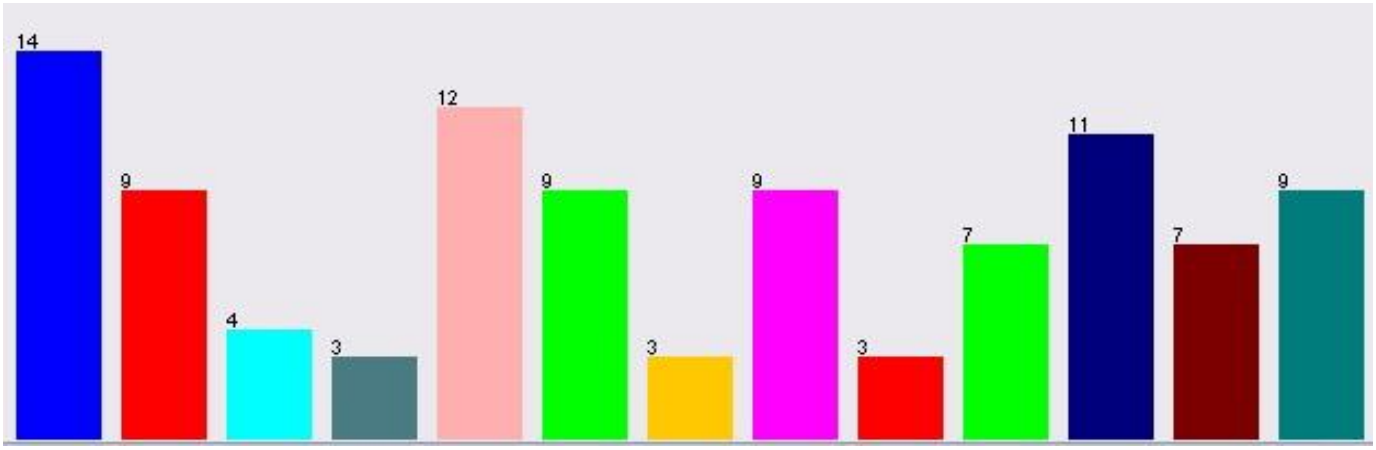


Figura 8. Relación del uso de de los dominios según la tabla anterior

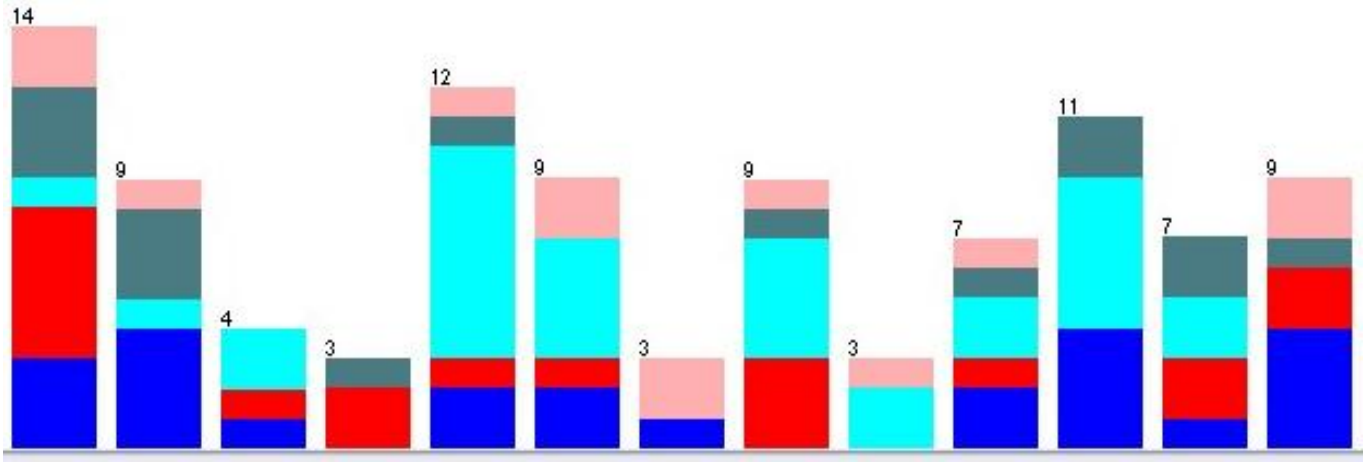


Figura 9. Relación entre usuarios y uso de los diferentes dominios

- Los datos representan 3 días de trabajo, el primero y el segundo se produjeron el 33 % de los accesos a internet cada uno y el tercero el 34 %.
- El 45% de los accesos a internet se realizan desde los laboratorios docentes y el 55% desde los laboratorios de producción.

Luego del análisis se verificó la calidad de los mismos, comprobando la existencia de valores nulos o incompletos. Se detectó que de las cuatro áreas identificadas, hay dos que no representan valores de conexión, como se muestra en la siguiente imagen.

Figura 10. Relación de los tipos de áreas.

3.3. Preparación de los datos

En la preparación de los datos se realizan actividades para obtener conjuntos de datos. Se aplican transformaciones y limpieza a los atributos con el propósito de obtener la calidad necesaria y aplicar las herramientas de modelación. Es primordial comprender bien los datos y analizar la influencia e importancia de los mismos de acuerdo a los objetivos propuestos, pudiéndose eliminar o agregar datos.

Luego de analizar los campos estudiados se puede determinar si es necesario adicionar nuevas columnas e introducir nuevos registros para construir los datos. En este sentido se realiza un proceso de discretización para poder aplicar algoritmos de MD, el cual consiste en limitar los posibles valores de un atributo, para convertir una variable continua en una variable discreta.

Hay dos razones principales que motivan la discretización:

- El algoritmo Apriori no acepta atributos numéricos para su aplicación, por lo que todos los atributos de este tipo deben ser convertidos.

- Sin embargo, no es suficiente con convertir estos atributos a texto, pues generar por ejemplo un árbol a partir de estos atributos, generaría tantas clasificaciones como valores existan y por lo tanto, habría una cantidad enorme de clasificaciones, lo cual no brinda utilidad a la hora de su aplicación.

Por tanto, se utilizó discretización en el siguiente caso:

En una primera observación se identifica la necesidad de transformar el atributo que representa la secuencia de caracteres de la fecha y la hora, y separarlos para mostrar resultados más comprensibles. Luego la hora en que fue realizada la solicitud, se transforma mediante una discretización en una variable nominal y se define el atributo llamado horario. De esta forma, se conforman las siguientes etiquetas: horario de mañana (8 am - 12 m), horario de almuerzo (12 m - 2 pm), horario de tarde (2 pm - 5 pm), horario de comida (5 pm - 8 pm), horario de noche (8 pm - 2 am) y horario de madrugada (2 am - 8 am). Para agregar la fecha se utilizó el filtro Add.

Así mismo, se utilizó el filtro Remove de la herramienta Weka para eliminar los siguientes campos:

rango_ip: su utilidad consiste en ubicar la facultad y el área desde donde se realiza una petición dado un IP.

u: a partir de este atributo se obtiene el campo dominio que representa la parte más significativa del identificador mencionado.

Los campos: rs, p, ct, hq, hp, id_ip, id_d, id_cd son eliminados ya que no son necesarios para responder a las funcionalidades propuestas.

En el análisis de MD es conveniente realizar transformaciones sobre el conjunto de datos con el fin de mejorar la precisión de los modelos de aprendizaje que se desarrollarán posteriormente.

3.3.1. Integrar los datos

Con la realización de esta tarea se analizan los datos y se combinan en el caso de que se encuentren en fuentes diversas. Luego se prosigue a la integración de la información que se extrae de tablas diferentes y se crea una o más tablas que contienen información útil sobre los mismos objetos. Además puede que se generen nuevos registros o columnas que generalicen la información de múltiples tablas.

La distribución de los campos se encuentra en varias colecciones, como se analizó en la fase de comprensión de los datos. Con el objetivo de agrupar estos datos se unieron todas las variables objetivo de la minería en una sola tabla. Para una mejor comprensión de los datos, en el archivo arff se utilizaron como nombre de los campos, otros que se acercan más a la descripción de los mismos.

En la siguiente figura se muestra una parte de los datos para analizar:

No.	1: fecha Nominal	2: ip Nominal	3: dominio Nominal	4: categoria Nominal	5: usuario Nominal	6: cuota Nominal	7: facultad Nominal	8: area Nominal	9: tiempo_r Nominal	10: bytes_t Nominal	11: **horario** Nominal
1	14-01-2012	10.33.20	youtube.com	ocio	rpelegrino	2.21Mb	F1	labdocente	12	40	mañana
2	13-01-2012	10.33.22	twitter.com	ocio	ecestrada	45.25Kb	F1	labproducción	7	35	almuerzo
3	15-01-2012	10.34.12	ecured.cu	nacional	ecestrada	1.2Mb	F2	labproducción	8	10	comida
4	14-01-2012	10.35.16	cubadebate.cu	nacional	yperez	92.04Kb	F6	labproducción	5	10	tarde
5	14-01-2012	10.33.21	mascotas.com	irrelevante	agonzález	45.25Kb	F1	labproducción	12	37	madrugada
6	15-01-2012	10.34.16	juventudrebel...	interes	rpelegrino	12Kb	F2	labdocente	7	20	comida
7	13-01-2012	10.35.18	prensalatina.cu	nacional	yperez	1.2Mb	F6	labdocente	8	20	almuerzo
8	14-01-2012	10.33.23	taringa.net	ocio	lsaco	92.04Kb	F1	labdocente	5	35	madrugada
9	15-01-2012	10.35.15	wikipedia.com	interes	yperez	1.2Mb	F6	labdocente	7	20	tarde
10	15-01-2012	10.34.14	patriagrande....	interes	ecestrada	12Kb	F2	labproducción	12	37	tarde
11	13-01-2012	10.34.10	facebook.com	ocio	rpelegrino	1.2Mb	F2	labdocente	12	40	noche
12	14-01-2012	10.35.17	ain.cu	nacional	yperez	45.25Kb	F6	labproducción	12	10	noche
13	13-01-2012	10.33.22	youtube.com	ocio	ecestrada	2.21Mb	F1	labproducción	8	40	noche
14	14-01-2012	10.33.20	twitter.com	ocio	rpelegrino	1.2Mb	F1	labdocente	8	35	comida
15	15-01-2012	10.33.21	ecured.cu	nacional	ecestrada	92.04Kb	F1	labproducción	7	10	almuerzo
16	14-01-2012	10.35.17	proverbia.net	irrelevante	yperez	92.04Kb	F6	labproducción	8	22	tarde
17	13-01-2012	10.34.16	patriagrande....	interes	yperez	12Kb	F2	labdocente	10	37	comida
18	14-01-2012	10.33.21	proverbia.net	irrelevante	lsaco	45.25Kb	F1	labproducción	10	35	noche
19	15-01-2012	10.35.15	mascotas.com	irrelevante	agonzález	12Kb	F6	labdocente	10	35	mañana
20	13-01-2012	10.33.22	youtube.com	ocio	ecestrada	2.21Mb	F1	labproducción	8	37	almuerzo
21	14-01-2012	10.33.23	ain.cu	nacional	agonzález	1.2Mb	F1	labdocente	8	10	almuerzo
22	15-01-2012	10.35.18	twitter.com	ocio	yperez	1.2Mb	F6	labdocente	12	35	tarde
23	13-01-2012	10.34.12	facebook.com	ocio	lsaco	2.21Mb	F2	labproducción	12	37	madrugada
24	15-01-2012	10.34.14	mascotas.com	irrelevante	ecestrada	45.25Kb	F2	labproducción	8	25	mañana
25	15-01-2012	10.33.20	proverbia.net	irrelevante	yperez	12Kb	F1	labdocente	7	35	tarde
26	14-01-2012	10.34.16	prensalatina.cu	nacional	rpelegrino	45.25Kb	F2	labdocente	8	20	noche
27	13-01-2012	10.33.21	facebook.com	ocio	lsaco	45.25Kb	F1	labproducción	12	35	comida
28	14-01-2012	10.35.16	taringa.net	ocio	ecestrada	12Kb	F6	labproducción	8	20	noche
29	14-01-2012	10.35.15	youtube.com	ocio	rpelegrino	2.21Mb	F6	labdocente	12	40	tarde
30	13-01-2012	10.34.16	facebook.com	ocio	lsaco	45.25Kb	F2	labdocente	8	35	mañana

Figura 11. Muestra de una parte de los datos que se analizan.

Una vez transformados los datos y creada la variable horario, se obtuvieron los siguientes resultados:

Figura 12. Comportamiento de los accesos según la distribución de horarios.

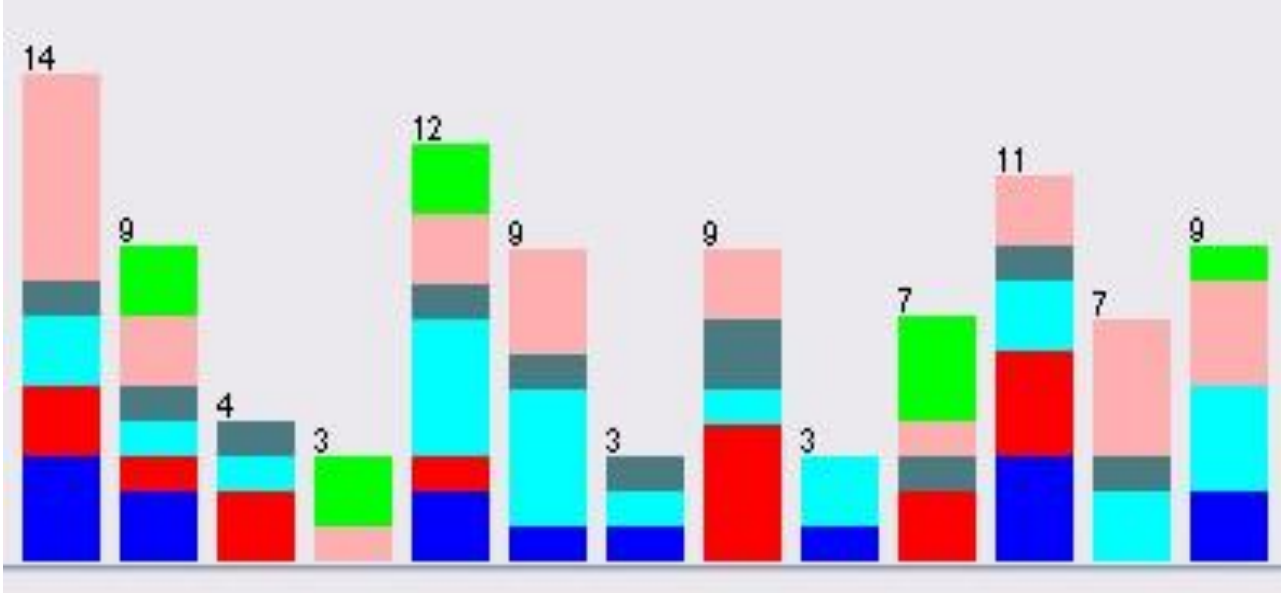


Figura 13. Relación de los dominios accedidos en los diferentes horarios

Como muestra la gráfica se decidió establecer los horarios en cinco categorías como se propuso anteriormente. Las tablas descritas constituyen las vistas minables para la próxima fase de modelación.

3.4. Modelado

El modelado es una tarea importante, en la que se utilizan técnicas que son aplicadas a un grupo de datos seleccionados. Sobre un mismo conjunto se pueden emplear diversas técnicas, pero se debe tener en cuenta que algunas plantean requerimientos específicos sobre la forma de los datos. Por lo tanto frecuentemente es necesario regresar a la fase de preparación de datos. Para realizar la fase de modelado se realizan las pruebas con el algoritmo Apriori.

Experimento aplicando Reglas de Asociación

Como se define en la investigación, el algoritmo Apriori es el que se propone y con el que se hacen las pruebas correspondientes sobre el conjunto de datos definido. Los algoritmos de asociación permiten la búsqueda automática de reglas que relacionan conjuntos de atributos entre sí, esto es lo que se obtiene a continuación.

Para evaluar las reglas se emplean la medida del soporte y la confianza, que indica el número de casos que predice la regla correctamente y viene expresado como el cociente entre el número de casos en que se cumple la regla y el número de casos en que se aplica. A continuación se realizan las pruebas que responden a las funcionalidades propuestas, utilizando como métrica la confianza.

El algoritmo Apriori para su funcionamiento posee una serie de parámetros que se describen a continuación:

Parámetro	**Descripción**
numRules (1000)	Número de reglas requerido. Se escogió un número lo suficientemente grande en relación con la cantidad de registros que se analizan, en este caso 1 000.
metricType (confianza)	Tipo de métrica que se emplea para generar las reglas. Se utiliza la confianza.
minMetric (0.8)	Mínimo valor de la métrica empleada. Se emplea el valor 0.8 para la confianza.
delta (0.05)	Constante por la que va decreciendo el soporte en cada iteración del algoritmo, en este caso 0.05. Este valor es recomendado por la aplicación y es el que se aplica.
upperBoundMinSupport (1.0)	Máximo valor del soporte de los itemsets. Si los itemsets tienen un soporte mayor, no se les toma en consideración. Se usa el valor 1.
lowerBoundMinSupport (0.5)	Mínimo valor del soporte de los itemsets. Debido a que el conjunto de datos a analizar no representa un gran número de registros se utiliza 0.5 como valor de mínimo soporte.
removeAllMissingsCols (false)	Si se activa, no se toman en consideración los atributos con todos los valores perdidos. Se usa su valor por defecto en falso.

Resultados obtenidos para conocer los usuarios que tienden a conectarse a sitios clasificados en diferentes horarios.

En este caso se aplicó el filtro Remove sobre los atributos distintos a: horario, usuarios y categoría.

Horario	Usuarios	Categoría
mañana	yperez	nacional
mañana	agonzález	interés
mañana	rpelegrino	ocio
almuerzo	yperez, lsaco	nacional
almuerzo	rpelegrino, ecestrada	ocio
tarde	agonzález.	interés
tarde	yperez	ocio, nacional e interés
comida	lsaco, agonzález	ocio
comida	rpelegrino, ecestrada, yperez, agonzález	nacional
comida	rpelegrino	irrelevante
comida	rpelegrino, yperez	interés
noche	lsaco, ecestrada	ocio
noche	lsaco, rpelegrino, ecestrada	irrelevante
noche	rpelegrino	nacional, interés
madrugada	lsaco, yperez, rpelegrino, ecestrada	ocio
madrugada	yperez	interés
madrugada	rpelegrino, ecestrada, lsaco	nacional
madrugada	agonzález	irrelevante

A continuación una parte de los resultados que muestra el WEKA con el valor de la confianza obtenida:

```
 1. categoria=ocio 4 ==> horario=madrugada 4    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 2. categoria=nacional 3 ==> horario=madrugada 3    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 3. usuario=lsaco 3 ==> horario=madrugada 3    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 4. usuario=yperez 2 ==> categoria=interes 2    <conf:(1)> lift:(5) lev:(0.16) [1] conv:(1.6)
 5. categoria=interes 2 ==> usuario=yperez 2    <conf:(1)> lift:(5) lev:(0.16) [1] conv:(1.6)
 6. categoria=interes 2 ==> horario=madrugada 2    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 7. usuario=rpelegrino 2 ==> horario=madrugada 2    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 8. usuario=ecestrada 2 ==> horario=madrugada 2    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 9. usuario=yperez 2 ==> horario=madrugada 2    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
10. categoria=ocio usuario=lsaco 2 ==> horario=madrugada 2    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
11. usuario=yperez horario=madrugada 2 ==> categoria=interes 2    <conf:(1)> lift:(5) lev:(0.16) [1] conv:(1.6)
12. categoria=interes horario=madrugada 2 ==> usuario=yperez 2    <conf:(1)> lift:(5) lev:(0.16) [1] conv:(1.6)
13. categoria=interes usuario=yperez 2 ==> horario=madrugada 2    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
14. usuario=yperez 2 ==> categoria=interes horario=madrugada 2    <conf:(1)> lift:(5) lev:(0.16) [1] conv:(1.6)
15. categoria=interes 2 ==> usuario=yperez horario=madrugada 2    <conf:(1)> lift:(5) lev:(0.16) [1] conv:(1.6)
16. usuario=agonzález 1 ==> categoria=irrelevante 1    <conf:(1)> lift:(10) lev:(0.09) [0] conv:(0.9)
17. categoria=irrelevante 1 ==> usuario=agonzález 1    <conf:(1)> lift:(10) lev:(0.09) [0] conv:(0.9)
18. categoria=irrelevante 1 ==> horario=madrugada 1    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
19. usuario=agonzález 1 ==> horario=madrugada 1    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
20. categoria=ocio usuario=rpelegrino 1 ==> horario=madrugada 1    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
21. categoria=ocio usuario=ecestrada 1 ==> horario=madrugada 1    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
22. categoria=nacional usuario=rpelegrino 1 ==> horario=madrugada 1    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
23. categoria=nacional usuario=ecestrada 1 ==> horario=madrugada 1    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
24. categoria=nacional usuario=lsaco 1 ==> horario=madrugada 1    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
25. usuario=agonzález horario=madrugada 1 ==> categoria=irrelevante 1    <conf:(1)> lift:(10) lev:(0.09) [0] conv:(0.9)
26. categoria=irrelevante horario=madrugada 1 ==> usuario=agonzález 1    <conf:(1)> lift:(10) lev:(0.09) [0] conv:(0.9)
27. categoria=irrelevante usuario=agonzález 1 ==> horario=madrugada 1    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
28. usuario=agonzález 1 ==> categoria=irrelevante horario=madrugada 1    <conf:(1)> lift:(10) lev:(0.09) [0] conv:(0.9)
29. categoria=irrelevante 1 ==> usuario=agonzález horario=madrugada 1    <conf:(1)> lift:(10) lev:(0.09) [0] conv:(0.9)
```

Figura 14. Visualización de una muestra de las reglas generadas en WEKA.

Resultados obtenidos para conocer la categoría de los sitios a los que tiende a conectarse un usuario especificado (lsaco), desde las diferentes áreas de la universidad en un determinado horario.

Sobre el conjunto de datos se aplicó el filtro Remove a los atributos distintos a: horario, usuarios, área y categoría.

Usuario	Horario	Categoría	Área
lsaco	noche	ocio	labproducción
lsaco	noche	irrelevante	labdocente

```
categoria=ocio area=labdocente horario=noche 2 ==> usuario=lsaco 2    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
categoria=irrelevante area=labproducción horario=noche 2 ==> usuario=lsaco 2    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
```

Figura 15. Visualización en WEKA.

A partir de este análisis se da respuesta a las siguientes funcionalidades:

1- De un usuario conocer la categoría de los sitios a los que tiende a conectarse.
2- De un usuario conocer la categoría de los sitios a los que tiende a conectarse, desde las diferentes áreas de la universidad.

Resultados obtenidos para conocer del dominio especificado (facebook.com), el horario en que se realizan más accesos desde las diferentes áreas de la universidad.

Sobre el conjunto de datos se aplicó el filtro Remove a los atributos distintos a: horario, área y dominio.

Dominio	Horario	Área
facebook.com	mañana	labdocente
facebook.com	madrugada	labproduccion

```
horario=mañana 2 ==> dominio=facebook.com area=labdocente 2    <conf:(1)> lift:(3) lev:(0.15) [1] conv:(1.33)
area=labproducción horario=madrugada 2 ==> dominio=facebook.com 2    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
dominio=facebook.com horario=madrugada 2 ==> area=labproducción 2    <conf:(1)> lift:(1.5) lev:(0.07) [0] conv:(0.67)
horario=madrugada 2 ==> dominio=facebook.com area=labproducción 2    <conf:(1)> lift:(1.5) lev:(0.07) [0] conv:(0.67)
```

Figura 16. Visualización en WEKA.

A partir de este análisis se da respuesta a las siguientes funcionalidades:

1- De un usuario conocer los dominios a los que tiende a conectarse.

2- De un usuario conocer los dominios a los que tiende a conectarse en un determinado horario.

3- De un usuario conocer los dominios a los que tiende a conectarse, desde las diferentes áreas de la universidad.

Resultados obtenidos para conocer de la facultad 1, los tipos de sitios a los que más se acceden.

Sobre el conjunto de datos se aplicó el filtro Remove a los atributos distintos a: facultad y categoría.

Facultad	Categoría
F1	ocio
F1	nacional

```
categoria=ocio 12 ==> facultad=F1 12    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
categoria=nacional 11 ==> facultad=F1 11    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
```

Figura 17. Visualización en WEKA.

Resultados obtenidos para conocer los dominios que más demoran en procesar la solicitud.

El campo que registra el tiempo en que se demora en procesar una solicitud fue necesario discretizarlo para obtener las clasificaciones, bajo, medio y alto. Luego sobre el conjunto de datos se aplicó el filtro Remove a los atributos distintos a: dominio y tiempo de respuesta. En este caso se configuró nuevamente los parámetros del algoritmo Apriori y se modificó el valor de la confianza a 0.5 para obtener los resultados.

Dominio	Tiempo de respuesta

facebook.com	alto
youtube.com	alto
twitter.com	alto

```
dominio=cubadebate.cu 3 ==> tiempo_respuesta=bajo 3    <conf:(1)> lift:(4) lev:(0.02) [2] conv:(2.25)
dominio=prensalatina.cu 11 ==> tiempo_respuesta=medio 9    <conf:(0.82)> lift:(1.52) lev:(0.03) [3] conv:(1.69)
dominio=ain.cu 7 ==> tiempo_respuesta=medio 5    <conf:(0.71)> lift:(1.32) lev:(0.01) [1] conv:(1.07)
dominio=proverbia.net 7 ==> tiempo_respuesta=medio 5    <conf:(0.71)> lift:(1.32) lev:(0.01) [1] conv:(1.07)
dominio=facebook.com 9 ==> tiempo_respuesta=alto 6    <conf:(0.67)> lift:(3.17) lev:(0.04) [4] conv:(1.78)
dominio=patriagrande.com.ve 9 ==> tiempo_respuesta=medio 6    <conf:(0.67)> lift:(1.23) lev:(0.01) [1] conv:(1.03)
dominio=ecured.cu 9 ==> tiempo_respuesta=bajo 6    <conf:(0.67)> lift:(2.67) lev:(0.04) [3] conv:(1.69)
dominio=mascotas.com 9 ==> tiempo_respuesta=medio 6    <conf:(0.67)> lift:(1.23) lev:(0.01) [1] conv:(1.03)
dominio=taringa.net 3 ==> tiempo_respuesta=medio 2    <conf:(0.67)> lift:(1.23) lev:(0) [0] conv:(0.69)
dominio=juventudrebelde.cu 3 ==> tiempo_respuesta=medio 2    <conf:(0.67)> lift:(1.23) lev:(0) [0] conv:(0.69)
dominio=wikipedia.com 12 ==> tiempo_respuesta=medio 7    <conf:(0.58)> lift:(1.08) lev:(0.01) [0] conv:(0.92)
dominio=youtube.com 14 ==> tiempo_respuesta=alto 7    <conf:(0.5)> lift:(2.38) lev:(0.04) [4] conv:(1.38)
dominio=twitter.com 4 ==> tiempo_respuesta=alto 2    <conf:(0.5)> lift:(2.38) lev:(0.01) [1] conv:(1.05)
```

Figura 18. Visualización en WEKA.

3.5. Evaluación y acciones de despliegue

Para la evaluación del modelo escogido se tiene en cuenta el cumplimiento de los objetivos propuestos. En este paso se revisa el proceso para determinar si debe repetirse alguna fase en caso de existir algún error. Si el modelo generado es válido se procede al despliegue del proyecto.

Se determina que las Reglas de Asociación con el uso del algoritmo Apriori es la técnica con la que se obtiene los mejores resultados a partir de cada funcionalidad propuesta.

Reglas obtenidas a partir de los datos seleccionados para la minería:

- ✓ En el horario de la mañana el usuario yperez tiende a conectarse a sitios nacionales.
- ✓ En los horarios de la mañana y la tarde el usuario agonzález tiende a conectarse a sitios de interés.
- ✓ En el horario de la mañana el usuario rpelegrino tiende a conectarse a sitios de ocio.

- ✓ En el horario de almuerzo los usuarios yperez y lsaco tienden a conectarse a sitios nacionales.
- ✓ En el horario de almuerzo los usuarios rpelegrino y ecestrada tienden a conectarse a sitios de ocio.
- ✓ En el horario de la tarde el usuario yperez tienden a conectarse a sitios de ocio, nacionales e interés.
- ✓ En el horario de comida los usuarios lsaco y agonzález tienden a conectarse a sitios de ocio.
- ✓ En el horario de comida el usuario rpelegrino tiende a conectarse a sitios irrelevantes.
- ✓ En el horario de comida los usuarios rpelegrino y yperez tienden a conectarse a sitios de interés.
- ✓ En el horario de la noche los usuarios lsaco y ecestrada tienden a conectarse a sitios de ocio.
- ✓ En el horario de la noche los usuarios lsaco, rpelegrino y ecestrada tienden a conectarse a sitios irrelevantes.
- ✓ En el horario de la noche el usuario rpelegrino tiende a conectarse a sitios nacionales y de interés.
- ✓ En el horario de la madrugada los usuarios lsaco, yperez, rpelegrino y ecestrada tienden a conectarse a sitios de ocio.
- ✓ En el horario de la madrugada el usuario yperez tiende a conectarse a sitios de interés.
- ✓ En el horario de la madrugada los usuarios rpelegrino, ecestrada y lsaco tienden a conectarse a sitios nacionales.
- ✓ En el horario de la madrugada el usuario agonzález tiende a conectarse a sitios irrelevantes.
- ✓ En el horario de la madrugada los usuarios lsaco, yperez, rpelegrino y ecestrada tienden a conectarse a sitios de ocio.
- ✓ En el horario de la madrugada el usuario yperez tiende a conectarse a sitios de interés.
- ✓ En el horario de la madrugada los usuarios rpelegrino, ecestrada y lsaco tienden a conectarse a sitios nacionales.

- ✓ El usuario lsaco cuando se conecta en el horario de la noche a sitios de ocio lo hace desde los laboratorios de producción.
- ✓ El usuario lsaco cuando se conecta en el horario de la noche a sitios irrelevantes lo hace desde los laboratorios docentes.
- ✓ El dominio facebook.com es más visitado en el horario de la mañana desde los laboratorios docentes.
- ✓ El dominio facebook.com es más visitado en el horario de la madrugada desde los laboratorios de producción.
- ✓ Los usuarios de la facultad 1 tienden a conectarse más a los sitios nacionales y de ocio.

Las reglas obtenidas responden a los objetivos deseados, por lo que se concluye que fueron cumplidos los propósitos trazados correspondientes al descubrimiento de patrones ocultos en los datos; que permiten describir las relaciones existentes en los atributos de los datos analizados.

En el proyecto no se propone repetir ningún paso, ya que después del análisis no se han encontrado fallas, ni se ha omitido ninguna variable que pudiera limitar el éxito de los resultados. Las propuestas de mejoras en los modelos se dejan para próximas iteraciones o proyectos que se deseen realizar.

Finalmente se determina que el proyecto puede ser desplegado, se toman los resultados de la evaluación y se concluye una estrategia para el despliegue. Las acciones propuestas son: resumen de los resultados obtenidos, decidir para cada resultado el conocimiento o la información proporcionada a sus usuarios y cómo estos podrán ser utilizados dentro de los sistemas de la organización.

3.6. Conclusiones

Con el desarrollo de este capítulo se ha llegado a las siguientes conclusiones parciales:

- ➢ La metodología CRISP-DM constituye una guía valiosa para el desarrollo de proyectos de MD.

- Con la herramienta Weka se llevan a cabo proyectos de MD con calidad, ya que implementa diferentes técnicas y algoritmos de minería que permiten un exhaustivo análisis de los datos.
- Las reglas de asociación resultan útiles para obtener relaciones dentro de los datos que describan el comportamiento de los usuarios en el uso de las cuotas de navegación.
- Al realizar los diferentes experimentos sobre los datos, el Apriori resulta eficiente para la obtención del conocimiento implícito a partir de los registros de AiresProxy y se obtienen los resultados adecuados en correspondencia con las funcionalidades propuestas.

Conclusiones Generales

Para la realización del presente trabajo se plantearon un conjunto de tareas que posibilitaron el desarrollo y cumplimiento de los objetivos propuestos, con esta investigación se arribó a los siguientes resultados:

- Con el estudio realizado de las diferentes técnicas de MD, las reglas de asociación han demostrado ser un mecanismo viable para el descubrimiento de conocimientos a partir de los registros de AiresProxy.

- Los algoritmos para obtener los elementos frecuentes en una base de datos, son una parte importante en el minado de las reglas, ofreciéndoles los elementos más importantes dentro del conjunto de datos. En este caso el Apriori es el algoritmo de reglas de asociación propuesto para su utilización en el sistema AiresProxy.

- Se proponen las funcionalidades que pueden ser aplicables en AiresProxy con el algoritmo Apriori, para obtener una mejor descripción del uso que le dan los usuarios a las cuotas de navegación. Estos nuevos reportes son de gran utilidad para la DRSI a la hora de tomar decisiones en base a un uso adecuado de la cuota de navegación.

- Para la aplicación de las funcionalidades propuestas es necesario realizar una reestructuración en el diseño de la BD, por lo que se detallan los cambios y se esboza la nueva estructura.

- Con el apoyo de la herramienta Weka para el análisis y experimentación de los datos del caso de estudio planteado, se comprobó la efectividad del algoritmo Apriori. Se obtuvieron reglas con un alto porcentaje sobre las tendencias y patrones en los datos que describen el comportamiento de los usuarios en el uso de las cuotas de navegación por Internet través de un servidor proxy.

Teniendo en cuenta todo lo expuesto en esta investigación, se concluye que los objetivos trazados se cumplieron de manera satisfactoria, con la documentación necesaria y la certeza de que el mismo puede utilizarse para futuras investigaciones.

Recomendaciones

- Aplicación de la propuesta en el sistema AiresProxy para enriquecer la descripción de la navegación por Internet y apoyar el sistema de toma de decisiones de la DRSI.

Referencias bibliográficas

[1]- Stringer, Gary & The University of Exeter. The Internet. Universidad de Exeter, Reino Unido. Creative Media y Tecnología de la Información (CMIT). 1999-2005.

[2]- López, Cesar Pérez y González, Daniel Santin. Minería de Datos, Técnicas y herramientas.1ra Edición, 2da Reimpresión. 2008. pág. 4.

[3]- Orallo, José Hernández, José Ramírez, y Cesar Ferri. Introducción a la Minería de Datos. Madrid. 2004.

[4]- López, JM Molina y Herrero, J García. Técnicas de Análisis de Datos. Aplicaciones Prácticas utilizando Microsoft Excel y WEKA. Universidad Carlos III: Madrid. 2006.

[5]- Vallejos, S. J. Minería de Datos. Universidad Nacional del Nordeste, Argentina. Licenciatura en Sistemas de Información. 2006.

[6]- Villanueva, Wladimiro Díaz. Almacenes de Datos (DataWarehouses). Universidad de Valencia. (Citado el 25 de abril del 2012).

[7]- Tuya, Javier, Román, Isabel Ramos y Cosín, Javier Colado. Técnicas cuantitativas para la gestión en la ingeniería de software. 2007.

[8]- SAS Institute Inc. SAS. [En línea] [Citado el: 26 de Diciembre de 2011.] Disponible en: http://www.sas.com

[9]- Fernández, Lic. Enrique. Asistente para la Gestión de Documentos de Proyectos de Explotación de Datos. Tesis de Magister en Ingeniería de Software. [En línea] 2006. [Citado el: 15 de Enero de 2012.] Disponible en: http://laboratorios.fi.uba.ar/lsi/rgm/tesistas/fernandez-tesisdemagister.pdf

[10]-López, Cesar Pérez y González, Daniel Santin. Minería de Datos, Técnicas y herramientas.1ra Edición, 2da Reimpresión. 2008. pág. 498.

[11]- Roche, Ariel. Árboles de decisión y Series de Tiempo. Facultad de Ingeniería, UDELAR, Uruguay. 2009.

[12]- Takeyas, Ing. Bruno López. Ingeniería en Sistemas Computacionales. Inteligencia Artificial. Algoritmo C4.5. 2005.

[13]- Arazuri, Eva Sanz y Elizondo, Ana Ponce de León. Claves en la aplicación del algoritmo CHAID. Un estudio del ocio físico deportivo universitario. 2010. Disponible en: http://redalyc.uaemex.mx/redalyc/pdf/2351/235116352011.pdf

[14]- Nettleton, David F. Técnicas para el análisis de datos clínicos. 2005. pág. 13.

[15]- Garre, Miguel, Cuadrado, Juan José y Sicilia, Miguel Ángel. Comparación de diferentes algoritmos de clustering en la estimación de coste en el desarrollo de software. [En línea] Disponible en: http://www.sc.ehu.es/jiwdocoj/remis/docs/GarreAdis05.pdf

[16]-Ordoñes Leyva, Yoanni y Avilés Vázquez, Ernesto. Herramienta informática de Minería de Uso de la Web sobre los registros de navegación por Internet. Universidad de las Ciencias Informáticas. Ciudad de La Habana. 2010.

[17]- Agrawal, R, Imielinski, T and Swami, A. Database mining: A performance perspective. IEEE Transactions on Knowledge and Data Engineering. 1993.

[18]- González, A, Yoan Rodríguez, Trinidad, J. Francisco Martínez, Ochoa, J. Ariel Carrasco y Shulcloper, José Ruiz. Minería de Reglas de Asociación sobre Datos Mezclados. 2009. Pág. 5.

[19]- Sarasa, R.B. Minería de Datos aplicada a la Gestión Docente del Instituto Superior Politécnico José Antonio Echeverría, en Facultad de Ingeniería Informática. Instituto Superior Politécnico José Antonio Echeverría. Ciudad Habana. 2008. pág. 92.

[20]- Data mining with SAS Enterprise Miner. [En línea] [Citado el: 6 de Enero de 2012.] Disponible en: http://www.sas.com/technologies/analytics/datamining/miner/

[21]- González, Erika Vilches y Broitman, Iván A. Escobar. Minería de Datos. [En línea] Septiembre de 2007. [Citado el: 8 de Noviembre de 2011.] Disponible en: http://www.erikavilches.com/km/mineria_datos.pdf

Bibliografía

1. Mark Hall Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Ian H. Witten y Peter Reutemann. The WEKA Data Mining Software: An Update. [En línea] 2009. [Citado el: 24 de marzo de 2012]. Disponible en: http://www.cs.waikato.ac.nz/~eibe/pubs/weka_update.pdf

2. Arias, Rigoberto Mora, Pino, Omar Vidal y Pérez, Lisandra Guerrero. Aplicación de técnicas de minería de datos con Weka Acknowledge Explorer. 2011.

3. Roberth Paúl Bravo Castro y María Esther Ruilova Rojas Árboles de clasificación (inteligencia artificial avanzada). Universidad Técnica Particular de Loja, Ecuador.

4. Pierrakos, Dimitrios, y otros. Web Usage Mining as a Tool for Personalization: A Survey. Kluwer Academic Publishers. 2003.

5. Vercellis, C. Business Intelligence: Data Mining and Optimization for Decision Making. Chichester, West Sussex, U.K: John Wiley & Sons Ltd. 2009.

6. Marín R. y Palma J. Inteligencia artificial, técnicas, métodos y aplicaciones. Madrid, España, 2008.

7. Licencia de software libre orientada a proteger la libre distribución, modificación y uso del software. Ingenierías, Enero-Marzo 2005, Vol. VIII, No. 26

8. Ministerio de la Informática y las Comunicaciones (MIC). [En línea] 2002-2012. [Citado el: 24 de abril de 2012]. Disponible en: http://www.mic.gov.cu/sitiomic/servlet/hinfo?1,67

9. EcuRed. [En línea]. [Citado el: 12 de mayo de 2012]. Disponible en: http://www.ecured.cu/index.php/Bases_de_datos

10. Sitio Oficial MongoDB. [En línea]. [Citado el: 4 de febrero de 2012]. Disponible en: http://www.mongodb.org/

11. Binary JSON. [En línea]. [Citado el: 8 de mayo de 2012]. Disponible en: http://bsonspec.org/

12. NoSQL. [En línea]. [Citado el: 23 de enero de 2012]. Disponible en: http://nosql-database.org/

13. Introducing JSON. [En línea]. [Citado el: 8 de mayo de 2012]. Disponible en: http://www.json.org/

14. Datamining (Minería de datos). [En línea]. [Citado el: 19 de febrero de 2012]. Disponible en: http://www.sinnexus.com/business_intelligence/datamining.aspx

15. Squid: Optimising Web Delivery. [En línea]. [Citado el: 27 de noviembre 2011]. Disponible en: http://www.squid-cache.org/

Glosario de Términos

UCI: Universidad de las Ciencias Informáticas.

Servidor proxy: es un equipo intermediario situado entre el sistema del usuario e Internet. Puede utilizarse para registrar el uso de Internet y también para bloquear el acceso.

Log: es un registro oficial de eventos o sucesos durante un periodo de tiempo en particular, se obtiene a partir del servidor proxy en este caso.

AiresProxy: proyecto desarrollado en la universidad que brinda reportes estadísticos a partir de los registros del servidor proxy.

URI: *Uniform Resource Identifier*. Es un identificador uniforme de recurso (una dirección de Internet que permite localizar un servicio, sea web, e-mail, ftp).

URL: *Uniform Resource Locator* (Localizador de Recurso Uniforme), dirección única que identifica a una página web en Internet.

IP: Dirección IP. Etiqueta numérica que identifica, de manera lógica y jerárquica, a una interfaz (elemento de comunicación/conexión) de un dispositivo (habitualmente una computadora) dentro de una red que utilice el protocolo IP (Internet Protocol).

Dominio: un dominio de Internet es una red de identificación asociada a un grupo de dispositivos o equipos conectados a la red Internet.

BD: Base de Datos es una entidad en la cual se pueden almacenar datos de manera estructurada, con la menor redundancia posible.

NoSQL: Not Only SQL. Movimiento de bases de datos: no-relacional, distribuida, de código abierto y escalable horizontalmente.

MongoDB: es una base de datos escalable y de alto rendimiento, de código abierto y del grupo NoSQL.

OLAP: es una solución utilizada en el campo de la llamada Inteligencia Empresarial, cuyo objetivo es agilizar la consulta de grandes cantidades de datos.

Datawarehouse: base de datos que almacena una gran cantidad de datos transaccionales integrados para ser usados en análisis de gestión por usuarios especializados.

MD: Minería de Datos, conjunto de técnicas y tecnologías que permiten explorar grandes bases de datos, de manera automática o semiautomática, con el objetivo de encontrar patrones repetitivos, tendencias o reglas que expliquen el comportamiento de los datos en un determinado contexto.

Weka: Waikato Environment for Knowledge Analysis. Herramienta para el análisis de datos.

JSON: Acrónimo de JavaScript Object Notation, no es más un formato ligero utilizado para el intercambio de datos, es un subconjunto de la notación literal de objetos de Java Script pero que no requiere el uso de XML.

BSON: la abreviatura de Binary JSON , es una serialización binaria codificada en JSON como los documentos.

Squid: es utilizado por cientos de proveedores de Internet en todo el mundo para implementar en un servidor proxy. Squid optimiza el flujo de datos entre cliente y servidor para mejorar el rendimiento y almacena en caché el contenido de uso frecuente para ahorrar ancho de banda.

HTTP: HyperText Transfer Protocol (Protocolo de transferencia de hipertexto) es el método más común de intercambio de información en la world wide web, el método mediante el cual se transfieren las páginas web a un ordenador.

PC: Personal Computer (Computadora Personal) es una microcomputadora diseñada en principio para ser usada por una sola persona a la vez.

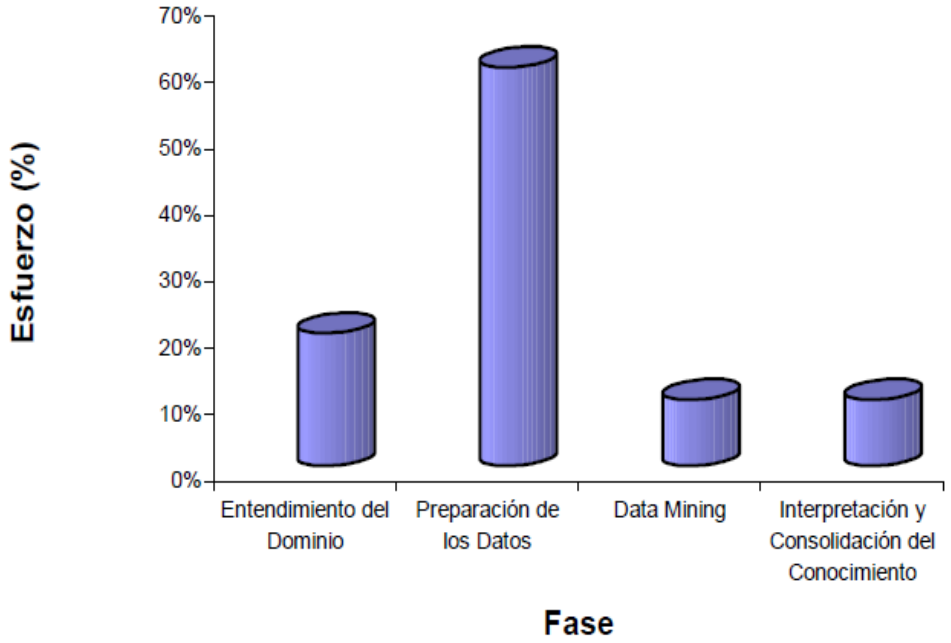
Licencia GPL: Licencia de software libre orientada a proteger la libre distribución, modificación y uso del software.

Megas: se usa el término mega para designar a un megabyte, que equivale a 1024 bytes.

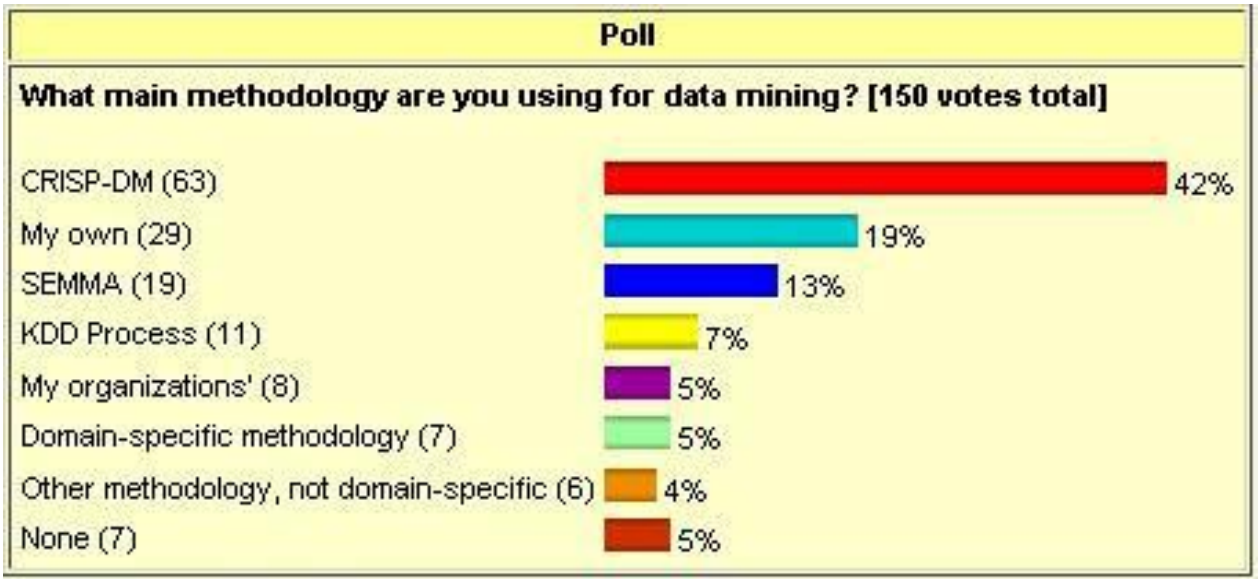
HTML: HyperText Markup Language (lenguaje de marcado de hipertexto), lenguaje compuesto de una serie de etiquetas o marcas que permiten definir el contenido y la apariencia de las páginas Web.

Anexos

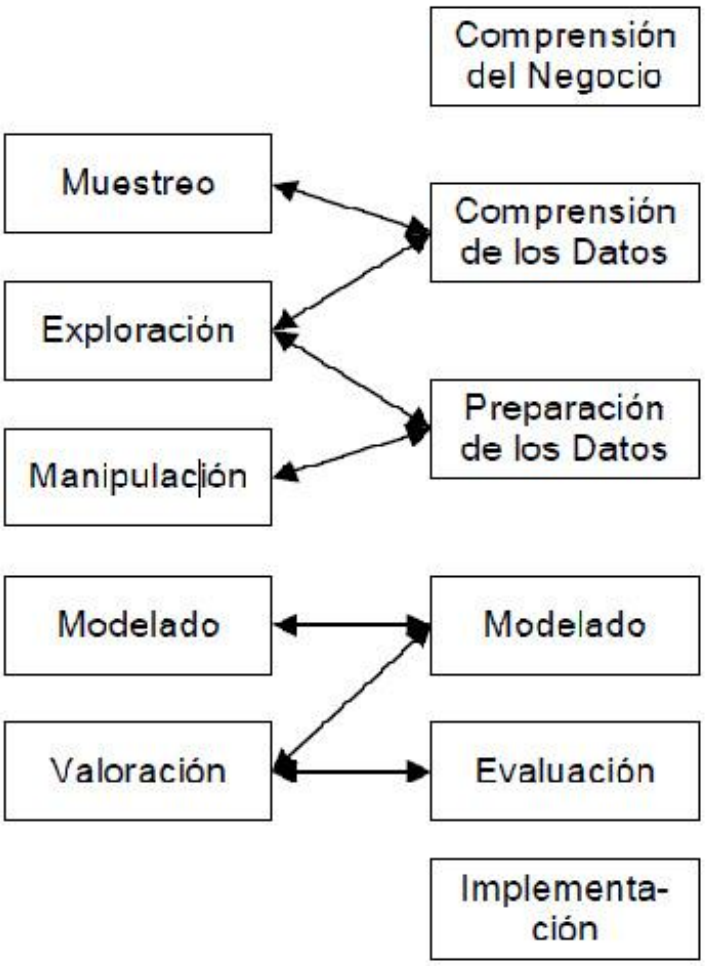
Anexo 1. Esfuerzo requerido por fases en un proceso de KDD [4].



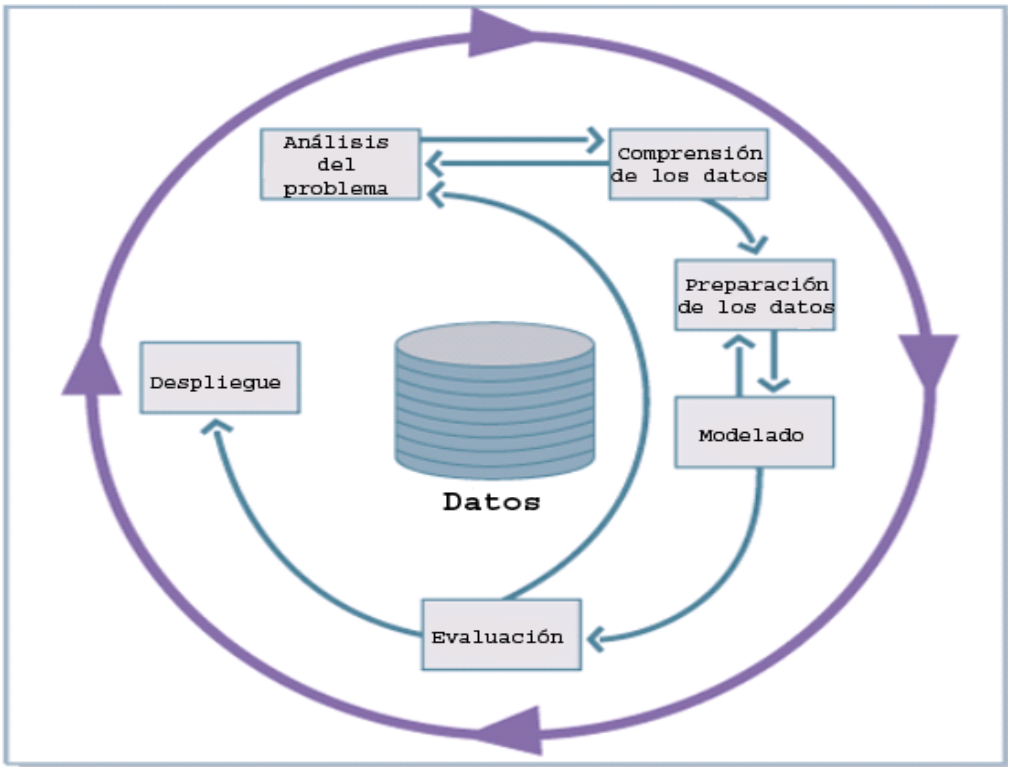
Anexo 2. Metodologías más empleadas durante un proceso de KDD [8]



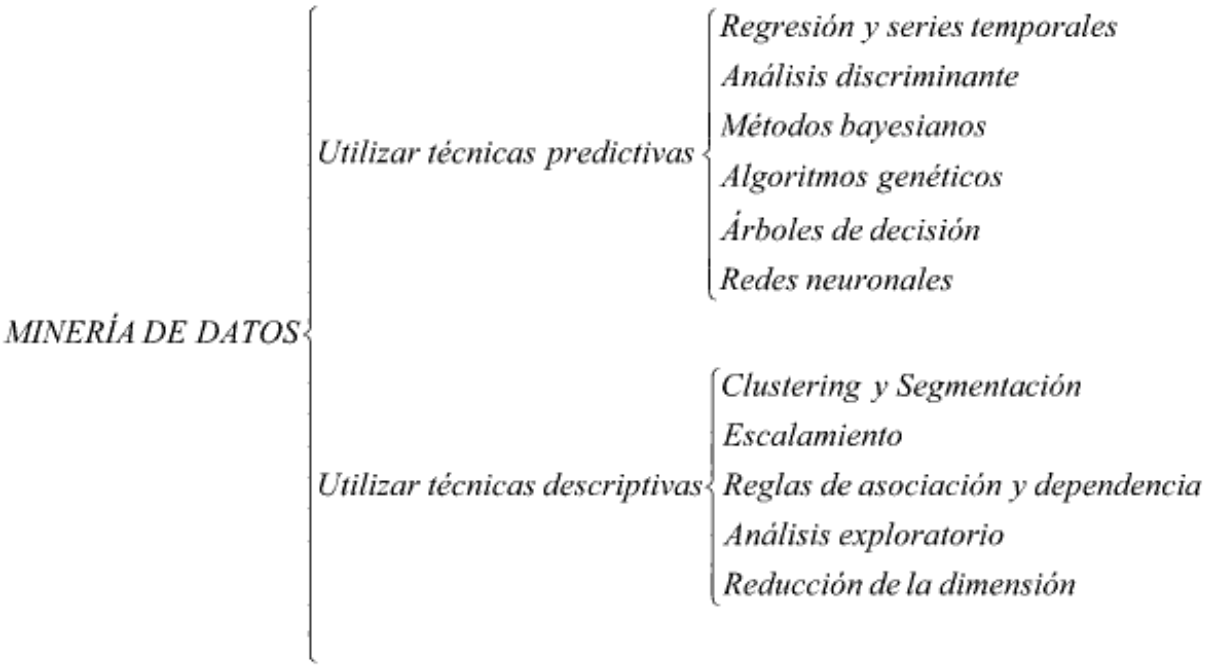
Anexo3. Comparativa de las interrelaciones entre las fases de las metodologías SEMMA y CRISP-DM [9].



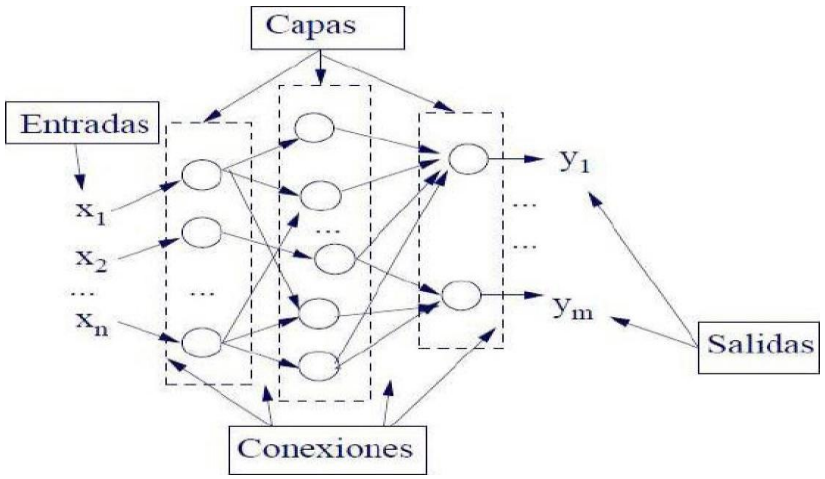
Anexo 4. Fases de la Minería de Datos [5].



Anexo 5. Técnicas de Minería de Datos [10].

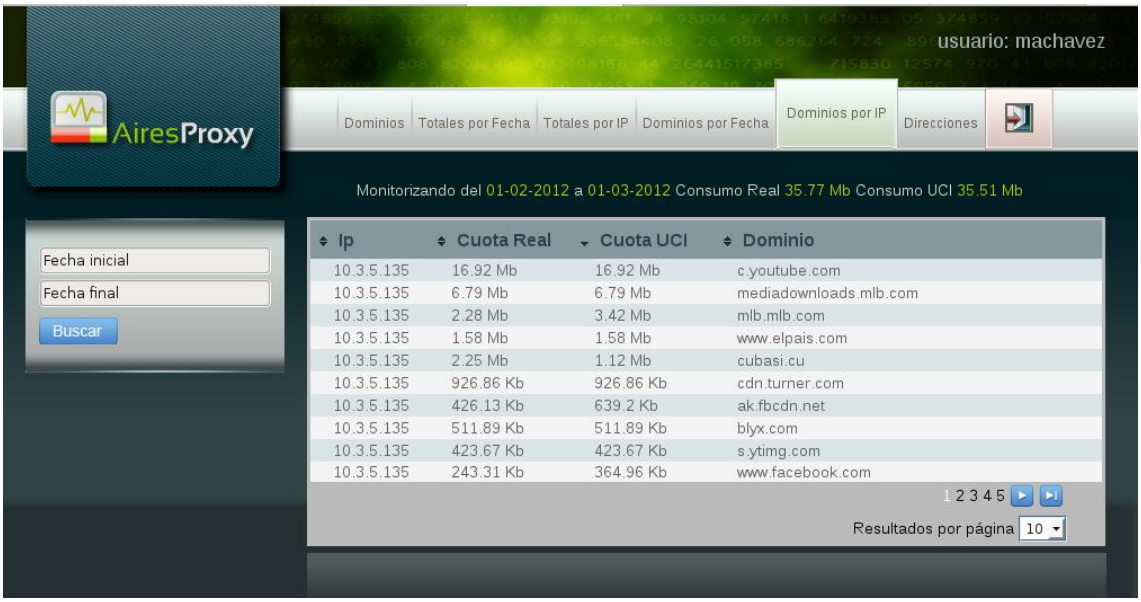


Anexo 6. Estructura de las Redes Neuronales [4].



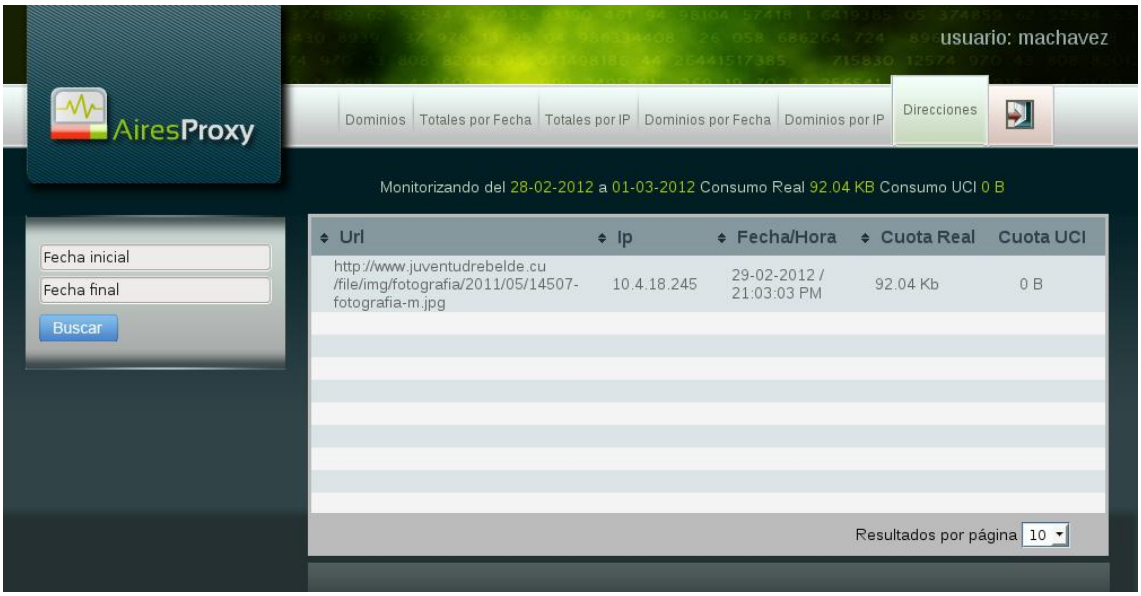
Anexo 7. AiresProxy: Dominios

Anexo 8. AiresProxy: Dominios por IP



Anexo 9. AiresProxy: Dominios por fecha

Anexo 10. AiresProxy: Direcciones



Anexo11. AiresProxy: Totales por fecha

Anexo 12. AiresProxy: Totales por IP