UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
Facultad 6

ALGORITMOS DE MINERÍA DE DATOS: ÁRBOLES DE DECISIÓN Y REGLAS DE INDUCCIÓN INTEGRADOS A POSTGRESQL

Tesis presentada en opción al título de
Máster en Informática Aplicada

Autor: Ing. Yadira Robles Aranda.
Tutores: Dr. Rafael Arturo Trujillo Rasúa.
MSc. Anthony Rafael Sotolongo.

La Habana, Cuba
Diciembre, 2012
"Año 54 de la Revolución"

"El conocimiento es poder. La información es libertadora. La educación es la premisa del progreso..."

Kofi Annan

DECLARACIÓN JURADA DE AUTORÍA

Yo Yadira Robles Aranda, con carné de identidad 84111218099, declaro que soy la autora principal del resultado que expongo en el presente trabajo titulado “Algoritmos de minería de datos: Árboles de decisión y Reglas de inducción integrados a PostgreSQL”, para optar por el título de Máster en Informática Aplicada.

El trabajo fue desarrollado durante el período comprendido entre los años 2011-2012 en colaboración con el Dr. Rafael Arturo Trujillo Rasúa y el MSc. Anthony Rafael Sotolongo, quienes me reconocen la autoría principal del resultado expuesto en esta investigación.

Finalmente declaro que todo lo anteriormente expuesto se ajusta a la verdad, y asumo la responsabilidad moral y jurídica que se derive de este juramento profesional.

Y para que así conste, firmo la presente declaración jurada de autoría en La Habana a los ___ días del mes de _____ del año _____.

_____

Ing. Yadira Robles Aranda.

AGRADECIMIENTOS

A mis padres por siempre estar presentes y brindarme su apoyo incondicional, así como a mis compañeros de trabajo y a todas aquellas personas que de una forma u otra contribuyeron al desarrollo de esta investigación.

DEDICATORIA

A mi familia por darme su apoyo incondicional y a Rolando Rivas Blanco por estar allí conmigo en todo momento, dándome consejo, ayuda y ser exigente conmigo.

RESUMEN

El desarrollo de las tecnologías y la informatización de las empresas ha incrementado el cúmulo de información almacenada en las bases de datos dificultando el proceso de análisis de éstas manualmente, por lo cual es necesario utilizar técnicas como la minería de datos que faciliten obtener conocimientos a partir de los datos recopilados.

En la presente investigación se analizó de las técnicas de minería de datos la de árboles de decisión y la de reglas de inducción para integrar varios de sus algoritmos al sistema gestor de base de datos (SGBD) PostgreSQL, debido a las deficiencias de las herramientas libres existentes para realizar el análisis de la información. En ella se desarrollaron mecanismos para optimizar el rendimiento de los algoritmos implementados con el objetivo de aprovechar las ventajas de PostgreSQL.

Además se comprobó la solución de la investigación mediante la aplicación de las técnicas a la base de datos del Sistema de Genética Médica y se realizó un experimento para comprobar que los algoritmos integrados al gestor permiten aprovechar las potencialidades de PostgreSQL para disminuir el tiempo del análisis de los datos.

Palabras claves: Árboles de decisión, minería de datos, PostgreSQL, reglas de inducción, sistema gestor de bases de datos.

ABSTRACT

The development of technologies and the computerization of business has increased the amount of information stored on the databases hindering the process of analyzing it by hand, so it is necessary to use techniques such as data mining to facilitate knowledge obtained from of the data obtained.

In the present study we analyze the data mining techniques of decision trees and rule induction algorithms integrate several of its base management system (DBMS) PostgreSQL, due to the shortcomings of the free tools available for conducting the analysis of the information. It mechanisms were developed to optimize the performance of the algorithms implemented in order to take advantage of PostgreSQL. Also the solution was found by applying research techniques to the database system Medical Genetics and an experiment was conducted to verify that the integrated algorithms to exploit the potential manager allow PostgreSQL to reduce analysis time data.

Keywords: Data mining, decision trees, management system databases PostgreSQL, rule induction.

ÍNDICE

INTRODUCCIÓN

La informatización de las empresas, organizaciones e instituciones ha generado un gran incremento de la información almacenada en las bases de datos, la cual es de gran utilidad cuando se quiere explicar el pasado, entender el presente y predecir la información futura, por lo que se hace necesario analizar la misma para la obtención de información útil para la organización.

Anteriormente estos análisis se realizaban manualmente mediante técnicas estadísticas, sin embargo esta forma de actuar es lenta, cara, altamente subjetiva y con el incremento del volumen de datos es prácticamente imposible el análisis sin la ayuda de herramientas o técnicas potentes.

Una de las técnicas más usadas para obtener el conocimiento analizando los datos presentes en las bases de datos, es la minería de datos, que nos permite obtener patrones o modelos a partir de los datos recopilados. Esta técnica se aplica en todo tipo de entornos, como por ejemplo en aplicaciones educacionales y financieras, en procesos industriales, policiales, políticos y en la medicina.

Dentro de la minería de datos existen diversas técnicas entre las cuales se encuentran la de inducción de reglas y árboles de decisión, las cuales según diversos estudios realizados se encuentran entre las más utilizadas. (KDnuggets Polls. Data mining methods, 2007) (Heughes Escobar, 2007) (ver anexo 3).

Para la aplicación de estas técnicas existen numerosas herramientas independientes del sistema gestor de bases de datos que permiten aplicarlas a grandes volúmenes de datos, sin embargo, la mayoría de éstas son propietarias como SPSS Clementine, SAS Enterprise Miner y MATLAB, las cuales no están al alcance de las organizaciones cubanas ya que son altamente costosas (ver anexo1) y otras como WEKA o YALE Rapid Miner que a pesar de que tienen licencia GPL es necesario conectarse al sistema gestor de bases de datos y cuando existe una gran cantidad de datos a analizar el proceso se vuelve engorroso y lento. (Soto Jaramillo, 2009). Además se debe garantizar la seguridad de los datos, pues la información viaja a través de la red. Otra alternativa que propone Weka es convertir los datos a la extensión

“arff”, pero ante cualquier actualización de los datos sería necesario volverlos a convertir. Para darle solución a estos problemas en la actualidad algunas empresas como Microsoft y Oracle han desarrollado módulos dentro de sus sistemas gestores de bases de datos que tienen incluido las técnicas de minería de datos, lo que les permite ganar en rapidez en los tiempos de respuesta ya que no sería necesario transformar “datos sin formato” en “información procesable” (preparación de los datos) o importación o vinculación con la herramienta encargada de hacer el análisis, se evita tener que contar con personal preparado en otras herramientas de análisis de datos, se proporciona a los analistas de datos un acceso directo pero controlado lo que acelera la productividad sin poner en riesgo la seguridad de los datos, sin embargo a pesar de estas ventajas éstos tienen el inconveniente de ser propietarios.

Actualmente Cuba está inmersa en migrar a software de código abierto con el objetivo de garantizar la seguridad nacional y lograr la independencia tecnológica en el país. Una de las tareas para lograr este objetivo es la migración a la tecnología de bases de datos PostgreSQL debido a que es el SGBD de código abierto más avanzado del mundo ya que soporta la gran mayoría de las transacciones SQL, control concurrente, ofrece modernas características como consultas complejas, disparadores, vistas, integridad transaccional y permite agregar extensiones de tipo de datos, funciones, operadores y lenguajes procedurales. (Vázquez Ortíz & Castillo Martínez, 2011) (The PostgreSQL Global Development Group, 2011). Sin embargo este sistema no tiene integrado estas técnicas de minería de datos.

Por lo cual es necesario lograr la independencia del sistema gestor de base de datos PostgreSQL para analizar los datos mediante las técnicas de minería de datos reglas de inducción y árboles de decisión.

Después de analizar la problemática, se identifica como ***problema científico:*** ¿Cómo aprovechar las potencialidades del sistema gestor de base de datos PostgreSQL para aplicar las técnicas de minería de datos reglas de inducción y árboles de decisión?

Se plantea como ***objeto de estudio*** las técnicas de reglas de inducción y árboles de decisión centrando su ***campo de acción*** en las técnicas de reglas de inducción y árboles de decisión en PostgreSQL**.**

Para darle solución al anterior problema científico se plantea como ***objetivo general*** integrar algoritmos de las técnicas de minería de datos reglas de inducción y árboles de decisión al sistema gestor de bases de datos PostgreSQL.

Objetivos específicos:

- Analizar las técnicas de minería de datos árboles de decisión y reglas de inducción.
- Implementar los algoritmos de minería de datos.
- Integrar los algoritmos implementados al SGBD PostgreSQL.
- Validar la solución propuesta.

Surge la **hipótesis** siguiente:

Si se integran algoritmos de las técnicas de minería de datos reglas de inducción y árboles de decisión al SGBD PostgreSQL se logra aprovechar las potencialidades del gestor para reducir el tiempo del proceso de análisis de los datos.

Para cumplir los objetivos se plantean las ***tareas de la investigación***. A continuación se enumeran según su orden:

1. Análisis de las técnicas de minería de datos.
2. Análisis de las herramientas de minería de datos.
3. Análisis de los lenguajes de programación y herramientas para la implementación en PostgreSQL.
4. Implementación de los algoritmos de minería de datos.
5. Integración de los algoritmos implementados al SGBD PostgreSQL.
6. Diseño del experimento para validar la solución propuesta.
7. Aplicación del experimento.
8. Análisis de los resultados del experimento.

Durante el desarrollo del presente trabajo los **métodos científicos de investigación** que fueron utilizados son:

- **Histórico-Lógico**: permitió realizar un estudio para definir las tendencias actuales de las técnicas de minería de datos y las herramientas utilizadas para la aplicación de éstas.
- **Hipotético- Deductivo:** gracias a este método se logró a partir de la hipótesis definida y siguiendo reglas lógicas de deducción se llegó a conclusiones, las que posteriormente son validadas.
- **Analítico-Sintético:** este método permitió analizar la información y arribar a conclusiones en la investigación.
- **Experimento:** con el empleo de este método se logró la evaluación de la solución propuesta determinando las relaciones entre las variables.

- **Muestra:** la investigación fue validada aplicando los algoritmos implementados a un conjunto de ejemplos almacenados en la base de datos referente a las personas con discapacidad. Para lo que se tomó una muestra de 57 600 casos.

Resultados Esperados

Al concluir la presente investigación se espera que el sistema gestor de bases de datos PostgreSQL permita realizar análisis mediante algunas técnicas de minería de datos (árboles de decisión y reglas de inducción) de forma independiente. Lo antes mencionado va a contribuir a garantizar la soberanía tecnológica del país.

Lo **novedoso** de la investigación está reflejado en la integración de varios algoritmos de las técnicas árboles de decisión y reglas de inducción con el SGBD PostgreSQL.

El presente documento se encuentra estructurado en tres capítulos.

En el capítulo 1 se realiza un estudio de las técnicas árboles de decisión y reglas de inducción así como de las herramientas más utilizadas para aplicar este proceso. También se define el lenguaje de programación y herramienta a

utilizar para implementar los algoritmos así como la metodología seleccionada para aplicar el proceso de minería de datos.

En el capítulo 2 se presenta una descripción de los algoritmos implementados incluyendo un fragmento del código, así como una breve descripción de todas las funciones implementadas para aprovechar las potencialidades del gestor. Además se muestra cómo fueron integrados todos al SGBD PostgreSQL.

En el capítulo 3 se realiza la aplicación de las técnicas de minería de datos a la base de datos de las personas con discapacidad y la validación de los algoritmos implementados mediante el diseño y aplicación de un experimento.

También contiene las conclusiones donde se exponen los principales resultados de la investigación y se proponen recomendaciones para continuar desarrollando el objeto de la investigación. Además se exponen todos los materiales consultados y referenciados, quedando organizados en referencias bibliográficas y bibliografía.

Capítulo 1: Fundamentación teórica.

Introducción

En este capítulo se exponen elementos relacionados con las técnicas de inducción de reglas y árboles de decisión precisándose términos importantes para su comprensión. Además se define el lenguaje de programación y la herramienta para implementar los algoritmos así como la selección de la metodología de desarrollo utilizada para aplicar el proceso de minería de datos.

1.1. Técnicas de minería de datos.

El descubrimiento de conocimiento en bases de datos (KDD, por sus siglas en inglés: Knowledge Discovery from Databases) se ha desarrollado en los últimos años como un proceso que consta de una secuencia iterativa de etapas o fases, las cuales son: preparación de los datos (selección y transformación), minería de datos, evaluación, interpretación y toma de decisiones.

Una de las fases más importantes dentro de este proceso es la minería de datos que integra técnicas de análisis de datos y extracción de modelos (U Fayyad, 1996), ésta se define como el análisis de archivos y bitácoras de transacciones, que trabaja a nivel del conocimiento con el fin de descubrir patrones, relaciones, reglas, asociaciones o incluso excepciones útiles para la toma de decisiones. (Rodríguez Suárez, 2009).

La minería de datos se basa en varias disciplinas, entre ellas la estadística, las bases de datos, el aprendizaje automático y otras en dependencia del negocio al cual se le aplica el proceso.

Las técnicas de minería de datos constituyen un enfoque conceptual y habitualmente son implementadas por varios algoritmos. (Molina López & García Herrero). Éstas pueden clasificarse en dependencia de su utilidad en técnicas de clasificación, predicción, asociación o de agrupamiento (clustering).

- **Las técnicas de predicción** permiten obtener pronósticos de comportamientos futuros a partir de los datos recopilados, de ahí que son aplicadas frecuentemente. Un ejemplo de la utilidad de estas técnicas es en aplicaciones para predecir el parte meteorológico o las posibles decisiones que puede tomar un cliente en determinadas circunstancias.

- **Las técnicas de agrupamiento** concentran datos dentro de un número de clases preestablecidas o no, partiendo de criterios de distancia o similitud, de manera que las clases sean similares entre sí y distintas con las otras clases. (Rodríguez Suárez, 2009). Su utilización ha proporcionado significativos resultados en lo que respecta a los clasificadores o reconocedores de patrones, como en el modelado de sistemas.
- **Las técnicas de reglas de asociación:** permiten establecer las posibles relaciones o correlaciones entre distintas acciones o sucesos aparentemente independientes; pudiendo reconocer como la ocurrencia de un suceso o acción puede inducir o generar la aparición de otros. (Molina López & García Herrero)
- **Las técnicas de clasificación:** consisten en definir una serie de clases, donde poder agrupar los diferentes casos. Dentro de este grupo se encuentran las técnicas árboles de decisión y reglas de inducción.

1.1.1 Árboles de decisión.

Un árbol de decisión es un conjunto de condiciones organizadas en una estructura jerárquica, de tal manera que la decisión final a tomar se puede determinar siguiendo las condiciones que se cumplen desde la raíz del árbol hasta alguna de sus hojas. Los árboles de decisión se utilizan desde hace siglos y son especialmente apropiados para expresar procedimientos médicos, legales, comerciales, estratégicos, matemáticos, lógicos, entre otros. (Solarte Martínez G. R., 2009). Éstos se caracterizan por la sencillez de su representación y de su forma de actuar, además de la fácil interpretación dado a que pueden ser expresados en forma de reglas de decisión.

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.

Una de las dificultades que se presenta al realizar el proceso de construcción de un árbol de decisión es escoger el atributo más apropiado. Este atributo debe ubicarse en la raíz del árbol para lo cual se debe realizar una prueba

estadística a cada uno de los atributos que permite determinar qué tan acertados se están clasificando los ejemplos de entrenamiento. Una vez que se obtiene el atributo más apropiado, se selecciona y se utiliza como nodo prueba en la raíz del árbol; luego para cada uno de los otros atributos se procede a generar un nuevo descendiente. Los datos de entrenamiento son divididos y asignados al nodo descendiente adecuado, es decir, se organizan las ramas de acuerdo al valor que toma cada atributo. Este procedimiento se realiza recursivamente en cada nodo descendiente, utilizando los datos de entrenamiento correspondientes. (Solarte Martínez & Soto Mejía, 2011).

Entre los algoritmos de árboles de decisión se encuentran el **CHAID** (Chi Squared Automatic Detector), que es un método de clasificación que utiliza estadísticos Chi-cuadrado para identificar divisiones óptimas, revisando la dependencia entre los atributos y el resultado de clasificación y seleccionando los más importantes. Además se halla el **CART** (también llamado C&RT- árbol de clasificación y regresión), que es un método que va dividiendo el árbol en particiones binarias, teniendo como objetivo reducir la impureza de los subconjuntos. También el **ID3** (dicotomía iterativa 3), que es el método desarrollado por Ross Quinlan, que selecciona los atributos divisores utilizando un análisis estadístico basado en la entropía (medida del nivel de desorden) de los conjuntos generados. (Sánchez Ramírez & Peña Villagra, diciembre, 2011).

El algoritmo ID3 se ha ido mejorando en términos de performance y nuevas funcionalidades para este algoritmo, dando origen a los algoritmos C4.5 y C5.0. Éstos están disponibles en la mayoría de las herramientas de minería de datos ya que son unos de los más divulgados debido a su gran utilidad.

Para la selección del algoritmo de árboles de decisión a implementar en la presente investigación se tuvo en cuenta el negocio donde se iba a aplicar, pues los valores de los atributos seleccionados para realizar el análisis no presentan valores continuos, ni se trabaja con fechas ni con horas y las tuplas con valores desconocidos serán eliminadas previamente. Todas estas características de los datos fuentes permiten seleccionar el algoritmo ID3.

1.2.2 Inducción de reglas.

Las técnicas de inducción de reglas permiten la generación de árboles de decisión, o reglas a partir de los datos de entrada. La información de entrada será un conjunto de casos donde se ha asociado una clasificación o evaluación a un conjunto de variables o atributos. (Omar Ruiz, 2008).

Las reglas permiten expresar disyunciones de manera más fácil que los árboles y tienden a preferirse con respecto a éstos por tender a representar "pedazos" de conocimiento relativamente independientes.

Las reglas representan funciones que establecen una relación entre los ejemplos (descritos mediante un conjunto de rasgos) y las clases de decisión. Se expresan de la forma If P then Q, donde P es la parte condicional formada usualmente por una conjunción de condiciones elementales (p1 and p2 and... pk), y Q es la parte de decisión que asigna un valor de decisión (clase) a un objeto que cumpla la condición. Las reglas constituyen patrones que establecen una dependencia entre los valores de los atributos de condición en P y el valor de decisión Q. (Filiberto, Bello, & Caballero, Oct. 2011).

Algunas de las ventajas de las reglas de inducción es que son las representaciones de hipótesis más "comprensibles" para el ser humano y el formalismo más popular de representación del conocimiento, de ahí que sean muy utilizadas en aplicaciones médicas.

Entre los algoritmos que implementan las técnicas de inducción de reglas se encuentran el **PRISM** (publicado en 1987 por Jadzia Cendrowska) basado en el algoritmo ID3, donde la meta principal del algoritmo consiste en adquirir las reglas de clasificación directamente desde el conjunto de datos de entrenamiento. (Gasparovica & Aleksejeva, 2010). Y además **OneR** que es un algoritmo de clasificación simple y efectivo de uso frecuente en el aprendizaje de máquinas, que obtiene un conjunto de reglas para el atributo con menor porciento de error de clasificación y según estudios realizados se ha demostrado que tiene un aprendizaje más rápido que el J48 (algoritmo C4.5) aunque éste lo supera en otros aspectos (Martins, 2009). También se encuentra el algoritmo **PART** que produce un conjunto de reglas del tipo If-Then que fue propuesto por Eibe Frank y Ian H. Witten en el 1998.

Para la selección de los algoritmos de inducción de reglas a implementar en la presente investigación se tuvo en cuenta que las características de los algoritmos permitieran analizar atributos con valores nominales, pues el negocio donde se van a aplicar los valores de los atributos seleccionados no presenta valores continuos, ni se trabaja con fechas, horas y las tuplas con valores desconocidos serán eliminadas previamente por lo cual se seleccionaron los algoritmos OneR y PRISM.

1.2. Herramientas para aplicar técnicas de minería de datos.

Para la aplicación de las técnicas de minería de datos existen diversas herramientas; algunas son independientes del sistema gestor de bases de datos y otras que son nativas de un gestor de bases de datos específico.

1.2.1 Herramientas nativas del gestor.

En los últimos años, empresas como ORACLE y SQL Server han incorporado algunos algoritmos o técnicas para el análisis de datos, buscando facilitar el proceso de descubrimiento del conocimiento para la toma de decisiones.

SQL Server Data Mining: es una herramienta que contiene las características necesarias para crear complejas soluciones de minería de datos ya que permite:

- Aplicar soluciones de minería de datos utilizando Microsoft Excel.
- Entender cómo, cuándo y dónde aplicar los algoritmos que se incluyen en el servidor de SQL.
- Realizar la extracción de datos de procesamiento analítico en línea (OLAP).
- Utilizar SQL Server Management Studio para acceder y proteger los objetos de minería de datos.
- Utilizar SQL Server Business Intelligence Development Studio para crear y gestionar proyectos de minería de datos.

(MacLennan, Tang, & Crivat, 2009).

Algunas de las ventajas de la minería de datos de Microsoft es la integración estrecha con la plataforma de base de datos de clase mundial SQL Server, ya

que aprovecha el desempeño, la seguridad y las características de optimización de SQL Server; extensibilidad ya que se puede extender la minería de datos de Microsoft para implementar algoritmos que no vienen incluidos en el producto.

Los algoritmos implementados por Microsoft son:

- Árboles de decisión.
- Bayes naive.
- Clústeres.
- Redes neuronales.
- Serie temporal.
- Regresión lineal.
- Clústeres de secuencia.
- Asociación

Oracle Data Mining: permite que las empresas creen aplicaciones de inteligencia de negocio avanzadas que explotan las bases de datos corporativas, descubren nuevos conocimientos e integran esa información en aplicaciones comerciales. (Haberstroh, 2008)

Oracle Data Mining incorpora las siguientes funcionalidades de minería de datos para la realización de clasificaciones, agrupamiento, predicciones y asociaciones.

- Agrupamiento(K-Means, O-Cluster).
- Árboles de Decisión.
- Atributo Relevante.
- Característica de Selección.
- Clasificador Bayesiano (Naive Bayes).
- Máquinas de Soporte Vectorial (Support Vector Machines).
- Modelos Lineales Generalizados.
- Reglas de Asociación(Apriori).

Para un mayor estudio de las anteriores técnicas ver el anexo 2 del presente documento.

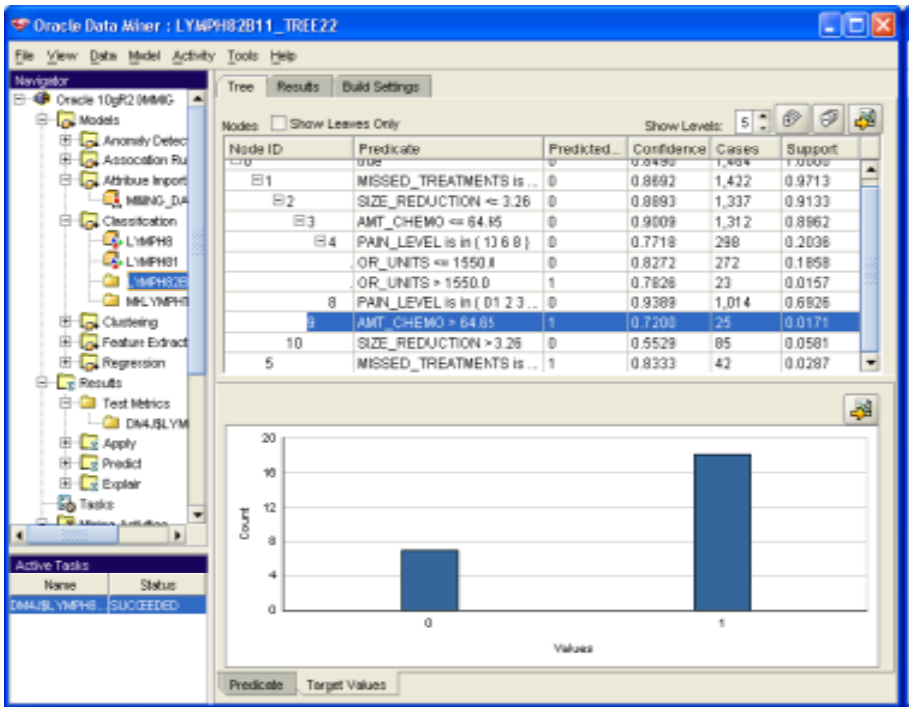

Figura 1: Herramienta Oracle Data Mining (Haberstroh, 2008).

Todas las funciones de los modelos son accesibles a través de una API basada en Java.

El carácter nativo de la solución es un plus fuerte, en tanto que las implementaciones de cada una de las etapas del proceso se encuentran incluidas en el motor.

1.2.2 Herramientas independientes del gestor.

Entre las herramientas libres más utilizadas para la minería de datos se encuentran Weka (Waikato Environment for Knowledge Analysis), que es una herramienta visual de distribución libre para el análisis y extracción de conocimiento a partir de datos. (Ver figura 2) (V.Ramesh, 2011).

Figura 2: Herramienta libre independiente del SGBD para aplicar las técnicas de minería de datos.

Las principales ventajas de la herramienta son:

- Es multiplataforma.
- Contiene una extensa colección de técnicas para pre-procesamiento y modelado de datos.
- Es fácil de usar, gracias a su interfaz gráfica.
- Soporta varias tareas de minería de datos, especialmente pre-procesamiento, agrupamiento, clasificación, regresión, visualización y selección.
- Permite la combinación de varios algoritmos basados en técnicas de minería de datos, para obtener mejores resultados en el descubrimiento de conocimiento.
- Es capaz de mostrar los datos en varios tipos de gráficos, con el objetivo de una mejor comprensión y análisis.

YALE Rapid Miner:

La herramienta fue desarrollada en el 2001 por el departamento de inteligencia artificial en la universidad de Dortmund en Java, es multiplataforma, es un software de código abierto GNU y con licencia GPL. En la última versión incluye características como las de implicar nuevos formatos de entrada de datos con operadores para Microsoft Excel y SPSS. Desde la perspectiva de la visualización YALE ofrece representaciones de datos en dispersión en 2D y 3D,

representaciones de datos en formato SOM (Self Organizing Map), coordenadas paralelas y grandes posibilidades de transformar las visualizaciones de los datos.

De forma general se puede decir que las herramientas de minería de datos de ORACLE y SQL Server son herramientas muy potentes, y que una de sus mayores fortalezas radica en la integración con el sistema gestor de base de datos, pero ambas son herramientas propietarias y muy costosas para las empresas cubanas. Por otra parte, herramientas como Weka y YALE Rapid Miner, son herramientas que a pesar de ser de libres y poseer diversas ventajas tienen las desventajas de que el proceso es engorroso ya que requiere de tiempo para la preparación, la vinculación de los datos con el gestor, extendiendo así el tiempo de respuesta de los análisis por lo cual es necesario realizar la integración de los algoritmos de minería de datos al SGBD PostgreSQL. (Robles Aranda, 2012)

1.3. Tecnologías utilizadas para el desarrollo de los algoritmos.

1.3.1 PostgreSQL.

PostgreSQL es un sistema gestor de bases de datos objeto-relacional (ORDBMS, Object Relational DataBase Management System por sus siglas en inglés), que está considerado como el sistema de base de datos de código abierto más avanzado del mundo porque proporciona un gran número de características que normalmente sólo se encontraban en los sistemas de bases de datos comerciales tales como DB2 u Oracle4. Estas características son DBMS Objeto-Relacional, Integridad Referencial, Lenguajes Procedurales y Cliente/Servidor (Domínguez-Rodríguez & Téllez-Sánchez, septiembre 2011).

A continuación se describen cada una de estas características:

- PostgreSQL es un SGBD Objeto-Relacional porque aproxima los datos a un modelo objeto-relacional y es capaz de manejar complejas rutinas y reglas. Como por ejemplo soporta consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, transactions, optimización de consultas, herencia y arreglos.

- La integridad referencial es utilizada para garantizar la coherencia de datos entre relaciones aparejadas.
- PostgreSQL soporta lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es similar al lenguaje procedural de Oracle, PL/SQL. Además permite usar Perl, C Python, o TCL como lenguaje procedural.
- La arquitectura cliente/servidor de PostgreSQL es similar al método del Apache 1.3.x para manejar procesos. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL.

1.3.2 Lenguajes de programación PL/pgSQL.

PL/pgSQL es un lenguaje procedural que se estructura en bloques y añade estructuras de control al lenguaje SQL, permite realizar cálculos complejos, el manejo de cadenas y consultas dentro del servidor de la base de datos, combinando el poder de un lenguaje procedimental y la facilidad de uso de SQL. Además se pueden escribir las palabras claves y los identificadores mezclando letras mayúsculas y minúsculas. Sus principales ventajas son: mayor rendimiento, soporte SQL y portabilidad.

El lenguaje PL/pgSQL es uno de los más utilizados dentro de PostgreSQL, debido a que guarda cierta similitud con PL/SQL de Oracle y a su facilidad de uso.

1.3.3 Herramienta pgAdminIII.

PGAdminIII es un cliente gráfico para el gestor de bases de datos PostgresSQL, es multiplataforma, y está diseñada para responder las necesidades de los desarrolladores siendo la más completa y popular con licencia Open Source y fue desarrollada en C++. La herramienta permite agregar funcionalidades, gestionar todos los objetos de la BD e incluye un editor SQL con resaltado de sintaxis. (Robinson, 2011).

Entre sus principales funcionalidades se encuentran que posee interfaz administrativa gráfica, herramienta de consulta SQL, editor de código procedural, agente de planificación SQL/shell/batch y administración de Slony-I.

1.4. Metodologías.

Para realizar el proceso de extraer conocimiento útil a partir de los datos almacenados existen metodologías que permiten llevar a cabo el proceso de minería de datos en forma sistemática y no trivial. Éstas ayudan a las organizaciones a entender el proceso de descubrimiento de conocimiento y proveen una guía para la planificación y ejecución de los proyectos. (Moine, Haedo, & Gordillo, 2011).

Dentro de las metodologías existentes se destacan SEMMA y CRISP-DM.

SEMMA, de las siglas en inglés (Sample, Explore, Modify, Model, Assess), es una metodología orientada a seleccionar, explotar y modelar un gran conjunto de datos propuestos por SAS Institute; con el objetivo de guiar el proceso de descubrimiento de patrones. Ésta cuenta con varias faces extracción de una muestra de los datos, si se cuenta con la muestra se propone explotar los datos en orden a simplificar el modelo, luego se seleccionan el algoritmo de explotación de datos más adecuado. La cuarta fase está orientada a ejecutar el algoritmo seleccionado con la muestra. La última fase consiste en la evaluación de los resultados. Esta metodología se centra más en las características técnicas del desarrollo del proceso.

CRISP-DM de las siglas en inglés (Cross-Industry Standard Process for Data Mining) fue desarrollada en 1999 por un grupo de empresas europeas como una metodología de libre distribución.

CRISP-DM tiene 6 faces encargadas de analizar el problema, analizar los datos, prepararlos, realizar la modelación, la evaluación, y explotación.

En la primera fase, comprensión del problema, esta metodología propone el estudio de los objetivos y requerimientos del proyecto para definir un plan preliminar para alcanzar los objetivos. En la fase de comprensión de los datos se recolectan los datos a utilizar, se describen los datos y se verifica la calidad de los mismos. La fase de preparación de los datos incluye la integración, selección, limpieza y transformación de los datos. En la fase de la modelación se seleccionan las técnicas y algoritmos a utilizar en dependencia del objetivo a resolver luego se pasa a evaluar comprobando que se cumplan los objetivos

del negocio y por último se le presentan al usuario informes sobre el conocimiento extraído.

Para la aplicación de los algoritmos de minería de datos implementados se va a seleccionar la metodología CRISP-DM porque realiza un análisis global del negocio al cual se va aplicar.

Conclusiones:

En el presente capítulo se analizaron las técnicas de minería de datos, profundizando en los árboles de decisión y reglas de inducción determinando que de éstas se iban a aplicar los algoritmos 1R, PRISM e ID3. Se estudiaron las herramientas utilizadas para la aplicación de estas técnicas de las cuales Oracle Data Mining y SQL Analysis Server presentan grandes ventajas gracias a su condición de estar integradas al SGBD, pero con el inconveniente de ser propietarias por lo cual es necesario implementar los algoritmos 1R, PRISM e ID3 para integrarlos al SGBD PostgreSQL. Se seleccionó la herramienta PgAdmin III y el lenguaje PL/pgSQL para la implementación de los algoritmos al SGBD y como metodología CRISP-DM para la aplicación de los algoritmos.

Capítulo 2: Propuesta de solución.

Introducción:

En el presente capítulo se presenta una breve descripción de los algoritmos de minería de datos implementados en la investigación, así como parte del código desarrollado. Además se muestra cómo se realiza la integración con el SGBD PostgreSQL.

2.1 Algoritmo 1R.

El algoritmo 1R propuesto por Robert C. Holte en 1993 es un clasificador muy sencillo, que únicamente utiliza un atributo para la clasificación. A pesar de que el autor lo cataloga como: "*Program 1R is ordinary in most respects.*" sus resultados pueden ser muy buenos en comparación con algoritmos mucho más complejos y su rendimiento promedio está por debajo de los de C4.5 en 5,7 puntos porcentuales de aciertos de clasificación según los estudios realizados por el autor del algoritmo. (HOLTE, 1993).

```
1R (ejemplos) {
  Para cada atributo (A)
    Para cada valor del atributo (Ai)
      Contar el número de apariciones de cada clase con Ai
      Obtener la clase más frecuente (Cj)
      Crear una regla del tipo Ai -> Cj
    Calcular el error de las reglas del atributo A
  Escoger las reglas con menor error
}
```

Figura 3: Pseudocódigo del algoritmo 1R (Molina López & García Herrero).

La implementación del algoritmo 1R se realizó utilizando el pseudocódigo mostrado en la figura 3. Esta función sólo permite trabajar con tablas que tengan atributos nominales y en la misma no debe haber atributos con valores desconocidos para obtener el resultado deseado.

La función toma como entrada el nombre de la tabla y la clase sobre la cual se va a realizar el análisis y se devuelve como resultados un conjunto de reglas para los atributos con la menor cantidad de errores.

A continuación se muestra un fragmento de la función implementada.

```
BEGIN
  total:=0;
  cont :=0;
  create temp table abab (atr varchar, val varchar, cannt integer, ero real
);
  create temp table abc (atribut varchar, totalero real );
  SELECT count(column_name) into cant_columna FROM
information_schema.columns where columns.table_name =$1;
  open columnas for SELECT column_name, table_name FROM
information_schema.columns where columns.table_name =$1;
  loop
    fetch columnas into atrib;
    IF (cant_columna-1) = cont THEN
    EXIT;
    else
      total:=0;
     For valores in  EXECUTE 'select distinct ' ||atrib.column_name || '
from ' ||atrib.table_name
      Loop
      aux :=0;
      EXECUTE 'select count(*) from ' ||$1|| ' where '
||atrib.column_name|| ' = ' || '''' ||valores|| ''''  into err;
      for valorclase in execute 'select distinct ' ||$2|| ' from '
||atrib.table_name|| ' where '||atrib.column_name|| '= '|| '''' ||valores||
''''
      loop
              EXECUTE  'select count(*) from ' ||$1|| ' where ' ||$2|| '=
'|| '''' ||valorclase||''''|| ' and ' ||atrib.column_name|| ' = ' || ''''
||valores|| '''' into cant  ;

              if cant > aux  then
              error:=err-cant;
              aux:= cant;
              aux2:= valores;
              aux3:=valorclase;
              reglas:=valores||'->'||valorclase;
              error1:=error/err;
              end IF;
```

Figura 4: Código del algoritmo 1R en el lenguaje Pl/pgsql.

2.2 Algoritmo PRISM.

El algoritmo PRISM es un algoritmo de cubrimiento sencillo que asume que no hay ruido en los datos. El objetivo de éste es ir creando reglas perfectas que maximicen p/t siendo p la cantidad de ejemplos positivos cubiertos por la regla y t la cantidad de ejemplos cubiertos por la regla. (Chesñevar, 2009)

Este algoritmo tiene como característica que va eliminando los ejemplos que va cubriendo por las reglas conformadas, por lo cual las reglas deben mostrarse e interpretarse en el orden que se van cubriendo.

```
PRISM (ejemplos) {
   Para cada clase (C)
      E = ejemplos
      Mientras E tenga ejemplos de C
         Crea una regla R con parte izquierda vacía y clase C
         Hasta R perfecta Hacer
            Para cada atributo A no incluido en R y cada valor v de A
               Considera añadir la condición A=v a la parte izquierda de R
               Selecciona el par A=v que maximice p/t
                  (en caso de empates, escoge la que tenga p mayor)
            Añadir A=v a R
      Elimina de E los ejemplos cubiertos por R
```

Figura 5: Pseudocódigo del algoritmo PRISM (Molina López & García Herrero)

La implementación del algoritmo PRISM se realizó utilizando el pseudocódigo de la figura 5. Únicamente permite atributos nominales y no puede haber atributos con valores desconocidos para obtener el resultado deseado.

La función toma como entrada el nombre de la tabla y la clase sobre la cual se va a realizar el análisis y se devuelve como resultados un conjunto de reglas las cuales se deben interpretar en el orden en que aparecen como estipula el algoritmo.

A continuación se muestra un fragmento de la función implementada en la figura 6.

```
for vc in execute 'select distinct ' ||$2|| ' from  ejemplo'
 loop
 delete from ejemplo;|
 execute 'insert into ejemplo select * from ' ||$1;
 y:=0;
 cont:=0; e:=4;
  while e > 0
  loop
    atrib:='';
    valores:='';
    while  y <> 1
     loop
          truncate table temporal;
          For atrib in SELECT column_name FROM information_schema.columns where
columns.table_name ='ejemplo'
          loop
               For valores in  EXECUTE 'select distinct ' ||atrib || ' from
ejemplo'
               Loop
                 if atrib <> atr and atrib <> $2  then
                 EXECUTE 'select count(*) from ejemplo where ' ||condic||atrib|| '
= ' || '''' ||valores|| ''''  into t;
                   execute 'select count(*) from ejemplo where ' ||condic||$2|| '=
'|| '''' ||vc||''''|| ' and ' ||atrib|| ' = ' || '''' ||valores|| '''' into p  ;
                 if p=0 or t=0 then
                 pt:=0;
                 else
                 pt:=p/t;
                 end if;
                 insert into temporal values (atrib, valores, p, t, pt);
                 end if ;
               end loop;
           end loop;

     select max(ag) into y from temporal limit 1;  -- condicionar contando
     select ac from temporal where ag =(select max(ag) from temporal limit 1)into
atributo;
```

Figura 6: Código del algoritmo PRISM en el lenguaje Pl/pgsql.

2.3 Algoritmo ID3.

El ID3 fue propuesto por J. R Quinlan en 1986, este es un algoritmo simple y, a la vez potente, cuya misión es la elaboración de un árbol de decisión como un método para aproximar una función objetivo de valores discretos, que es resistente al ruido en los datos y que es capaz de hallar o aprender de una disyunción de expresiones.

Para decidir qué atributo es el más apropiado a usar en cada nodo del árbol se utiliza una propiedad estadística llamada ganancia de información, que mide qué tan bien clasifica ese atributo a los datos de entrenamiento. Así que elije el nodo del árbol que tenga mayor ganancia de información y luego expande sus ramas utilizando la misma metodología. La ganancia de información es una diferencia de entropías. El concepto de entropía se basa en la teoría de la

información. Esta teoría fue desarrollada inicialmente por Claude Shannon a mediados del siglo XX (Solarte Martínez & Soto Mejía, 2011).

1. Seleccionar el atributo A_i que maximice la ganancia $G(A_i)$.
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 A_i.
4. Por cada sucesor:
    a. Si sólo hay ejemplos de una clase, C_k, entonces etiquetarlo con C_k.
    b. Si no, llamar a ID3 con una tabla formada por los ejemplos de ese nodo, eliminando la columna del atributo A_i.

Figura 7: Pseudocódigo del algoritmo ID3 (Molina López & García Herrero).

La implementación de algoritmo ID3 se realizó utilizando el pseudocódigo de la figura 7. Únicamente permite atributos nominales y no puede haber atributos con valores desconocidos para obtener el resultado deseado.

La función toma como entrada el nombre de la tabla y la clase sobre la cual se va a realizar el análisis y se devuelve como resultados un conjunto de reglas derivadas del árbol de decisión.

A continuación se muestra un fragmento de la función implementada (ver figura 8).

```
while finc > 0
 loop
    delete from aa;
    execute 'select count(*) from ejemplo where ' ||$2||' in (''Sí'', ''+'',
''Si'', ''p'')' into p;
    execute 'select count(*) from ejemplo where ' ||$2||' in (''No'',
''-'',''n'')' into n;
    pn:=p+n;
    if p=0 or n=0 then
     i:=0;
          else
     i:= -(p/pn)*(log(2,p)-log (2,pn))-(n/pn)* (log (2, n) - log
(2,pn));--informaci'on
     end if;
     cont:=0;
    SELECT count(column_name) into cant_columna FROM information_schema.columns
where columns.table_name = 'ejemplo';
    open columnas for SELECT column_name, table_name FROM
information_schema.columns where columns.table_name ='ejemplo';
     loop
      e :=0;
      fetch columnas into atrib;
       IF  (cant_columna-2) = cont   THEN
        exit;
       else
       if atrib.column_name <> $2 then --agregue
        For valores in  EXECUTE 'select distinct ' ||atrib.column_name || ' from
ejemplo'
         Loop

          EXECUTE 'select count(*) from ejemplo where ' ||atrib.column_name|| ' =
' || '''' ||valores|| ''' and '||$2||' in (''Sí'', ''+'',''Si'',''p'')'  into p1;
          EXECUTE 'select count(*) from ejemplo where ' ||atrib.column_name|| ' =
' || '''' ||valores|| ''' and ' ||$2||' in (''No'', ''-'',''n'')' into n1;

          pn1:=p1+n1;
          if p1=0 or n1=0 then
           i1:=0;
```

Figura 8: Código del algoritmo ID3 en el lenguaje Pl/pgsql.

2.4 Mecanismos para optimizar el rendimiento de los algoritmos implementados.

Generalmente las tablas sobre las cuales se realizan análisis de minería de datos cuentan con un gran volumen de información, lo cual puede retrasar el resultado de dicho estudio.

Una de las opciones que brinda PostgreSQL para mejorar el rendimiento en estos casos es el particionado o el indexado de las tablas para obtener un mejor desempeño a la hora de consultarlas.

2.4.1 Particionado de tablas.

.

El particionado de tablas es una técnica que consiste en descomponer una enorme tabla (padre) en un conjunto de tablas hijas. Esta técnica trae como resultado reducir la cantidad de lecturas físicas a la base de datos cuando se ejecutan las consultas.

En PostgreSQL los tipos de particionado existentes son por rango y lista. (The PostgreSQL Global Development Group, 2011)

- Particionado por rango: se crean particiones mediantes rangos definidos en base a cualquier columna que no se solape entre los rangos de valores asignados a diferentes tablas hijas.

- Particionado por lista: Se crean particiones por valores.

En la presente investigación se implementa una función que permite realizar el particionado de la tabla según los valores de la clase (particionado por lista). Lo cual permite agilizar la búsqueda a la hora de clasificar.

La función tiene como parámetros de entrada el nombre de la tabla a la que se le desea hacer las particiones y el nombre de la clase, creándose tantas particiones como valores tenga la clase. (Ver figura 9).

Las tablas creadas por la función tendrán como nombre la concatenación de máster más el antiguo nombre de la tabla para el caso de la tabla padre y para las hijas es la concatenación del antiguo nombre de la tabla más el valor de la clase por la cual se creó la partición.

```
DECLARE
  valor varchar ;
  parametros VARCHAR:='';
  c varchar;
  cant integer;
BEGIN
execute 'create table master'||$1||' as select * from '||$1|| ' limit 0' ;
 For valor in  EXECUTE 'select distinct ' ||$2 || ' from ' ||$1
 loop
   parametros:='';
   execute 'create table ' ||$1||valor||' as select * from '||$1|| ' where
'||$2||' = '||'''' ||valor|| ''''  ;
   execute 'ALTER TABLE '||$1||valor|| ' ADD CONSTRAINT ' ||$1||valor||'_chk
CHECK ('||$2||' = '||''''' ||valor|| ''''||')';
   execute 'ALTER TABLE '||$1||valor|| ' INHERIT master'||$1;
   execute 'SELECT count(*)FROM information_schema.columns WHERE
columns.table_name = '||''''||$1||''''' into cant;
   for c in SELECT  information_schema.columns.column_name FROM
information_schema.columns WHERE  columns.table_name = $1 order by
ordinal_position
   loop
    if cant=1 then
    parametros:= parametros||'new.'||c ;
    else
    parametros:= parametros||'new.'||c||',';
    end if;
    cant:=cant-1;
   end loop;
   execute 'CREATE OR REPLACE RULE ' ||$1||valor||'1 AS ON INSERT TO
master'||$1||' WHERE ('||$2||' = '||''''' ||valor|| ''''||')'|| ' DO INSTEAD
  INSERT INTO ' ||$1||valor|| ' VALUES ('||parametros||')';
 end loop;
 raise notice'Debe verificar que este habilitado a "on" el parámetro
constraint_exclusion (SET constraint_exclusion = on;) en el archivo de
configuración postgresql.conf para que las consultas sean optimizadas.';
END;
```

Figura 9: Código de la función para crear particiones en el lenguaje Pl/pgsql.

2.4.2 Indexado de Tablas.

Los índices son objetos de las bases de datos que permiten agilizar las operaciones, permitiendo un rápido acceso a los datos, evitando sobrecargar el CPU ya que sin índice, se debe recorrer secuencialmente toda la tabla para mostrar el resultado deseado. Éstos se definen sobre las columnas más consultadas.

PostgreSQL tiene diferentes métodos de acceso para crear los índices como el Btree, Rtree y Hash. El Btree se usa para operadores de igualdad o de rango de datos para lo cual utilizan los siguientes operadores <,<=,=,>,=,> , el Hash sólo usa simples comparaciones de igualdad, Gist los operadores en que pueden variar en dependencia de la estrategia de indexación (clase de

operador). Soporta consultas usando: <<, &<, &>, >>, <<|, &<|, |&>, |>>, @>, <@, ~=, &&.

En la presente investigación se implementa una función que permite crear índices parciales según los valores de la clase que utilicen el método de acceso Btree. Lo cual permite acelerar la búsqueda a la hora de clasificar.

La función tiene como parámetro de entrada el nombre de la tabla a la que se le desea crear el índice y el nombre de la clase. (Ver figura 10).

```
CREATE OR REPLACE FUNCTION public.indices (
  tabla varchar,
  clase varchar
)
RETURNS void AS
$body$
DECLARE
  valor varchar ;
  parametros VARCHAR:='';
  c varchar;
  cant integer;
BEGIN
For valor in  EXECUTE 'select distinct ' ||$2 || ' from ' ||$1
 loop
   execute ' CREATE INDEX idx_' ||valor|| ' ON '||$1|| '( '||$2|| ')
where '||$2||  ' = '
      ||'''' ||valor|| ''''  ;
 end loop;
END;
$body$
LANGUAGE 'plpgsql'
VOLATILE
CALLED ON NULL INPUT
SECURITY INVOKER
```

Figura 10: Código de la función para crear índices en el lenguaje Pl/pgsql.

2.5 Integración de los algoritmos al SGBD PostgreSQL.

A partir de la versión 9.1 PostgreSQL brinda facilidades para que los usuarios puedan crear, cargar, actualizar y administrar extensiones utilizando el objeto de base de datos EXTENSION. (PostgreSQL, 2011).

Entre las ventajas de esta nueva funcionalidad se encuentra, que en lugar de ejecutar un script SQL para cargar objetos que estén “separados” en su base de datos, se tendrá la extensión como un paquete que contendrá todos los objetos definidos en ella, lo cual trae gran beneficio al actualizarla o eliminarla ya que por ejemplo se pueden eliminar todos los objetos utilizando el comando DROP EXTENSION sin necesidad de especificar cada uno de los objetos

definidos dentro de la extensión, además se cuenta con un repositorio para obtener extensiones y contribuir con éstas (http://pgxn.org/).

La integración de los algoritmos implementados con el SGBD se va a realizar mediante la creación de una extensión por las ventajas que PostgreSQL brinda para su creación.

2.5.2 Creación de la extensión minería de datos.

Para la creación de la extensión se crean dos archivos, en el primero se definen las características de la extensión y en el segundo los objetos SQL que se desean agregar. Los mismos deben ser ubicados dentro del directorio de la instalación “C:\Archivos de programa\PostgreSQL\9.1\share\extension”

En el archivo “mineria_datos.control” creado para agregar la extensión donde se cargarán las funciones de los algoritmos implementados se definieron los siguientes parámetros:

- Comment: una breve descripción sobre el contenido de la extensión creada.
- Encoding: El tipo de codificación utilizado.
- Default_version: La versión de la extensión.
- Schema: El esquema donde se almacenarán los objetos creados por la extensión.

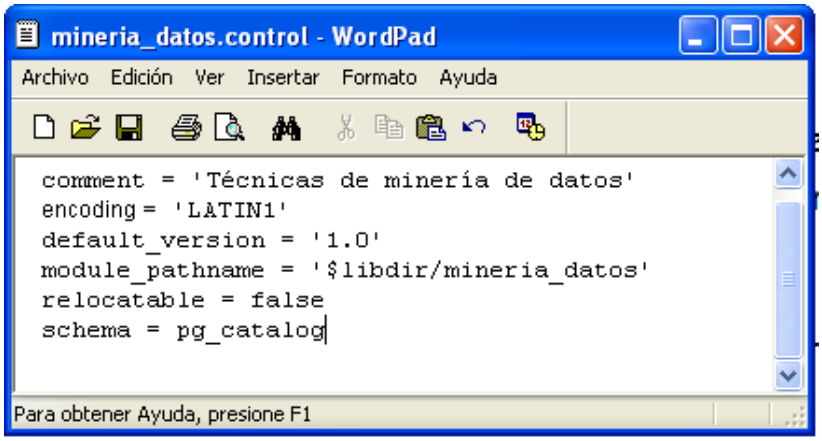

Figura 11: Archivo que contiene las características de la extensión.

Una vez definido en el archivo “mineria_datos.control” se especifica el archivo que contendrá el código de las funciones desarrolladas “mineria_datos--1.0.sql”

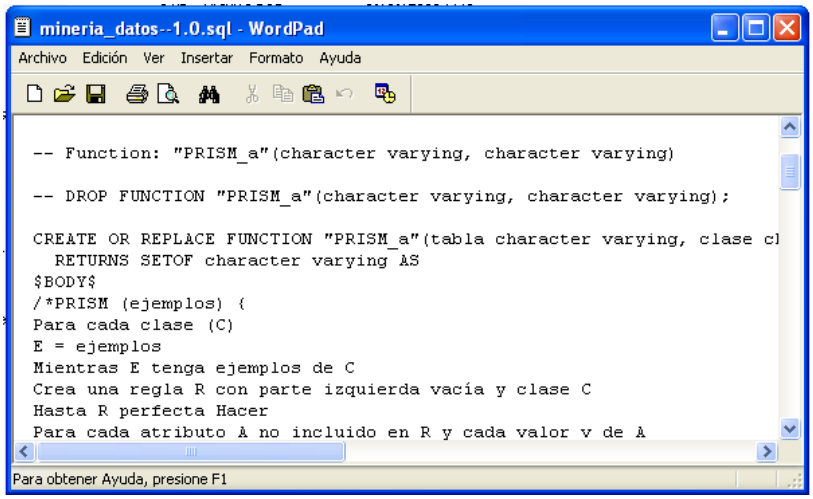

Figura 12: Archivo que contiene el código los objetos definidos en la extensión.

2.5.3 Trabajo con la extensión de minería de datos.

Para que los usuarios puedan utilizar la extensión de minería de datos simplemente deben ejecutar el comando “CREATE EXTENSION mineria_datos” que cargará la extensión como se puede apreciar en la imagen 13.

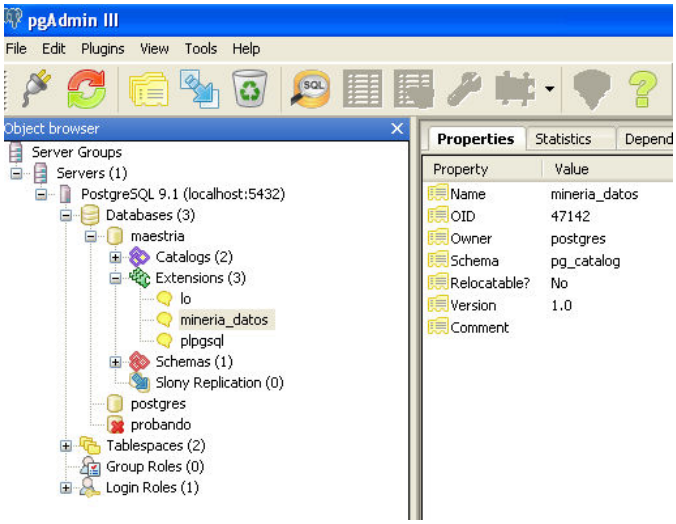

Figura 13: La extensión "mineria_datos" creada.

Para consultar las funciones agregadas por la extensión se debe consultar en el esquema pg_catalog la carpeta de funciones como se muestra en la figura 14.

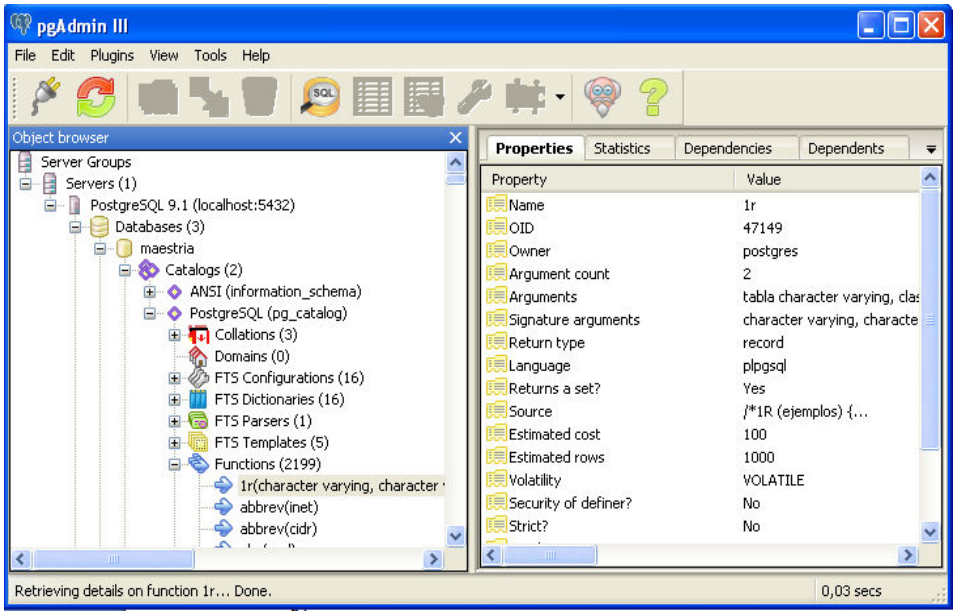

Figura 14: Funciones de la extensión "mineria_datos" ubicada en el esquema pg_catalog.

Conclusiones:

En el presente capítulo se realizó un análisis de los algoritmos implementados exponiendo sus características, ventajas y el pseudocódigo de cada uno de ellos. Se mostró además parte del código de los tres algoritmos implementados. Para aprovechar las potencialidades del SGBD se implementaron dos funciones que permiten realizar el particionado de tabla por lista y el indexado de las tablas por el atributo clase como mecanismo para disminuir los tiempos de análisis de los algoritmos implementados, y por último se describe cómo a través de la creación de una extensión se integraron los algoritmos al SGBD PostgreSQL.

Capítulo 3: Aplicación y validación de la solución propuesta.

Introducción:

En el presente capítulo se va a realizar el proceso de minería de datos a los estudios de genética médica utilizando la metodología CRISP-DM y los algoritmos implementados. Además se va a realizar un experimento para comparar los resultados de los algoritmos implementados con otra herramienta y se va a demostrar cómo influyen los mecanismos implementados en el rendimiento de los algoritmos.

3.1 Genética médica.

De las técnicas de minería de datos aplicadas a la medicina para el análisis de los datos se destacan las de árboles de decisión y reglas de inducción, las cuales han sido utilizadas en diferentes investigaciones como en el diagnóstico de la necesidad de administrar fármacos en pacientes con síntomas de enfermedad cardiovascular (Solarte Martínez & Soto Mejía, 2011), en el diagnóstico de personas con glaucoma neo-vascular (Landín Sorí & Romero Sánchez, 2012), en la clasificación del dengue hemorrágico (Vega Riverón & Sánchez Valdés, 2012), en la compresión de señales de banda como la imagen médica y estudios sobre el cáncer de ovario (Bertenshaw1 & Yip1, 2010).

Entre las ramas de la medicina se encuentra la genética médica, la cual tiene como objetivo principal en Cuba estudiar cómo garantizar la reducción del impacto de las enfermedades genéticas sobre la salud y el bienestar de los individuos a través de estrategias de prevención. Ello permite ayudar a las personas con “desventajas” genéticas a vivir y a reproducirse de forma tan normal como sea posible, así como reducir la frecuencia y las manifestaciones clínicas de los defectos congénitos severos. (Marcheco Teruel, 2008) Una de las acciones realizadas por el Centro Nacional de Genética Médica para alcanzar este objetivo ha sido la informatización de varios estudios realizados junto a la Universidad de las Ciencias Informáticas, pero debido al volumen de la información recopilada se decidió analizar el comportamiento a través de las técnicas de análisis de datos.

A continuación se muestra cómo se aplicó el proceso de minería de datos, utilizando la metodología CRISP-DM y los algoritmos integrados al SGBD, a la base de datos de la aplicación de genética médica.

3.2 Comprensión del negocio.

Los **objetivos del negocio** son enmarcados a:

1. Conocer las relaciones que se establecen entre las características de los pacientes.
2. Determinar cuáles son las características de las personas que pueden influir en el grado de la discapacidad intelectual.

Como **objetivo de minería de datos** se identificó:

- Obtener reglas que permitan descubrir la influencia que tienen en el grado de la discapacidad intelectual parámetros como el color de la piel, si la madre tuvo enfermedades infecciosas, si ingirió medicamentos, si recibió calor o radiación durante el embarazo, si hubo complicaciones al nacer el paciente o si existen antecedentes de personas con discapacidad intelectual en la familia.

3.3 Comprensión de los datos.

La comprensión de datos está relacionada con la recolección y descripción de la información inicial con la que se comienza el proceso de obtener conocimiento, una vez establecidos los objetivos a seguir. Además, se desarrollan actividades que permiten su exploración, a fin de identificar problemas con su calidad. (Brito Sarasa, 2008).

La información de la estructura de la base de datos utilizada fue obtenida de la investigación realizada en el 2009 por Félix Mario Marrero Hernández (Marrero Hernández, 2009).

La información recopilada fue obtenida de una única fuente, que de manera centralizada es la empleada para recopilar los datos de los pacientes. Esta base de datos cuenta con 142 tablas.

Inicialmente la base de datos se encontraba sobre el gestor MySQL pero se migró a PostgreSQL con el objetivo de aprovechar las potencialidades del

SGBD. A continuación se describen los atributos significativos para darle solución a los objetivos propuestos en el epígrafe 3.1.

Tabla 1: Resumen descriptivo de la información relevante recopilada.

Nombre del Atributo	Tipo de dato	Descripción	Nombre de la Tabla
Folio	character varying	Almacena el identificador de cada paciente	
Color_Piel	character varying		tcolor_piel
Tipo_Complic	character varying		tcomplic_r_nac
Enferm_Infec_Embar	character	Almacena si la madre durante el embarazo tuvo enfermedades infecciosas	tmadre_emb
ing_medicamentos	character	Almacena si la madre durante el embarazo ingirió medicamentos	tmadre_emb
calor_mad_emb	character	Almacena si la madre estuvo expuesta durante el embarazo a calores	tmadre_emb
radiaciones	character	Almacena si la madre estuvo expuesta durante el embarazo a	tmadre_emb

	radiaciones	
Tipo_Calor_Madre	Indica el tipo de calor	ttipo_calor_madre
Antecedentes RM	Almacena si el paciente ha tenido familiares con discapacidad intelectual	tdatosclinicos
Id_Grado_De_RM	Identificador de la discapacidad intelectual.	trm
Grado_De_RM	Almacena el grado de la discapacidad intelectual	tgrado_rm

Explorar y verificar la calidad de los datos.

Para realizar el análisis el juego de datos utilizado para mostrar en la investigación es ficticio, debido a la sensibilidad de los datos almacenados en la base de datos sobre la cual se está realizando el estudio.

Los resultados de la exploración realizada a los datos para verificar su calidad:

- De los 10 630 registros de la tabla tmadre_emb sólo 4 tienen los campos de los atributos radiaciones, calor_mad_emb, Enferm_Infec_Embar, ing_medicamentos vacíos.
- Los atributos Id_Grado_De_RM, Antecedentes RM, Tipo_Calor_Madre, Tipo_Complic, Color_Piel y folio no tienen ningún valor vacío.

3.4 Preparación de datos.

3.4.1. Selección de datos.

De los atributos vistos en el epígrafe 3.2 se excluyeron de la selección:

- Folio: se empleó sólo como enlace entre las tablas. Presenta gran variabilidad pues cada instancia en única.
- Id_Grado_De_RM : se empleó sólo como enlace entre las tablas.
- Tipo_Calor_Madre: No aporta información relevante, y es muy variable.

3.4.2. Limpieza y transformación de datos.

Para la limpieza de los datos se aplicó un filtro para eliminar las instancias que tenían valores vacíos.

Para un mayor entendimiento en el trabajo con los valores de los atributos se realizaron las siguientes transformaciones:

- Enferm_Infec_Embar, ing_medicamentos, calor_mad_emb y radiaciones toma los valores de 1, 2 y 0 que se sustituyen por los valores Sí, No y No Sabe respectivamente.

3.4.3. Integración de datos.

Los diferentes campos seleccionados se encuentran en diferentes tablas por lo cual se implementó una tabla como resultado de una vista que contiene **57 600** registros con un total de 7 campos.

3.5 Modelado.

La selección de las técnicas de minería de datos y algoritmos a emplear depende de los objetivos de minería de datos propuestos en la fase de comprensión del negocio.

Las técnicas seleccionadas para darle cumplimiento a este objetivo son árboles de decisión y reglas de inducción y de ellas los algoritmos ID3, 1R y PRISM.

Para obtener las reglas mediante el algoritmo 1R integrado al SGBD PostgreSQL se realizó la siguiente consulta “Select * from 1r (‘casos, ‘grm’)”

para ejecutar la función implementada pasándole como parámetros el nombre de la tabla y atributo clase. El resultado obtenido por la función es:

"ei"= ‘No’->Moderado

"ei"= ‘No Sabe’->Ligero

"ei"= ‘Si’->Severo

Estos resultados se pueden interpretar de la siguiente forma:

1. Si la madre no tuvo enfermedades infecciosas durante el embarazo entonces el grado de la discapacidad intelectual es Moderado.
2. Si la madre tuvo enfermedades infecciosas durante el embarazo entonces el grado de la discapacidad intelectual es Severo.
3. Si no se sabe si la madre no tuvo enfermedades infecciosas durante el embarazo entonces el grado de la discapacidad intelectual es Ligero.

Las reglas obtenidas del algoritmo PRISM al ejecutar la consulta “Select * from prism (‘casos, ‘grm’)” son:

" if ei = No then grm = Moderado"

" if radiaciones = No Sabe and calor_mad_emb = No Sabe then grm = Moderado"

" if ei = Si then grm = Severo"

" if calor_mad_emb = No Sabe and radiaciones = Si and ei = No Sabe then grm = Severo"

" if ei = No Sabe and radiaciones = No then grm = Ligero"

" if ei = No Sabe and calor_mad_emb = No then grm = Ligero"

Estos resultados se pueden interpretar de la siguiente forma:

1. Si la madre no tuvo enfermedades infecciosas durante el embarazo entonces el grado de la discapacidad intelectual es Moderado.

2. Si no se sabe si la madre estuvo expuesta a radiaciones y calores durante el embarazo el grado de la discapacidad intelectual es Moderado.
3. Si la madre tuvo enfermedades infecciosas durante el embarazo entonces el grado de la discapacidad intelectual es Severo.
4. Si la madre no sabe si recibió calor durante el embarazo, ni si tuvo enfermedades infecciosas y estuvo expuesta a radiaciones durante el embarazo entonces el grado de la discapacidad intelectual es Severo.
5. Si la madre no sabe si tuvo enfermedades infecciosas y no estuvo expuesta a radiaciones durante el embarazo entonces el grado de la discapacidad intelectual es Ligero.
6. Si la madre no sabe si tuvo enfermedades infecciosas y no estuvo expuesta a calores durante el embarazo entonces el grado de la discapacidad intelectual es Ligero.

Las reglas obtenidas del algoritmo ID3 al ejecutar la consulta “Select * from id3 (‘casos, ‘grm’)” son:

"if ei = No sabe then grm = Moderado"

"if ei = Si then grm = Severo"

"if ei = No sabe and radiaciones = No then grm = Ligero"

"if ei = No sabe and radiaciones = No sabe then grm = Ligero "

"if ei = No sabe and radiaciones = Si then grm = Severo"

Estos resultados se pueden interpretar de la siguiente forma:

1. Si la madre no tuvo enfermedades infecciosas durante el embarazo entonces el grado de la discapacidad intelectual es Moderado.
2. Si la madre tuvo enfermedades infecciosas durante el embarazo entonces el grado de la discapacidad intelectual es Severo.
3. Si la madre no sabe si tuvo enfermedades infecciosas y estuvo expuesta a radiaciones durante el embarazo entonces el grado de la discapacidad intelectual es Severo.

4. Si la madre no sabe si tuvo enfermedades infecciosas y no estuvo expuesta a radiaciones durante el embarazo entonces el grado de la discapacidad intelectual es Ligero.
5. Si la madre no sabe si tuvo enfermedades infecciosas ni si estuvo expuesta a radiaciones durante el embarazo entonces el grado de la discapacidad intelectual es Moderado.

3.6 Evaluación e implementación.

El objetivo del negocio correspondiente al descubrimiento de patrones ocultos en los datos; que permite clasificar el grado de discapacidad intelectual, basado en las relaciones que se establecen entre los atributos seleccionados fue cumplido, por lo que pueden considerarse los modelos como aceptados, desde el punto de vista analítico, para apoyar la toma de decisiones administrativas del Centro de Genética Médica.

Para la implementación del proyecto los directivos del Centro de Genética Médica son los encargados de emprender acciones y determinar, si así lo estiman conveniente, una estrategia a seguir, basada en la información descubierta por los algoritmos. Además se generará un informe con todo el proceso de minería, que servirá de apoyo o consulta para el proceso administrativo y la toma de decisiones.

La integración de la extensión de minería de datos al sistema le permite a los especialistas de genética médica el desarrollo de una interfaz para llamar a las funciones de los algoritmos de minería de datos y ante cualquier actualización de los datos de la aplicación poder volver a realizar un análisis de la información para obtener conocimiento desde la aplicación de genética médica.

A continuación se realiza un experimento para validar los resultados obtenidos en esta investigación.

3.7 Diseño del experimento.

Un experimento se realiza generalmente con el objetivo de validar una hipótesis en la cual se busca establecer relaciones entre los diferentes factores, analizando si una o más variables independientes afectan a una o más variables dependientes y por qué lo hacen.

Roberto Hernández Sampieri[1] define como experimento a "un estudio de investigación en el que se manipulan deliberadamente una o más variables independientes (supuestas causas) para analizar las consecuencias que la manipulación tiene sobre una o más variables dependientes (supuestos efectos), dentro de una situación de control para el investigador". (Martínez Valenzuela, 2007)

El primer paso para diseñar un experimento es definir las variables independientes y las variables dependientes.

La variable independiente es aquella que identifica la causa en una relación entre variables y la dependiente expresa el efecto que puede causar una variación en la variable independiente.

En este caso se definieron como variables independientes si la tabla sobre la cual se va a realizar el análisis está particionado y el indexado o sea si se crearon índices sobre la tabla sobre la cual se va a realizar el análisis. Como variables dependientes se identificaron el tiempo de respuesta.

Luego de haber definido las variables se define la operacionalización o definición operacional que es especificar la manera en que se manipularán las variables independientes.

Para la manipulación de las variables *particionado* e *índices* se utilizará el nivel dos: presencia-ausencia de esta variable independiente.

El cuarto paso es definir cómo se van a medir las variables dependientes. En este caso la variable del *tiempo de respuesta* se va a medir por intervalos de tiempo en segundos.

Para un mejor entendimiento del diseño del experimento se resume en las tablas 2 y 3 la definición operacional.

Tabla 2: Operacionalización de las variables independientes.

Variable	Tipo de variable	Operacionalización	Categorías
Índices	Independiente	Si la tabla sobre la cual se va a realizar el	- Indexado - No Indexado

[1]**Dr. Roberto Hernández Sampieri** Director del Centro de Investigación de la Universidad de Celaya y profesor en el Instituto Politécnico Nacional

		análisis se le realizaron índices o no.	
Particionado	Independiente	Si la tabla sobre la cual se va a realizar el análisis está o no particionada.	- Particionado - No particionado

Tabla 3: Operacionalización de las variables dependientes.

Variable	Unidad de medida
Tiempo de respuesta	Intervalo de tiempo (segundos)

Después de haber realizado la Operacionalización de las variables se debe definir cómo se va a realizar el control o validez interna de la situación experimental para saber realmente si las variables independientes que nos interesan tienen o no efecto en las dependientes. Para lo cual se definieron 2 grupos de comparación equivalentes en todo, excepto en la manipulación de las variables independientes.

3.8 Aplicación del experimento.

Para el entorno del experimento se seleccionó una computadora Haier con un procesador Intel Celeron 2000 MHz, una memoria RAM de 1 Gb y un disco duro de 120 Gb. Además se cuenta con el servidor de PostgreSQL 9.1, la herramienta PgAdminIII y como instrumento de medición el EXPLAIN ANALYZE.

3.8.1. Validación del mecanismo de particionado.

Para validar que el particionado de tabla propuesto en el epígrafe 2.4.1 mejora el rendimiento de los algoritmos se creó el “Grupo A” que al cual no se le va aplicar el particionado y el “Grupo B” donde se van a crear particiones. En los dos casos los grupos cuentan con **576 000** registros y para lograr la equivalencia en los dos grupos en el momento del análisis se van a chequear que estén corriendo los mismos procesos.

Como resultado del particionado en la tabla *casosb* se obtienen 4 tablas: la mastercasosb que es la tabla padre, la casosbLigero que contiene todos los registros clasificados con valor “Ligero”, la casosbModerado con los registros con valores Moderado y la casosbSevero con los registros con valores Severo.

Para medir los tiempos de respuesta se utilizó la sentencia “explain analyze” y se analizaron 3 veces los tiempos de respuestas para cada grupo realizando luego un promedio de los tiempos obtenidos.

Al aplicar el algoritmo 1R integrado al SGBD PostgreSQL sin crear particiones el tiempo de respuesta es de aproximadamente de 26,83 segundos y luego de haber particionado es de 18,28 segundos. (Ver anexo 6).

En la figura 15 se observa cómo el análisis en la tabla particionado por el valor de la clase es menor que en la tabla normal quedando demostrado que el mecanismo de particionado de datos agiliza el resultado del algoritmo 1R.

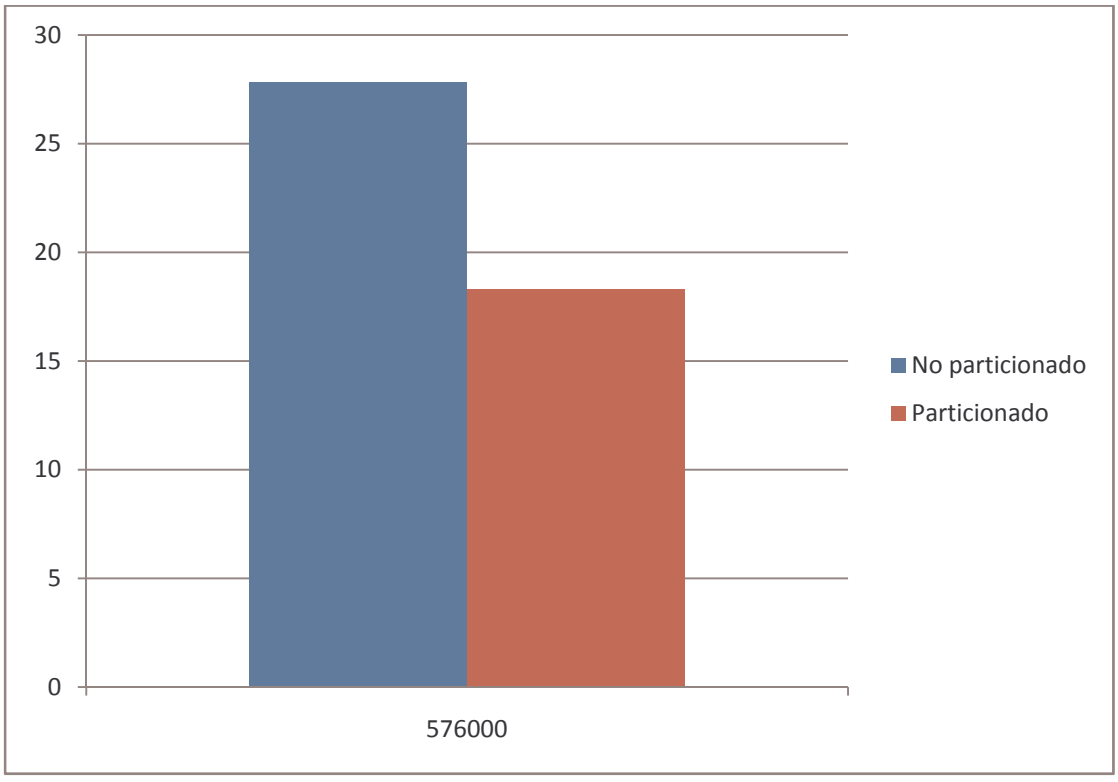

Figura 15: Comparación de los resultados de los análisis en tablas particionada y sin particionar.

3.8.2. Validación del mecanismo de indexado.

Para la aplicación de la validación se crearon dos grupos el “Grupo A” al cual no se le va a crear índices y el “Grupo C” donde se van a crear índices parciales para cada valor de la clase. En los dos casos los grupos cuentan con **576 000** registros y para lograr la equivalencia en los dos grupos en el momento del análisis se van a chequear que estén corriendo los mismos procesos.

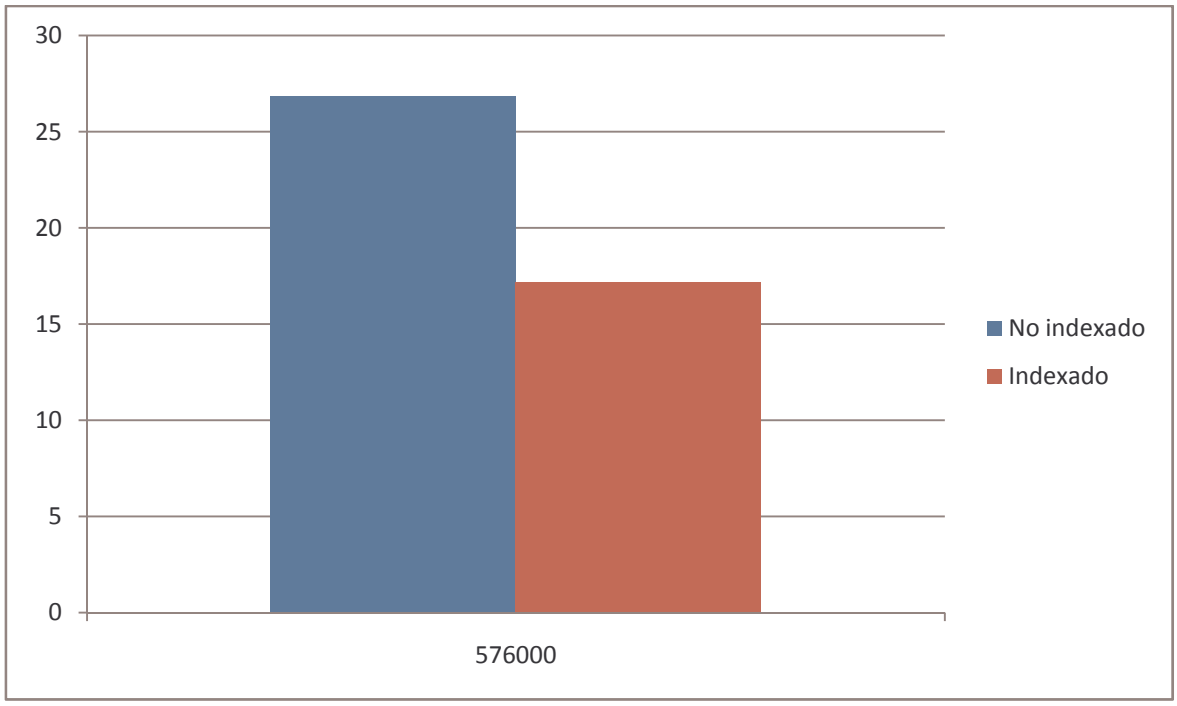

Figura 16: Comparación de los resultados de los análisis en tablas con índices y sin índices creados.

Al realizar el análisis a las tablas sin crear índices los resultados se obtuvieron en 26,83 segundos y al crear los índices parciales sobre la clase el tiempo de respuesta disminuyó a 17,203 segundos. Lo anterior demuestra que al crear los índices en las tablas los resultados se obtienen con mayor rapidez, lo que evidencia que los algoritmos integrados a PostgreSQL permiten aprovechar las potencialidades del SGBD.

Conclusiones:

En el presente capítulo se realizó la aplicación del proceso de minería de datos a la base de datos de la aplicación de genética médica, donde se analizaron los campos importantes para darle cumplimiento a los objetivos de negocio y minería de datos propuestos. Para la integración de la información

se realizó una tabla como resultado de una vista a la cual se le aplicaron los algoritmos, obteniendo como resultado un conjunto de reglas que permiten clasificar el grado de la discapacidad intelectual. Además se realizó la validación de los algoritmos integrados al SGBD PostgreSQL para lo cual se diseñó un experimento, con el cual se demostró que la integración de los algoritmos a PostgreSQL permite aprovechar las potencialidades del gestor para disminuir los tiempos de análisis de los datos a través de los mecanismos implementados.

CONCLUSIONES GENERALES

De los resultados obtenidos en esta investigación se puede decir que: las técnicas de minería de datos árboles de decisión y reglas de inducción permiten obtener como resultado final reglas que por sus características son unas de las formas de representar más divulgadas y unas de la más compresivas de entender por las personas.

Además se pudo evidenciar que las herramientas libres existentes de minería de datos tienen el inconveniente de ser independientes del SGBD por lo cual se implementaron tres algoritmos de las técnicas de clasificación y se integraron al SGBD PostgreSQL a través de la creación de una extensión, lo que contribuye a la soberanía tecnológica del país y a que el gestor sea más competitivo.

En la investigación también se desarrollaron funciones que permiten aprovechar los mecanismos de optimización del gestor para mejorar los resultados de respuesta del los algoritmos implementados.

Se realizó la validación de los algoritmos implementados utilizando un diseño de experimento en el cual se observó que la integración de los algoritmos a PostgreSQL permite aprovechar las potencialidades del gestor para disminuir los tiempos de análisis de los datos a través de los mecanismos implementados.

RECOMENDACIONES

Integrar otros algoritmos de minería de datos al SGBD PostgreSQL de la técnica de Reglas de Asociación, pues es tan descriptiva como las utilizadas en la investigación y está entre las más utilizadas.

REFERENCIA BIBLIOGRÁFICA

Bertenshaw1, G. P., & Yip1, P. (2010). Multianalyte Profiling of Serum Antigens and Autoimmune and Infectious Disease Molecules to Identify Biomarkers Dysregulated in Epithelial Ovarian Cancer . *Revista Cancer Epidemiology, Biomarkers & Prevention* .

Brito Sarasa, R. (2008). Minería de Datos aplicada a la Gestión Docente del Instituto Superior Politécnico José Antonio Echeverría. *Tesis para optar por la Maestría en Informática Aplicada.*

Chesñevar, C. I. (2009). *Datamining y aprendizaje automatizado.* Obtenido de http://cs.uns.edu.ar/~cic/dm2009/downloads/transparencias/05_dm%20(Learning_rules).pdf

Domínguez-Rodríguez, K. M., & Téllez-Sánchez, L. (septiembre 2011). Sistema de apoyo a la toma de decisiones en el proceso de negociación comercial. *Revista Ciencias Holguín* .

Filiberto, Y., Bello, R., & Caballero, Y. (Oct. 2011). ALGORITMO PARA EL APRENDIZAJE DE REGLAS DE CLASIFICACION BASADO EN LA TEORÍA DE LOS CONJUNTOS APROXIMADOS EXTENDIDA. *Revista DYNA* , vol.78 no.169 Medellín.

Gasparovica, M., & Aleksejeva, L. (2010). Using Fuzzy Algorithms for Modular Rules. *Scientific Journal of Riga Technical University Computer Science. Information Technology and Management Science* , 94-98.

Haberstroh, R. (2008). *Oracle Data Mining Tutorial.*

Heughes Escobar, V. (2007). Minería Web de Uso y perfiles de Usuario:Aplicaciones con Lógica Difusa. *Tesis de Doctorado* .

HOLTE, R. C. (1993). Very Simple Classification Rules Perform Well on Most Commonly Used Datasets. *Machine Learning* .

KDnuggets Polls. Data mining methods. (marzo de 2007). Recuperado el 12 de septiembre de 2012, de http://www.kdnuggets.com/polls/2007/data_mining_methods.htm

Landín Sorí, M., & Romero Sánchez, R. (2012). Árboles de decisiones para el diagnóstico y tratamiento en pacientes con glaucoma neovascular. *Revista Archivo Médico de Camaguey* .

MacLennan, J., Tang, Z., & Crivat, B. (2009). *Data Mining with Microsoft SQL Server 2008.* Wiley Publishing, Inc., Indianapolis, Indiana.

Marcheco Teruel, D. C. (2008). Genética comunitaria: la principal prioridad para la genética médica en Cuba. *Revista Cubana Genetica Comunitaria* , 3-4.

Marrero Hernández, F. M. (2009). Desarrollo de la aplicación informática a utilizar en el Estudio Integral de las personas con discapacidad en Venezuela, misión Dr. José Gregorio Hernández. *Trabajo de Diploma.*

Martínez Valenzuela, V. (octubre de 2007). *Diseño Experimental*. Obtenido de http://www.slideshare.net/hayimemaishte/diseo-experimental

Martins, A. C. (2009). Estudo Comparativo de Três Algoritmos de Machine Learning Dados Electrocardiográficos. Recuperado el 2012, de http://www.dcc.fc.up.pt/~ines/aulas/0910/MIM/trabs_ano_anterior/nonam e.pdf

Moine, J. M., Haedo, A., & Gordillo, S. (2011). Estudio comparativo de metodologías para minería de datos. *Evento: XIII Workshop de Investigadores en Ciencias de la Computación* (págs. p. 278-281). RedUNCI .

Molina López, J. M., & García Herrero, J. *TÉCNICAS DE ANÁLISIS DE DATOS.*

Omar Ruiz, S. B. (2008). APLICACIÓN DE MINERÍA DE DATOS PARA DETECCIÓN DE PATRONES EN INVESTIGACIONES BIOTECNOLÓGICAS. *Revista ESPOL Ciencia* .

PostgreSQL. (12 de SEPTIEMBRE de 2011). Obtenido de http://www.postgresql.org/about/press/presskit91/es/

Robinson, C. (2011). *Basic introduction into pgAdmin III and SQL queries.* Obtenido de http://library.thehumanjourney.net/658/

Robles Aranda, Y. (2012). Propuesta de integración de las técnicas de minería de datos Árboles de decisión y Reglas de inducción al Sistema Gestor de Bases de Datos . *UCIENCIA.* La Habana.

Rodríguez Suárez, Y. .. (2009). Herramientas de Minería de Datos. *Revista Cubana de Ciencias Informáticas , 3*.

Sánchez Ramírez, H., & Peña Villagra, B. (diciembre, 2011). Modelos de Data Mining Asociados al Fraude. La Experiencia Chilena. *Revista de Administración Tributaria CIAT/AEAT/IEF* , 117-134.

Solarte Martínez, G. R., & Ocampo S., C. A. (s.f.). *ESTADO DEL ARTE EN LA UTILIZACIÓN DE TECNICAS AVANZADAS PARA LA BUSQUEDA DE INFORMACIÓN NO TRIVIAL A PARTIR DE DATOS.* Recuperado el 1 de diciembre de 2011, de http://www.lenhs.ct.ufpb.br/html/downloads/serea/trabalhos/A15_15.pdf.

Solarte Martínez, G. R., & Ocampo S., C. A. (2009). Técnicas de clasificación y análisis de representación del conocimiento para problemas de diagnóstico. *ScientiaEtTechnica* .

Solarte Martínez, G. R., & Soto Mejía, J. A. (2011). Árboles de decisiones en el diagnóstico de enfermedades cardiovasculares. *Revista Scientia et Technica* .

Soto Jaramillo, C. M. (2009). INCORPORACIÓN DE TÉCNICAS MULTIVARIANTES EN UN SISTEMA GESTOR DE BASES DE DATOS. *Tesis de maestría* .

The PostgreSQL Global Development Group. (2011). *PostgreSQL 9.1.0 Documentation.*

U Fayyad, G. P.-S. (1996). *Data Mining and Knowledge Discovery in Databases: An overview, Communications of ACM.* Obtenido de http://dl.acm.org/citation.cfm?id=240464

V.Ramesh, P. P. (agosto de 2011). *Performance Analysis of Data Mining Techniques for Placement Chance Prediction.* Recuperado el diciembre de 2011, de http://www.ijser.org/researchpaper%5CPerformance-Analysis-of-Data-Mining-Techniques-for-Placement-Chance-Prediction.pdf.

Vázquez Ortíz, Y., & Castillo Martínez, G. (2011). Propuesta de un plan de capacitación para la preparación y futura certificación en PostgreSQL. *Revista Cubana de Ciencias Informáticas* .

Vega Riverón, B., & Sánchez Valdés, L. (2012). Clasificación de dengue hemorrágico utilizando árboles de decisión en la fase temprana de la enfermedad . *Revista Cubana de Medicina Tropical* .

BIBLIOGRAFÍA

Bertenshaw1, G. P., & Yip1, P. (2010). Multianalyte Profiling of Serum Antigens and Autoimmune and Infectious Disease Molecules to Identify Biomarkers Dysregulated in Epithelial Ovarian Cancer . *Revista Cancer Epidemiology, Biomarkers & Prevention* .

Brito Sarasa, R. (2008). Minería de Datos aplicada a la Gestión Docente del Instituto Superior Politécnico José Antonio Echeverría. *Tesis para optar por la Maestría en Informática Aplicada.*

Cao, L., Yu, P. S., Zhang, C., & Zhang, H. (2007). *Data Mining for Business Applications.*

Chesñevar, C. I. (2009). *Datamining y aprendizaje automatizado.* Obtenido de http://cs.uns.edu.ar/~cic/dm2009/downloads/transparencias/05_dm%20(Learning_rules).pdf

Chiu, S., & Tavella, D. (2008). *Data Mining and Market Intelligence for Optimal Marketing Returns.*

Domínguez-Rodríguez, K. M., & Téllez-Sánchez, L. (septiembre 2011). Sistema de apoyo a la toma de decisiones en el proceso de negociación comercial. *Revista Ciencias Holguín* .

Filiberto, Y., Bello, R., & Caballero, Y. (Oct. 2011). ALGORITMO PARA EL APRENDIZAJE DE REGLAS DE CLASIFICACION BASADO EN LA TEORÍA DE LOS CONJUNTOS APROXIMADOS EXTENDIDA. *Revista DYNA* , vol.78 no.169 Medellín.

Frawley, W., & Piatetsky-Shapiro, G. (1992). *Knowledge discovery in databases: An overview.* Obtenido de https://www.aaai.org/ojs/index.php/aimagazine/article/viewArticle/1011

Gasparovica, M., & Aleksejeva, L. (2010). Using Fuzzy Algorithms for Modular Rules. *Scientific Journal of Riga Technical University Computer Science. Information Technology and Management Science* , 94-98.

Gorunescu, F. (2011). *Data Mining Concepts Models and Techniques.*

Guillén-Navarro, E., Ballesta-Martínez, M., & López-González, V. (2011). Genética y enfermedad. Concepto de genética médica. *Revista Nefrología. Órgano Oficial de la Sociedad Española de Nefrología* , 3-10.

Haberstroh, R. (2008). *Oracle Data Mining Tutorial.*

Han, J., & Kamber, M. (2007). *Data Mining: Concepts and Techniques.* Asma Stephan.

Heughes Escobar, V. (2007). Minería Web de Uso y perfiles de Usuario:Aplicaciones con Lógica Difusa. *Tesis de Doctorado* .

HOLTE, R. C. (1993). Very Simple Classification Rules Perform Well on Most Commonly Used Datasets. *Machine Learning* .

KDnuggets Polls. Data mining methods. (marzo de 2007). Recuperado el 12 de septiembre de 2012, de http://www.kdnuggets.com/polls/2007/data_mining_methods.htm

Landín Sorí, M., & Romero Sánchez, R. (2012). Árboles de decisiones para el diagnóstico y tratamiento en pacientes con glaucoma neovascular. *Revista Archivo Médico de Camaguey* .

MacLennan, J., Tang, Z., & Crivat, B. (2009). *Data Mining with Microsoft SQL Server 2008.* Wiley Publishing, Inc., Indianapolis, Indiana.

Marcheco Teruel, D. C. (2008). Genética comunitaria: la principal prioridad para la genética médica en Cuba. *Revista Cubana Genetica Comunitaria* , 3-4.

Marrero Hernández, F. M. (2009). Desarrollo de la aplicación informática a utilizar en el Estudio Integral de las personas con discapacidad en Venezuela, misión Dr. José Gregorio Hernández. *Trabajo de Diploma.*

Martínez Valenzuela, V. (octubre de 2007). *Diseño Experimental*. Obtenido de http://www.slideshare.net/hayimemaishte/diseo-experimental

Martins, A. C. (2009). Estudo Comparativo de Três Algoritmos de Machine Learning Dados Electrocardiográficos. Recuperado el 2012, de http://www.dcc.fc.up.pt/~ines/aulas/0910/MIM/trabs_ano_anterior/nonam e.pdf

Moine, J. M., Haedo, A., & Gordillo, S. (2011). Estudio comparativo de metodologías para minería de datos. *Evento: XIII Workshop de Investigadores en Ciencias de la Computación* (págs. p. 278-281). RedUNCI .

Molina López, J. M., & García Herrero, J. *TÉCNICAS DE ANÁLISIS DE DATOS.*

Olson, D. L., & Delen, D. (2008). *Advanced Data Mining Techniques.*

Omar Ruiz, S. B. (2008). APLICACIÓN DE MINERÍA DE DATOS PARA DETECCIÓN DE PATRONES EN INVESTIGACIONES BIOTECNOLÓGICAS. *Revista ESPOL Ciencia* .

Poncelet, P., Teisseire, M., & Masseglia, F. (2007). *Data Mining Patterns: New Methods and Applications.*

PostgreSQL. (12 de SEPTIEMBRE de 2011). Obtenido de http://www.postgresql.org/about/press/presskit91/es/

Robinson, C. (2011). *Basic introduction into pgAdmin III and SQL queries.* Obtenido de http://library.thehumanjourney.net/658/

Robles Aranda, Y. (2012). Propuesta de integración de las técnicas de minería de datos Árboles de decisión y Reglas de inducción al Sistema Gestor de Bases de Datos . *UCIENCIA.* Ciudad de la Habana.

Rodríguez Suárez, Y. .. (2009). Herramientas de Minería de Datos. *Revista Cubana de Ciencias Informáticas , 3*.

Sánchez Ramírez, H., & Peña Villagra, B. (diciembre, 2011). Modelos de Data Mining Asociados al Fraude. La Experiencia Chilena. *Revista de Administración Tributaria CIAT/AEAT/IEF* , 117-134.

Solarte Martínez, G. R., & Ocampo S., C. A. (s.f.). *ESTADO DEL ARTE EN LA UTILIZACIÓN DE TECNICAS AVANZADAS PARA LA BUSQUEDA DE INFORMACIÓN NO TRIVIAL A PARTIR DE DATOS.* Recuperado el 1 de diciembre de 2011, de http://www.lenhs.ct.ufpb.br/html/downloads/serea/trabalhos/A15_15.pdf.

Solarte Martínez, G. R., & Ocampo S., C. A. (2009). Técnicas de clasificación y análisis de representación del conocimiento para problemas de diagnóstico. *ScientiaEtTechnica* .

Solarte Martínez, G. R., & Soto Mejía, J. A. (2011). Árboles de decisiones en el diagnóstico de enfermedades cardiovasculares. *Revista Scientia et Technica* .

Soto Jaramillo, C. M. (2009). INCORPORACIÓN DE TÉCNICAS MULTIVARIANTES EN UN SISTEMA GESTOR DE BASES DE DATOS. *Tesis de maestría* .

Taniar, D. (2008). *ebook Data Mining and Knowledge Discovery Technologies Advances in Data Warehousing and Mining.*

The PostgreSQL Global Development Group. (2011). *PostgreSQL 9.1.0 Documentation.*

U Fayyad, G. P.-S. (1996). *Data Mining and Knowledge Discovery in Databases: An overview, Communications of ACM.* Obtenido de http://dl.acm.org/citation.cfm?id=240464

V.Ramesh, P. P. (agosto de 2011). *Performance Analysis of Data Mining Techniques for Placement Chance Prediction.* Recuperado el diciembre de 2011, de http://www.ijser.org/researchpaper%5CPerformance-Analysis-of-Data-Mining-Techniques-for-Placement-Chance-Prediction.pdf.

Vázquez Ortíz, Y., & Castillo Martínez, G. (2011). Propuesta de un plan de capacitación para la preparación y futura certificación en PostgreSQL. *Revista Cubana de Ciencias Informáticas* .

Vazquez Ortíz, Y., Mesa Reyes, Y., & Castillo Martínez, G. (2012). COMUNIDAD TÉCNICA CUBANA DE POSTGRESQL: ARMA PARA LA MIGRACIÓN DEL PAÍS A TECNOLOGÍAS DE BASES DE DATOS DE CÓDIGO. *UCIENCIA.*

Vega Riverón, B., & Sánchez Valdés, L. (2012). Clasificación de dengue hemorrágico utilizando árboles de decisión en la fase temprana de la enfermedad . *Revista Cubana de Medicina Tropical* .

ANEXOS

Anexo 1: Costos de las herramientas para aplicar el proceso de minería de datos.

Tabla 4: Costo de la herramienta MATLAB en euro por cada puesto de trabajo.

Herramienta Matlab	Precio
Sólo el intérprete, las funciones básicas y la interfaz gráfica.	$24,00.00
Statistics toolkit. Sin esto no tenemos funciones estadísticas	$1,000.00
Signal processing toolkit.	$1,000.00

Tabla 5: Costo de la herramienta SPSS en euros por cada puesto de trabajo.

Herramienta	Precio
IBM SPSS Modeler Client Professional Authorized User License + SW Subscription & Support 12 Months (D0EMZLL)	24,432.00
IBM SPSS Modeler Professional Authorized User Initial Fixed Term License + SW Subscription & Support 12 Months (D0EC5LL)	10,701.00
IBM SPSS Modeler Premium Concurrent User License + SW Subscription & Support 12 Months (D0EPGLL)	97,540.00

Tabla 6: Costo de la herramienta SAS Enterprise Miner en euros por cada puesto de trabajo.

Herramienta SAS Enterprise Miner	Precio
SAS Enterprise Miner anual	150.000

Oracle Database			
	Software Update License & Support	Processor License	Software Update License & Support
Database Products			
Oracle Database			
Standard Edition One	39.60	5,800	1,276.00
Standard Edition	77.00	17,500	3,850.00
Enterprise Edition	209.00	47,500	10,450.00
Personal Edition	101.20	-	-
Mobile Server	-	23,000	5,060.00
NoSQL Database Enterprise Edition	44	10,000	2,200.00
Enterprise Edition Options:			
Real Application Clusters	101.20	23,000	5,060.00
Real Application Clusters One Node	44.00	10,000	2,200.00
Active Data Guard	44.00	10,000	2,200.00
Partitioning	50.60	11,500	2,530.00
Real Application Testing	50.60	11,500	2,530.00
Advanced Compression	50.60	11,500	2,530.00
Advanced Security	50.60	11,500	2,530.00
Label Security	50.60	11,500	2,530.00
Database Vault	101.20	23,000	5,060.00
OLAP	101.20	23,000	5,060.00
Advanced Analytics	101.20	23,000	5,060.00
Spatial	77.00	17,500	3,850.00
In-Memory Database Cache	101.20	23,000	5,060.00
Retail Data Model	176.00	40,000	8,800.00
Communications Data Model	176.00	40,000	8,800.00
Airlines Data Model	176.00	40,000	8,800.00

Figura 17: Precio de Oracle en USD.

Tabla 7: Precio de SQL Server en USD.

SQL Server	Precio
Enterprise Edition Base License (includes 1-year support)	$34,369
Management Tools Option	Included
Security Option	Included
Compression Option	Included
Spatial Option	Included
Replication Option	Included
Total Cost	$34,369

Anexo 2: Breve descripción de las técnicas de minería de datos implementadas en Oracle Data Mining (Haberstroh, 2008).

Algorithm	Model Details
Apriori (association rules)	Association rules and frequent itemsets
Decision Tree	The full model with its content and rules
Generalized Linear Models	Attribute-level coefficient and statistics from `GET_MODEL_DETAILS_GLM` and global model information from `GET_MODEL_DETAILS_GLOBAL`
k-Means	For each cluster: statistics and hierarchy information, centroid, attribute histograms, and rules
MDL (attribute importance)	Ranked importance of each attribute
Naive Bayes	Conditional probabilities and priors
Non-Negative Matrix Factorization	Coefficients
O-Cluster	For each cluster: statistics and hierarchy information, centroid, attribute histograms, and rules
Support Vector Machine	Coefficients for linear models

Figura 18: Breve descripción de las técnicas de minería de datos implementadas en Oracle Data Mining (Haberstroh, 2008).

Anexo 3: Técnicas de minería de datos más empleadas (KDnuggets Polls. Data mining methods, 2007).

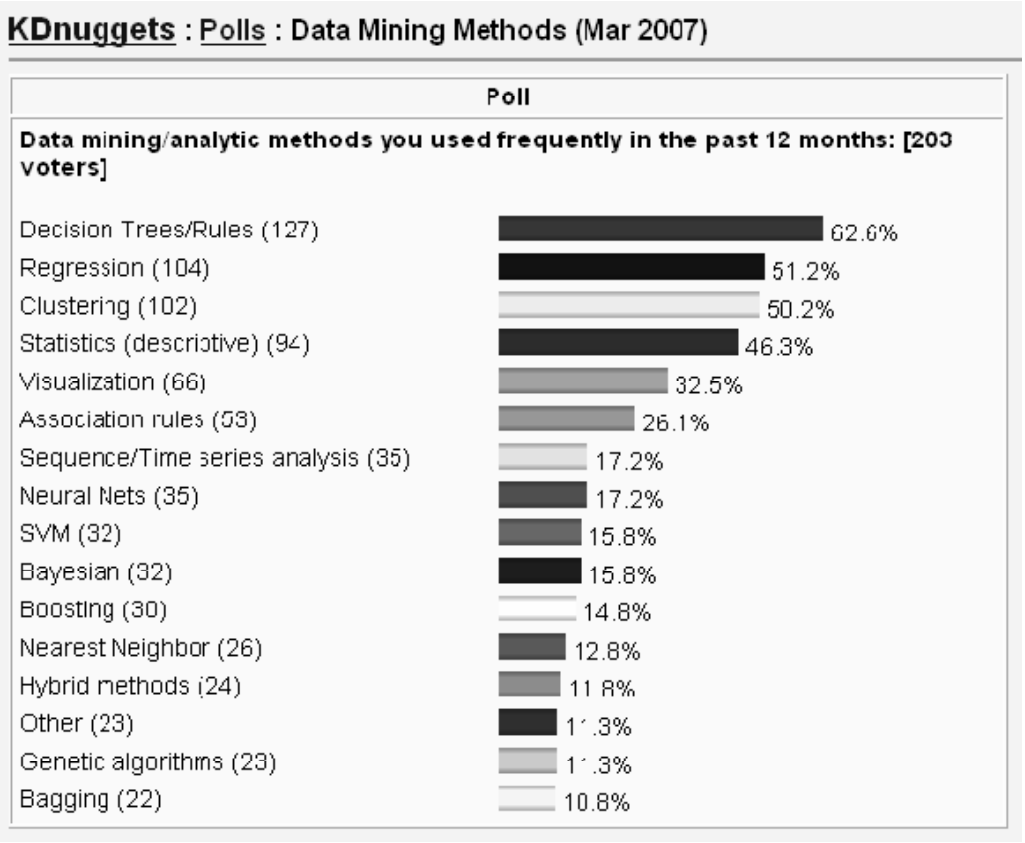

Figura 19: Técnicas de minería de datos más empleadas (KDnuggets Polls. Data mining methods, 2007).

Anexo 4: Tiempo de respuesta de los análisis realizados mediante los algoritmos integrados al SGBD PostgreSQL en el capítulo 3.

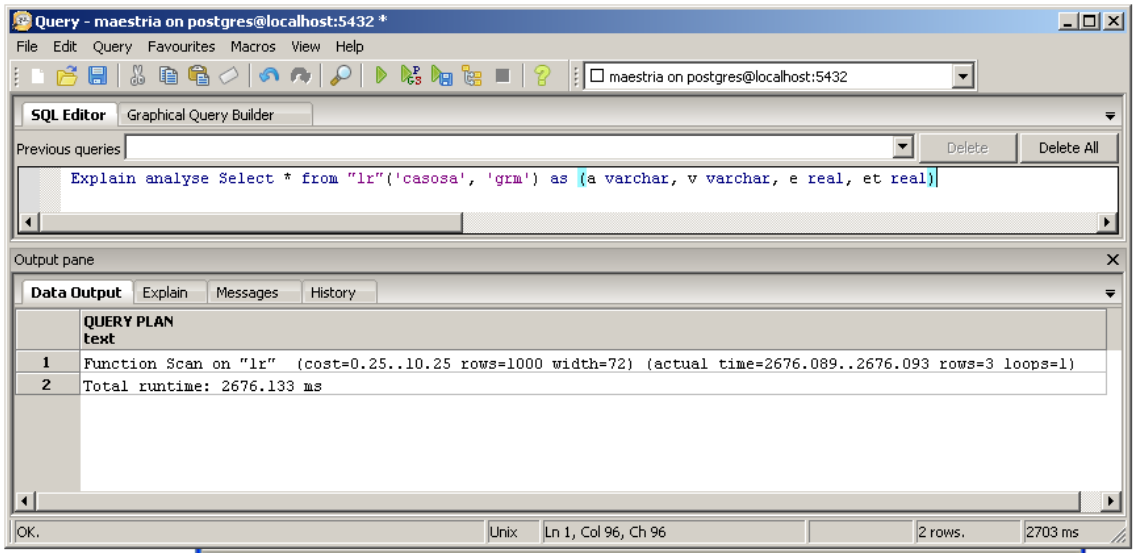

Figura 20: Tiempo de respuesta del análisis realizado mediante el algoritmo 1R integrados a PostgreSQL con 57600 registros.

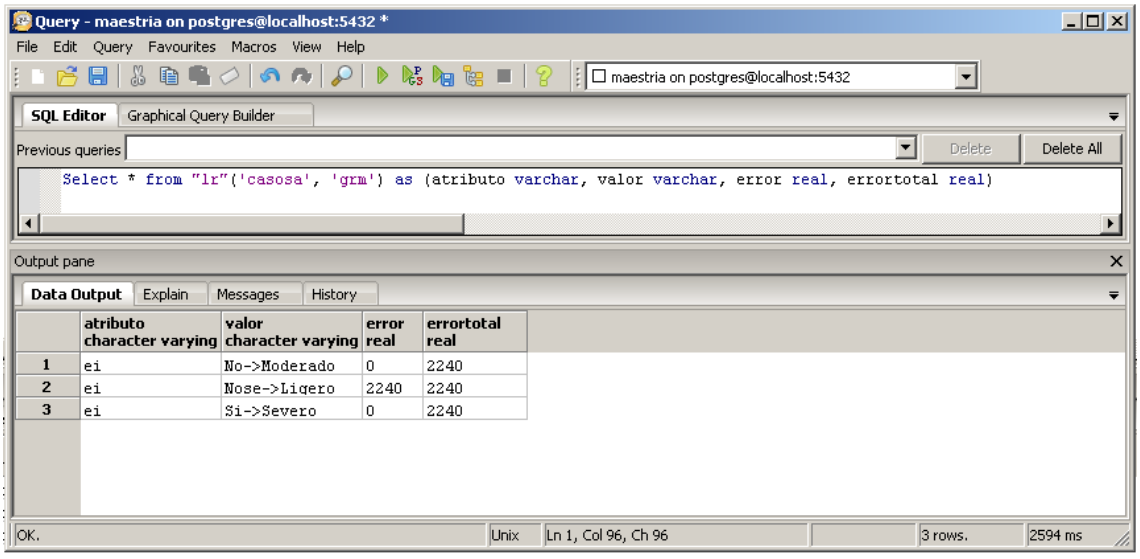

Figura 21: Resultado del análisis realizado mediante el algoritmo 1R integrados a PostgreSQL con 57600 registros.

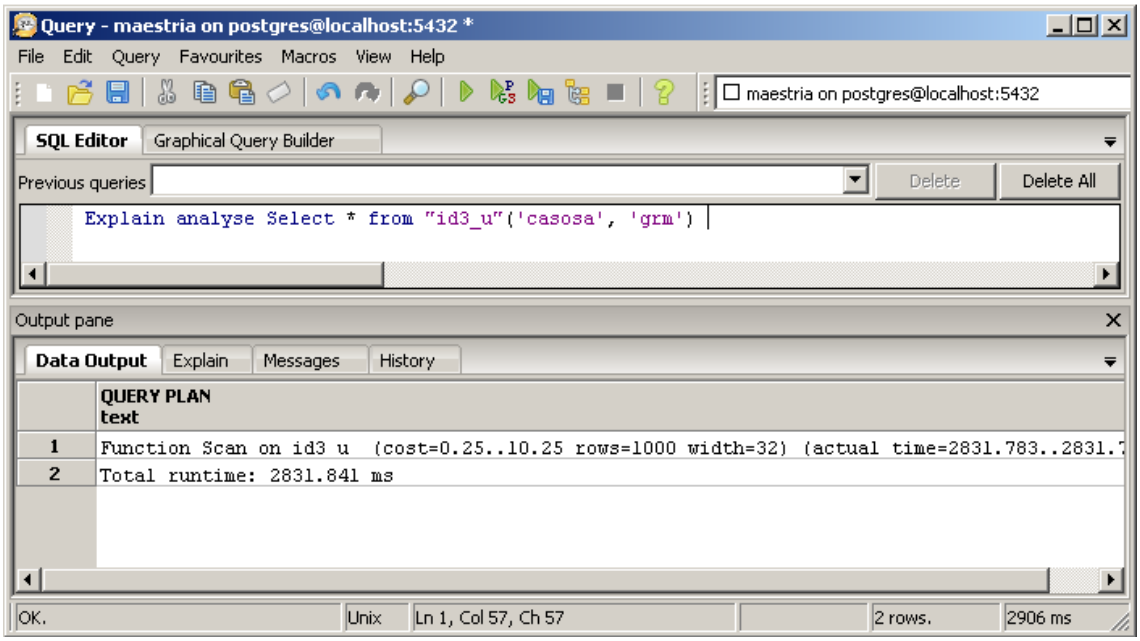

Figura 22: Tiempo de respuesta del análisis realizado mediante el algoritmo ID3 integrados a PostgreSQL con 57600 registros.

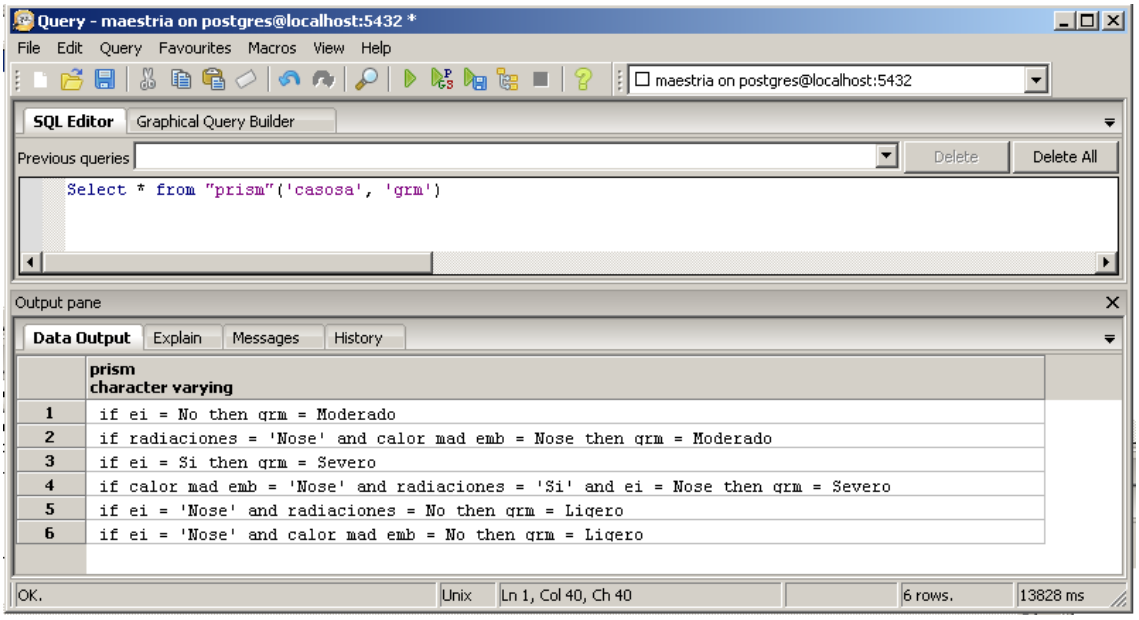

Figura 23: Resultado del análisis realizado mediante el algoritmo PRISM integrados a PostgreSQL con 500010 registros.

Anexo 5: Análisis de los tiempos de respuesta utilizando el particionado de datos.

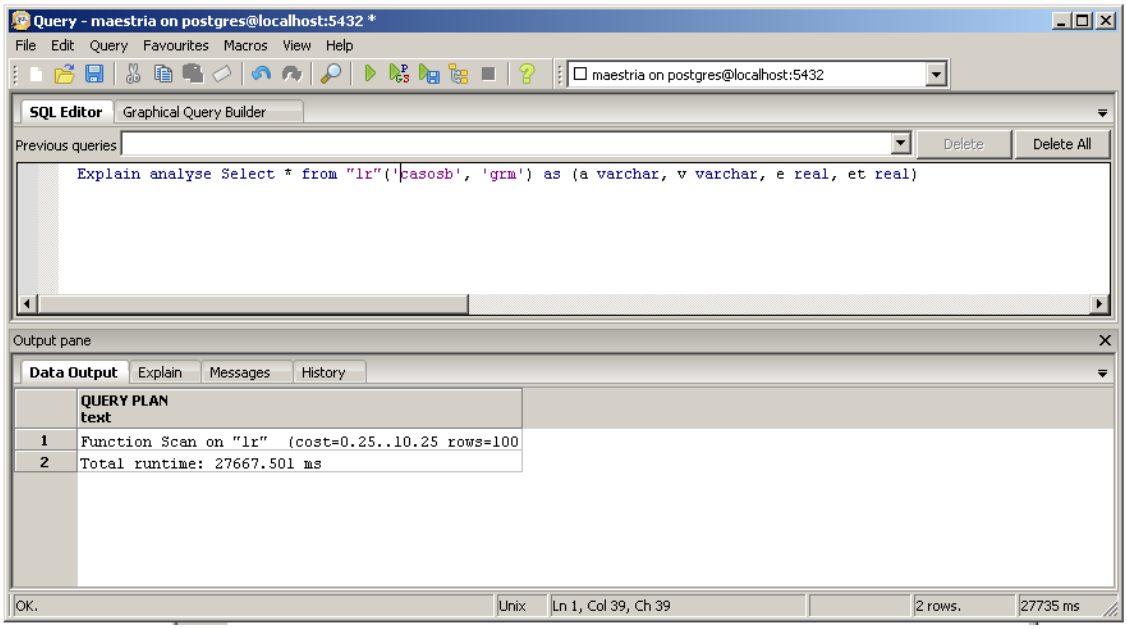

Figura 24: Tiempo de respuesta del algoritmo 1R aplicado a una tabla sin particiones.

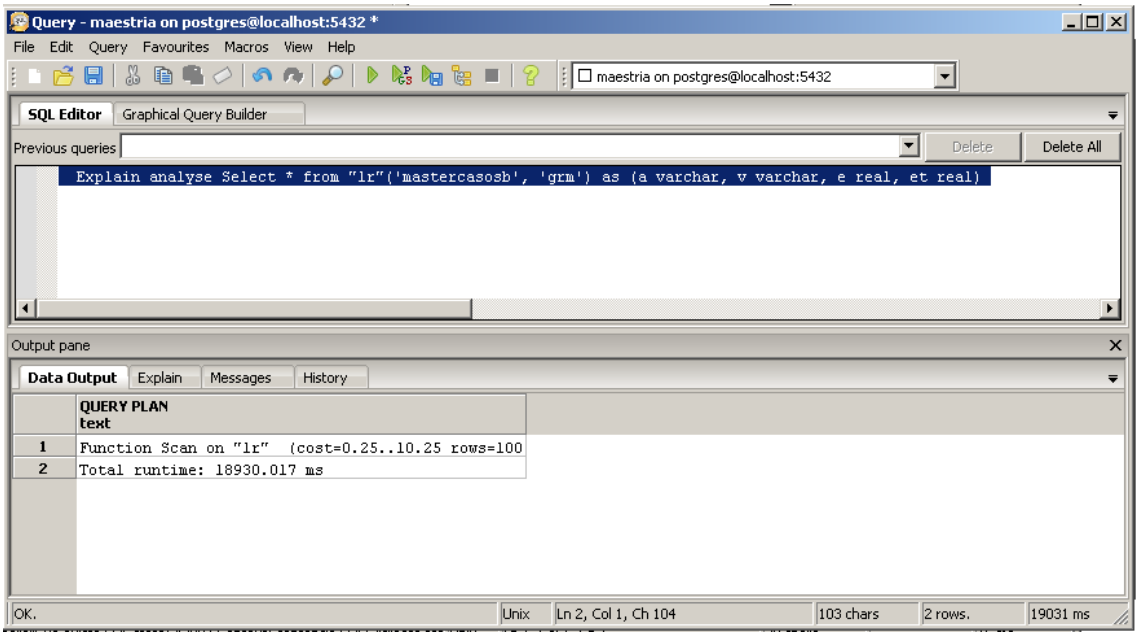

Figura 25: Tiempo de respuesta del algoritmo 1R aplicado a una tabla particionada.