TESIS DE LICENCIATURA

"DESARROLLO DE UN LENGUAJE DE ESPECIFICACION ORIENTADO A HACER DATA MINING BASADO EN CONOCIMIENTO"

Autor: ALICIA CASTIGLIEGO

Director de Tesis: Lic. OSVALDO GONZALEZ

ABRIL 1999.-

INDICE

RESUMEN

Se estudia la explotación de datos en grandes bases de datos, la extracción de información útil que no es posible obtener con las herramientas convencionales de consultas. Estos procesos son conocidos con los nombres de Knowledge Discovery Data (KDD) y Data Mining (DM).

La mayoría de las herramientas de DM existentes hoy, no están orientadas a un dominio particular, por lo cual sólo son capaces de resolver problemas específicos desde un proceso general, con la consiguiente pérdida del análisis en particular de cada dominio que este procedimiento implica. Tampoco contemplan las etapas previas al proceso de data mining (preparación de los datos), suponiendo que ésto fue realizado en otro ámbito.

Como consecuencia de esta realidad, en esta tesis se plantea la construcción en lenguaje de diseño de un generador de ambientes de KDD parametrizable a los efectos de poder actuar sobre diferentes dominios con la correspondiente especificidad que cada caso requiera.

También se estudian las posibles metodologías de trabajo que permitan guiar el proceso previo de preparación y limpieza de los datos. En este punto se ofrece el diseño de una herramienta que permitirá al usuario la elección de las etapas y del orden en que se ejecutarán para cumplimentar con esta faz previa al proceso de extracción de información.

Finalmente se trabajará sobre ejemplos típicos, investigando sus problemas asociados, al mismo tiempo que se plantearán posibles soluciones.

ABSTRACT

This thesis presents a research about the knowledge discovery data process and the data mining methods.

Most of data mining tools that exist nowadays are formulated in general terms. Given a database to be exploited, there is no way to establish how "dirty" or "clean" it is, i.e. which attributes are relevant and which ones are not. Consequently, their technological result is ineffectual to solve problems of specific domains. On the other hand they don't induce a working methodology, and as a result, a great deal of the KDD process must be carried out manually.

As a consequence of this reality, the construction of an environment generator that will carry out the KDD steps will be proposed, starting with the recognition and apprehension of the domain where the data mining is required.

What will be proposed here is not the development of an automatic KDD tool, but rather an environment where, by means of a powerful language (the KDD Language), domain relations will be specified so that they will be able to guide the behaviour of algorithms in general.

Finally, typical examples together with their associated problems and probable solutions will be showed.

1.- INTRODUCCION

Las herramientas de data mining que se comercializan en la actualidad están enfocadas al trabajo directo sobre las tablas y bases de datos, lo que implica su imposibilidad de uso en áreas ajenas a la organización de los sistemas de computación.

Esto constituye una limitación importante a los efectos de poder incorporar al estudio, conocimiento que el experto en el dominio de trabajo pueda tener como producto de su experiencia diaria.

Con el objetivo de superar esta falencia, se propone mudar el enfoque del trabajo, trasladándolo al nivel del conocimiento, definido por Newell en su primer mensaje como presidente de la AAAI, de la siguiente manera:

"Existe un nivel de conocimiento dentro de los niveles o capas que caracterizan a un sistema de computación inmediatamente por encima del nivel simbólico o de programa. El análisis del nivel del conocimiento debe focalizarse en la clase de información sobre el "mundo" que permite expresar un esquema de representación y su contenido de información, al margen de las estructuras de datos y operaciones implementadas (nivel simbólico). En este nivel de análisis, las nociones de estructuras de datos y algoritmos son irrelevantes y lo que importa es lo que el conjunto de sentencias es capaz de decir acerca del mundo."

Es por ello que en esta tesis se plantea el desarrollo de un lenguaje de especificación orientado a hacer data mining basado en conocimiento. Con el uso de dicho lenguaje se le permitirá al usuario, no conocedor de sistemas, la posibilidad de expresar todo su universo de trabajo, elegir una secuencia apropiada de tareas preparatorias de los datos y llegar a la aplicación de los algoritmos de data mining con la parametrización adecuada de manera tal de poder ejecutarlos inteligentemente.

En el capítulo 2.- se estudian las diferentes definiciones existentes sobre los procesos de knowledge discovery data y data mining. Se muestran posibles etapas en la preparación de los datos y los diferentes métodos de data mining.

En el capítulo 3.- se describen los problemas asociados a los modelos tradicionales, haciendo hincapié en el enfoque general que presentan hoy los métodos de data mining y la inexistencia de herramientas capaces de realizar, previo a la ejecución de los algoritmos específicos, todas las tareas de preparación de los datos de manera conjunta y sistemática.

En el capítulo siguiente se toman cada uno de los problemas descriptos y se plantean posibles soluciones para cada caso. Concretándose en el capítulo 5, la presentación del KDD Language, un lenguaje de especificación de dominios, con sus tres componentes básicas: de definición del dominio, de sistemas y de descripción de acciones.

El capítulo 6.- está dedicado al ambiente de parametrización o interfaz con el usuario. Se propone una estructura de diseño y su correspondiente lógica de trabajo.

En el capítulo 7.- se desarrollan ejemplos típicos, mostrando sus problemas asociados y las ventajas de tratarlos con el KDD Language.

Por último se presentan las conclusiones de todo el trabajo realizado y en el capítulo 9.- quedan planteados los posibles futuros trabajos así como también posibles extensiones del lenguaje.

2. KNOWLEDGE DISCOVERY DATA (KDD) Y DATA MINING (DM)

2.1. INTRODUCCION AL KDD – DM

Las múltiples actividades que el hombre desarrolla en la actualidad provocan la generación de cada vez más diversidad de datos, cuya cantidad crece como resultado de los avances tecnológicos que permitieron su generación, recolección y almacenamiento de manera más eficiente.

Pero estos avances deben ir acompañados con técnicas computacionales capaces de analizarlos en su profundidad. Es imprescindible lograr extraer el más alto nivel posible de información escondida en ellos, útil para la toma de decisiones, la exploración o el mejor entendimiento del fenómeno generador de esos datos.

Tradicionalmente esta tarea se hizo a mano con la ayuda de técnicas estadísticas y herramientas de consultas convencionales, pero con el crecimiento indiscriminado de datos almacenados se tornó indispensable la automatización del análisis. De lo contrario, se corría el riesgo de no aprovechar al máximo todo lo que los datos tenían para ofrecer.

La posibilidad de este nuevo análisis constituye el tema central del campo de investigación denominado Knowledge Discovery Data (KDD).

Este proceso de KDD es estudiado como un conjunto de sucesivas tareas encargadas de preparar los datos y guiarlos hacia la obtención de la mayor y mejor información posible. Como corolario de esas tareas se encuentra la aplicación de los métodos de data mining (DM), que constituyen la etapa específica de descubrimiento de conocimiento dentro de todo el proceso de KDD.

Existen diferentes maneras de describir el significado de KDD y Data Mining, a tal punto que varios autores los muestran como sinónimos.

En esta tesis se suscribe la definición adoptada por la Primera Conferencia Internacional sobre KDD, en Montreal, en 1995, donde se propuso que el término KDD fuera utilizado para la descripción de todo el proceso de extracción de conocimiento de los datos y el término data mining para referirse a la etapa específica de descubrimiento del conocimiento dentro de todo el proceso de KDD.

El reciente crecimiento del interés por data mining puede ser explicado a partir de los siguientes factores:

- Durante los 80', las grandes organizaciones construyeron gigantes bases de datos conteniendo datos acerca de sus clientes, competidores, productos, etc., que en sí mismas constituían una mina de oro, ya que ocupaban

gigabytes de datos con gran cantidad de información oculta que no podía ser fácilmente obtenida con SQL. Los algoritmos de data mining podrían hallar agrupamientos interesantes o comportamientos regulares de los datos que permitirían obtener la información oculta para una herramienta de consultas convencional.

- Como el uso de redes continua creciendo, se torna más sencilla la interconexión de diferentes bases de datos, y como consecuencia de ello, la mayor necesidad de encontrar y analizar el comportamiento de mayores volúmenes de información.

- Durante los últimos años se ha profundizado el estudio sobre diferentes técnicas de machine learning. Las redes neuronales y los algoritmos genéticos hacen más sencilla la búsqueda de conexiones interesantes entre diferentes bases de datos.

- El modo de trabajo cliente/servidor, tan expandido en el último tiempo, ha permitido el acceso directo a información proveniente de sistemas centrales. Esto hace que también se masifique la necesidad de poder analizar y extraer la información útil que subyace en esos datos.

Data Mining versus herramientas de consulta:

En primer lugar es necesario plantear que la ejecución de data mining y de las herramientas de consulta es complementaria. Las herramientas de data mining no reemplazan a las de consulta, pero brindan al usuario la posibilidad de obtener información adicional.

La elección por una de ellas tiene que ver con el grado de conocimiento que se tiene sobre la información que se quiere encontrar. Si se conoce exactamente lo que se quiere obtener (por ejemplo: ¿ quiénes compraron el producto X en un determinado mes del año ?), bastará con aplicar SQL y no tendrá sentido ejecutar una herramienta de data mining. Pero si se sabe vagamente lo que se quiere obtener (por ejemplo: ¿ cuál sería un óptimo agrupamiento de mis clientes ?), en este caso sería prácticamente imposible lograrlo con SQL, habría que ir jugando con prueba y error sobre diferentes conjuntos posibles, lo que llevaría muchísimo tiempo y no aseguraría una segmentación óptima, mientras que al aplicar un algoritmo de data mining, se lo podría resolver en minutos ó a lo sumo un par de horas.

Aplicaciones prácticas de Data Mining:

Tanto en Estados Unidos como en Europa puede observarse un crecimiento en la cantidad de entidades que optan por la aplicación de herramientas de data mining.

American Express y AT&T utilizan KDD para analizar sus archivos de clientes. En el Reino Unido, la BBC ha aplicado técnicas de data mining para analizar figuras visuales. Merece una mención particular las aplicaciones vinculadas a spatial data mining, mediante las cuales pueden ser clasificadas diferentes características geográficas. En el mundo financiero, tanto los bancos como las empresas emisoras de tarjetas de crédito incorporan permanentemente estas herramientas [Cas97].

.

En este marco es conveniente aclarar que el proceso de KDD encierra la resolución de muchos más problemas que el mining en sí mismo. El 80% del proceso de KDD consiste en la preparación de los datos y el restante 20% se encarga de la explotación. La manipulación de los datos, el uso de rutinas de cleaning y coding son etapas más importantes dentro del KDD que el reconocimiento de patrones en sí mismo. En este sentido es necesario remarcar que la correcta preparación de los datos es directamente proporcional a la mejor explotación posterior de los mismos.

2.2. DIFERENTES DEFINICIONES

Persiste una discusión acerca de la correcta definición del proceso de KDD y DM. Existen autores que no muestran diferencias entre uno y otro.

Por ejemplo en [Fra], kdd y data mining, son definidos como sinónimos, de la siguiente manera:

"data mining o knowledge discovery data (KDD) como también es conocido, es la extracción no trivial de la información implícita, previamente desconocida y potencialmente útil de los datos ... data mining es la búsqueda de relaciones y patrones globales que existen en las bases de datos grandes pero que están escondidas entre el amontonamiento de los datos."

Y en el mismo sentido los trató la AIAI en su editorial, planteando la siguiente definición:

"Data Mining y Knowledge Discovery in Databases son términos usados de manera intercambiable y una definición de alto nivel podría ser la siguiente: el proceso no trivial de identificación de patrones válidos, nuevos, potencialmente útiles y a la larga comprensibles en los datos. Data mining no es un simple proceso y no existe ninguna herramienta que pueda realizar la tarea automáticamente. Data mining puede ser auxiliada por herramientas, pero siempre requerirá tanto del experto en DM como un experto en la descripción del dominio."

Por otro lado están quienes los consideran de manera diferenciada, presentando a los métodos o algoritmos de data mining como una parte integrante de todo el proceso de KDD.

De esta manera es que en [Adr96], se adopta la siguiente definición:

"KDD es la extracción no trivial de la información implícita, previamente desconocida y potencialmente útil de los datos, mientras que el término data mining debe ser usado exclusivamente para la etapa de descubrimiento de conocimiento dentro del proceso de KDD."

Y por último, en [Fay96] se suscriben las siguientes definiciones, mostrando diferencias entre ambos conceptos:

"Knowledge Discovery in Databases es el proceso no trivial de identificación de patrones válidos, nuevos, potencialmente útiles y a la larga comprensibles en los datos. El proceso de KDD es el encargado de muestrear, preprocesar y transformar la base de datos de acuerdo a una selección requerida, para aplicarle algoritmos de data

mining y poder enumerar patrones, evaluar los resultados e identificar los subconjuntos considerados conocimiento, a partir de los patrones enumerados."

En cuanto a data mining señala:

"Data mining es la etapa dentro del proceso de KDD consistente en el análisis de los datos y el descubrimiento de algoritmos que, bajo limitaciones de eficiencia computacional aceptables, produce una enumeración particular de patrones sobre los datos."

En esta tesis, tal cual fue planteado en el punto 2.1.-, se suscribe la definición adoptada por la Primera Conferencia Internacional sobre KDD, en Montreal, en 1995, donde se propuso que el término KDD fuera utilizado para la descripción de todo el proceso de extracción de conocimiento de los datos y el término data mining para referirse a la etapa específica de descubrimiento del conocimiento dentro de todo el proceso de KDD.

A continuación se describirán en profundidad cada uno de estos procesos, mostrando sus diferentes componentes y particularidades.

2.3. DESCRIPCION DEL PROCESO DE KDD

A partir de las definiciones aportadas para abordar el proceso de KDD, se torna necesario describir la secuencia de tareas que los autores sugieren como componentes de dicho proceso.

Tampoco existe una única manera de describir esta secuencia por lo cual en primer término se mostrarán algunas de ellas, para luego adoptar la propia.

Por ejemplo en [Adr96], se plantean las siguientes etapas:

- Data Selection: Consiste en la selección de los atributos que se utilizarán de la base de datos.
- Cleaning: En esta etapa se procede a limpiar los datos seleccionados desde diferentes aspectos: corrigiendo errores de tipeo, eliminando duplicados, inconsistencias o transformando en valor nulo aquellos valores incorrectos.
- Enrichment: Es probable que se quiera adjuntar información proveniente de otras bases de datos que pueden ser propias o ajenas a la organización. La idea en esta etapa es enriquecer físicamente los datos hasta aquí estudiados con otros que sirvan para un resultado más completo del data mining.
- Coding: Se trata de establecer rangos a los valores de los atributos. Se los divide en la cantidad de subconjuntos deseados con el propósito de que no existan infinitos valores posibles. También existe otra forma de codificar y tiene que ver con la transformación de algunos valores en otros, por ejemplo suelen transformarse las fechas en años, los domicilios en regiones, etc. y por último, con frecuencia se hace flatenning, es decir se reemplaza cada atributo con sus n valores posibles, por n atributos con valores binarios.

- Data mining: Esta es la verdadera etapa de explotación de los datos, es decir, de descubrimiento de la información. Existen diferentes técnicas posibles de aplicar y cada una de ellas funcionará mejor para determinado propósito. Aquí sólo escribiremos sus nombres:
    - query tools
    - Statistical techniques
    - On line analytical processing (OLAP)
    - Case-based learning (k-nearest neighbor)
    - Decision trees
    - Association rules
    - Neural networks
    - Genetic algorithms
- Reporting: Se preparan y entregan al usuario los reportes necesarios con los resultados de los estudios realizados.

En [Fra], las etapas del proceso son sintetizadas de diferente manera:

- Selection: Se trata de seleccionar o segmentar los datos de acuerdo a algún criterio, siendo el resultado de esta etapa, la generación de subconjuntos de datos.
- Preprocessing: Esta es la etapa donde se "limpian" los datos, en la cual la información innecesaria es removida. Aquí los datos son reconfigurados con un formato consistente.
- Transformation: Los datos son transformados de manera tal de hacerlos usables y navegables.
- Data mining: En este paso se realiza la extracción de patrones de los datos. Se definen medidas de certeza que éstos deberán cumplir para ser agrupados en subconjuntos.
- Interpretation and evaluation: Los patrones identificados por el sistema deben ser interpretados de manera tal de poder servir de soporte para las decisiones que deberán tomar los usuarios.

Y profundizando en [Fay96], se pueden citar:

- Identifying the goal: Se define el objetivo desde el punto de vista del usuario y se desarrolla una descripción del dominio de aplicación.
- Creating a target data set: Se selecciona un conjunto de datos o se focaliza en un subconjunto de variables o se obtienen muestras de datos sobre los cuales se realizará el descubrimiento de la información.
- Data cleaning and preprocessing: se realizan operaciones básicas como la remoción de suciedad y se deciden las estrategias a seguir con los datos incorrectos.
- Data reduction and projection: se buscan modelos útiles para representar los datos, de acuerdo al goal definido. Se utilizan métodos de reducción o transformación para reducir el número efectivo de las variables en consideración o para encontrar invariantes representaciones de los datos.
- Matching the goals to a particular data mining method: se puede optar por alguno de los siguientes métodos, entre otros:
    - Summarization

    - Classification
    - Regression
    - Clustering
- Choosing the data mining algorithms: Se trata de la elección de los métodos que se utilizarán para la búsqueda de patrones en los datos. Esto incluye la decisión acerca de cuáles modelos y parámetros serán apropiados.
- Data mining: se ejecuta la búsqueda de patrones de interés en una representación determinada como podría ser:
    - Classification rules or trees,
    - Regression,
    - Clustering
- Interpreting mined patterns: Posiblemente se retorne a alguna de las etapas descriptas para realizar las iteraciones correspondientes. Esta etapa también puede contemplar la visualización de los patrones o modelos extractados o la visualización de los datos sobre los modelos extractados dados.
- Consolidating discovered knowledge: El conocimiento hallado puede incorporarse a otro sistema para profundizar alguna acción o simplemente documentar lo encontrado y reportarlo a las partes interesadas.

Si se hiciera un análisis global de los procesos descriptos, podrían citarse muchas similitudes, se trata en términos generales de la preparación de los datos para abordar de la mejor manera posible la etapa de extracción de conocimiento. En todos los casos se seleccionan atributos, se los limpia y se los codifica. Pero existen diferencias vinculadas con la mayor especificidad de las diferentes acciones. En la propuesta que se presentará más adelante en esta tesis, se tratará de lograr un proceso más completo aún, capaz de contemplar todas las acciones necesarias para optimizar la ejecución de los algoritmos de data mining.

2.4. DESCRIPCION DE LOS METODOS DE DATA MINING

El estudio de los métodos de data mining puede ser focalizado desde diferentes aspectos.

A partir del análisis de los objetivos, pueden distinguirse dos tipos de modelos:

- Modelos de verificación: son aquellos en los cuales el sistema se limita a verificar las hipótesis elaboradas por el usuario.

- Modelos de descubrimiento: son aquellos en los cuales el sistema es capaz de encontrar de manera autónoma nuevos patrones de datos. A su vez, estos modelos pueden ser subdivididos en dos grupos:

    - Predictivos: donde el sistema encuentra patrones con el propósito de predecir el comportamiento futuro de diferentes entidades.

- Descriptivos: donde el sistema encuentra patrones con el propósito de presentarlos al usuario de manera comprensible.

Aunque los límites entre la predicción y la descripción pueden no presentarse de manera clara, la distinción resulta útil a los efectos de comprender el objetivo final del modelo elegido.

En cuanto a un posible agrupamiento de los métodos de data mining, podrían citarse los siguientes grupos:

- Classification: los datos son clasificados en clases predefinidas.
- Regression: los datos son mapeados en variables predictivas de valores reales y se establecen relaciones funcionales entre las variables.
- Clustering: se identifican un número finito de categorías o conjuntos a los efectos de describir los datos. Está íntimamente relacionado con el método de probabilidades de estimación de densidades, el que consiste en técnicas para estimar la probabilidad de la densidad multivariable de los datos en todas las variables o campos de las bases de datos.
- Summarization: se generan descripciones compactas para subconjuntos de datos. Como por ejemplo las reglas de asociación o el uso de técnicas de visualización multivariable.
- Dependency Modeling: se descubren modelos que describen dependencias significativas entre las variables.
- Change and Deviation Detection: se descubren los cambios más significativos producidos en los datos a partir de modificaciones efectuadas previamente o valores normativos.

Para estudiar los diferentes algoritmos de data mining es necesario profundizar en el estudio de los algoritmos de machine learning o de pattern recognition.

Cualquier técnica que ayude a obtener mayor información de los datos es útil, por lo cual las técnicas de data mining forman un grupo muy heterogéneo de estudio. Aunque diferentes algoritmos puedan ser utilizados con diferentes propósitos, aquí se describirán algunos de ellos:

Query tools:

Un primer paso en un proyecto de data mining debe ser el efectuar un rápido análisis de los datos usando las herramientas de consultas tradicionales. En un primer término es importante conocer aspectos básicos de la estructura de los datos que se poseen.

Con SQL se puede acceder a la información que guardan los datos en su superficie, que en general podría tratarse del 80 % de la información interesante. El restante 20 % requiere técnicas más avanzadas y lo que sucede en la actualidad en las grandes organizaciones con grandes volúmenes de datos para analizar, es que ese 20 % resulta de vital importancia.

Una buena manera de comenzar con la aplicación de las técnicas del mining es por ejemplo la extracción de información estadísitica, como la obtención de la media correspondiente a los diferentes valores de las variables numéricas.

El resultado que puedan arrojar estos cálculos estadísticos son de mucha importancia porque permiten obtener normas con las cuales poder juzgar la performance del reconocimiento de patrones o de los algoritmos de aprendizaje. Para demostrar la verdadera eficiencia de un algoritmo en la obtención de mayor información, deberá mejorar lo preestablecido con los cálculos estadísticos.

Para avanzar un paso más en la profundización de la búsqueda, suelen aplicarse métodos denominados de naive prediction, los que consisten en cálculos combinados entre variables que arrojan resultados sencillos y que siguen marcando parámetros que necesariamente deberán ser mejorados para avanzar en la explotación de los datos.

Visualization techniques:

Las técnicas de visualización son métodos muy útiles en el descubrimiento de patrones y suelen ser usadas en el comienzo del proceso a los efectos de obtener una buena idea de dónde podría ser aconsejable encarar la búsqueda de los patrones escondidos.

Una técnica elemental que puede ser de gran valor es llamada Scatter Diagram, a partir de la cual la información de dos atributos es explayada en el espacio cartesiano y permite de manera muy sencilla, observar el comportamiento de los datos y las relaciones que se establecen entre ellos.

Los diagramas de Scatter suelen ser usados para identificar subconjuntos de datos en los cuales focalizar la profundización de la búsqueda posterior.

Así mismo, una mejor manera de encarar la explotación es a partir del uso de herramientas que efectúan descripciones tridimensionales y le permiten al usuario explorar de manera interactiva, estructuras en tres dimensiones.

Likelihood and distance:

Existen otras razones para concebir a los registros como puntos en espacios de datos multidimensionales. El espacio metafórico es muy útil en el contexto de data mining, permite determinar la distancia entre dos registros en el espacio de datos: registros que se encuentran cercanos uno del otro, muestran que tienen mucho en común, mientras que los registros alejados entre sí, representan individuos que poco tienen en común.

Con este método, los registros se transforman en puntos en el espacio y puede resultar sencilla la visualización de nubes de datos; a tal punto que con la sola inspección ocular de la gráfica, suelen identificarse interesantes conjuntos que marcan comportamientos en los datos. Pero no siempre es así y en muchas ocasiones son necesarias técnicas más avanzadas.

OLAP tools:

Las herramientas OLAP encierran la idea de poder expandir el concepto de dimensionalidad: una tabla con n atributos independientes, puede ser vista como un espacio n dimensional. No son aconsejables para aplicar sobre el tipo de

almacenamiento standard de datos, ya que en general su utilización implica la definición de múltiples claves.

OLAP resulta una importante etapa dentro del proceso de la explotación de los datos, pero es necesario remarcar diferencias importantes entre estas herramientas y los algoritmos de data mining: las herramientas OLAP no crean conocimiento y no pueden buscar nuevas soluciones, por otra parte los algoritmos de data mining no requieren un formato de almacenamiento especial de los datos y pueden actuar directamente sobre las bases de datos relacionales.

K-Nearest neighbor:

Cuando los registros son interpretados como puntos en el espacio de datos, es posible definir el concepto de vecindad: los registros que están vinculados entre sí, viven en el mismo vecindario.

La filosofía básica que encierra este método es "te comportarás como tus vecinos lo hacen". Si se busca predecir el comportamiento de determinado individuo, debe comenzarse por identificar el comportamiento de por ejemplo 10 individuos que vivan cerca suyo en el espacio de datos. Habrá que calcular la media de sus valores y esa será la predicción para el individuo investigado. Con la letra k se denota el número de vecinos que se toman para el análisis.

A este algoritmo no se lo puede definir como una técnica de aprendizaje, sino como un método de búsqueda. A los efectos de realizar las comparaciones descriptas, requiere gran cantidad de operaciones y eso determina que en la práctica, su aplicación se realice sólo sobre muestras representativas de datos o sobre bases de tamaños limitados.

Decision trees:

Al poseer una tabla con datos sobre el comportamiento de determinadas variables, podrá observarse que las tareas de predicción y la clasificación están íntimamente ligadas. Una manera de predecir si un cierto atributo mostrará determinado comportamiento, de hecho implica asumir que pertenece a un mismo tipo de atributo y que por lo tanto mostrará el mismo tipo de comportamiento.

El proceso de generar árboles de decisión consiste en determinar cuál es el atributo más influyente en la determinación del comportamiento que se quiere investigar y encontrar para él un valor umbral adecuado que permita separar en dos conjuntos a todos sus valores, seguir con este mismo procedimiento para cada uno de los atributos restantes y armar de esta forma una estructura de árbol para toda la base de datos. Existen muchos algoritmos capaces de armar estos árboles de manera automática y son en general muy efectivos, con grado de complejidad n(logn).

Estos algoritmos funcionan muy bien sobre grandes bases de datos y otra de sus ventajas es que son capaces de brindar una verdadera descripción interna de la composición de todo el proceso de decisión.

A pesar de poseer muchas ventajas, no siempre constituyen el método más apropiado.

Association rules [Sri95]**:**

Se trata de encontrar implicancias entre atributos de las bases de datos. Por ejemplo si se afirma que el 90% de las mujeres que poseen autos deportivos rojos y perros pequeños, usan Chanel 5, en realidad se estaría representando que para el 90% de los registros de la base de datos en los cuales sexo es femenino, tipo de auto es deportivo, color de auto es rojo y mascota es perro pequeño, entonces seguro que perfume será Chanel 5. Este tipo de reglas son muy usadas en marketing porque pueden marcar, como en este caso, perfiles de consumidores.

No resulta difícil desarrollar algoritmos que encuentren este tipo de reglas en grandes bases de datos, pero sin embargo puede suceder que muchas de las reglas halladas no sean realmente representativas. Para trabajar con algún parámetro a tal efecto es necesario medir el soporte y la confianza de las reglas.

El soporte de una regla es quel que indica el porcentaje de registros de la base de datos que soportan dicha regla, es decir, que contiene todos los valores mencionados en la regla. Pero en muchos casos esta medida no resulta suficiente y debe complementarse con la confianza, es decir, la proporción de registros en los que se cumple el consecuente de la regla con respecto a la cantidad de registros para los cuales se hallan presentes los valores del antecedente.

Las reglas de asociación son útiles en la medida que se tenga una idea sobre la búsqueda que se desea realizar, ya que no existe ningún algoritmo que devuelva automáticamente todas las reglas interesantes presentes en las bases de datos. Si la búsqueda de reglas no se realiza con algún tipo de guía, seguramente sucederá que junto con reglas útiles, se hallen muchas asociaciones inútiles y por el contrario si se generan búsquedas muy limitadas, seguramente el resultado obviará reglas interesantes.

Como en el caso de los árboles de decisión, resulta conveniente encarar la búsqueda de asociaciones bajo un entorno de atributos interesantes que actúen a manera guía en el estudio que se desee realizar.

Sequential patterns [Agr95]**:**

Son algoritmos capaces de descubrir secuencias en el comportamiento de los datos.

En general son usados para el reconocimiento de secuencias temporales, es decir, para determinar patrones que cumplen los datos de manera sucesiva a lo largo del tiempo.

Al igual que en las reglas de asociación, el soporte constituye una medida importante que garantiza la representatividad de las secuencias halladas. En este caso el soporte se corresponde con la proporción de registros que cumplen con el patrón de secuencia hallado.

3.- PROBLEMAS ASOCIADOS A LOS MODELOS TRADICIONALES

Continuando con el análisis de los procesos de KDD descriptos en el apartado correspondiente y profundizando dicho análisis, es preciso destacar diversos problemas asociados que serán descriptos a continuación y para los cuales se propondrán posibles soluciones en el punto 4.-.

a.- En ningún caso se plantea la existencia de herramientas capaces de automatizar las tareas del KDD, dejándolas siempre como trabajo para realizar a mano, previamente a la aplicación de los algoritmos de data mining. Esto último ya fue planteado por la AIAI cuando decía:

"Data mining no es un simple proceso y no existe ninguna herramienta que pueda realizar la tarea automáticamente."

En este punto es necesario aclarar que si bien en esta tesis no se mostrará la automatización total de las tareas, se trabajará sobre la base de poder guiar todo el proceso intentando disminuir al máximo posible el trabajo que deberá realizarse de manera artesanal.

b.- A partir del ítem anterior, se desprende el problema de la falta de documentación propia de procesos encarados en forma manual, con el consiguiente riesgo de desorganización y pérdida de control de las tareas realizadas o pendientes de realizar.

c.- Por otra parte, cuando se presentan los diferentes métodos de data mining como por ejemplo en [Sri95], [Hei96] o [Sri96], puede observarse la formulación directa de los diferentes algoritmos, obviando la discusión de todo el proceso previo de preparación y limpieza de los datos, cuando en realidad no es difícil imaginar que de aplicarse cualquiera de los métodos de extracción de información sobre los datos tal cual son almacenados en las bases de datos, sin ser previamente seleccionados, codificados o limpiados, existen altas probabilidades de hallar información sucia, redundante y en definitiva, muy poco útil.

En función de esto es necesario plantear una primera conclusión: la importancia de abordar de manera conjunta la totalidad del proceso de KDD, en un todo indisoluble la preparación de los datos y la extracción de la información, generando además la posibilidad de realizarlo con una metodología de trabajo tal que reduzca al mínimo posible las tareas manuales.

d.- Otro problema asociado a los modelos tradicionales de KDD y DM, es el hecho de estar confeccionados para encontrar las soluciones de manera general. No existen herramientas capaces de diferenciar posibles dominios de aplicación, lo que implica que cada problema por más particularidades que posea, siempre es resuelto desde un proceso general, con la consiguiente pérdida del análisis particular que cada caso requiere.

Ejemplos de lo recientemente expuesto pueden encontrarse en los sistemas difundidos en Internet, tales como Business Miner, Pilot o Knowledge Seeker, los cuales muestran que funcionarían de la misma manera si se tratara de un sistema bancario, del marketing de un supermercado o del análisis de los datos de una investigación científica.

La ventaja de este tipo de procesos radica en su posibilidad de encarar con un mismo software, gran diversidad de problemas y funcionan muy bien en la generación de árboles, reglas, etc., pero a veces son tan globales que las estructuras generadas no pueden ser aprovechadas para establecer políticas particulares debido a que no pueden ser orientadas a dominios específicos y todo esto provoca que se pierdan mejores resultados.

En este sentido se refieren en [Fay97]:
"...En el corto plazo deberíamos evitar realizar herramientas generales para concentrarnos sólo en clases especializadas de problemas. Si las posibilidades de integración de data mining existen, entonces se torna posible construir soluciones especializadas para cada caso: las aplicaciones de data mining se dedican a tareas específicas pero usando la infraestructura general y común en la comunicación con las bases de datos, representando datos y modelos y extrayendo estadísticas útiles de los datos. A medida que se conozcan más herramientas especializadas se irá concluyendo que para cierta clase de problemas, será posible resolverlos con herramientas generales, mientras que para otros, será necesario trabajar con especialistas."

Es por esto que en esta tesis, si bien se seguirán aplicando algoritmos de data mining generales, la idea es plantear la posibilidad de parametrizarlos a los efectos de que puedan actuar de manera más específica sobre un dominio determinado, aprovechando el conocimiento incorporado para obtener resultados más acordes al problema que se pretende resolver.

e.- Vinculado al ítem anterior, es importante remarcar la necesidad de que el usuario pueda contar con un ambiente que le resulte natural para poder explicitar su problema y que al mismo tiempo le permita independizarse de la necesidad de conocer acerca de los sistemas, las bases de datos, su organización, etc..

Por todo esto es que se puede plantear una segunda conclusión: la necesidad de crear herramientas capaces de permitir al usuario realizar una descripción de su problema en un lenguaje que le resulte natural y de trabajar – a partir de la parametrización de los algoritmos de data mining- sobre diferentes dominios de manera de poder aprovechar al máximo el conocimiento incorporado en cada caso y obtener por lo tanto, mejores resultados.

4.- PROPUESTA DE TRABAJO PARA ENCARAR LA SOLUCION DE LOS PROBLEMAS ASOCIADOS

Tomando como punto de partida la puntualización de los problemas asociados descriptos en el apartado anterior, al igual que las dos conclusiones enumeradas en el mismo estudio, se propone a los efectos de encarar la resolución de los problemas analizados, la formulación de un lenguaje de especificación de dominios: el KDD Language.

Dicho lenguaje surgirá como producto del estudio pormenorizado de los diferentes procesos de kdd descriptos así como también de los diversos métodos de data mining existentes y será construido sobre la base de primitivas cuya semántica refleje una metodología de trabajo propia creada para guiar al usuario en su proceso de preparación de los datos y extracción de la información.

El KDD Language servirá para afrontar cada uno de los problemas descriptos, de la siguiente manera:

a.- Con las primitivas de descripción de acciones, cuyo uso ordenado será sugerido al usuario a través de la interfaz diseñada a tal efecto, se lo guiará en todo el proceso de KDD de manera tal de reducir al máximo posible la realización manual de las tareas.

b.- En cada etapa del proceso de KDD, el usuario podrá decidir de qué manera generar el resultado correspondiente. Será sugerida la creación de nuevas tablas que contengan el conocimiento incorporado.

c.- Al estar conformado por tres componentes básicas: las primitivas de descripción del dominio, las primitivas de sistemas y las primitivas de descripción de acciones, será posible con su uso, lograr un enfoque completo de todo el proceso, vinculando naturalmente la preparación de los datos con la etapa de extracción de información y posibilitando de esta forma un trabajo unido y coherente con la concepción defendida en esta tesis: la obtención de resultados confiables en la explotación de datos es directamente proporcional a la buena preparación de los mismos.

d.- Con las primitivas de descripción del dominio, el usuario podrá definir todo su universo de trabajo, podrá representar todo el conocimiento que posea sobre el estudio a realizar. Este conocimiento podrá pertenecer a información existente en tablas y bases de datos o podrá ser producto de su experiencia personal y no correlacionarse con valores almacenados físicamente. Toda esta descripción le permitirá aplicar los algoritmos generales de data mining parametrizados según su dominio de acción y por lo tanto podrá obtener resultados más específicos y representativos de su investigación, pudiendo además orientar la búsqueda de información oculta, sobre objetos o propiedades seleccionadas particularmente para ello.

e.- Permitirá al usuario la especificación tanto de los elementos componentes de su investigación como así también las acciones a realizar a los efectos de preparar los datos. De esta forma el uso de este lenguaje, a través de la interfaz diseñada a tal

efecto, lo acercará de manera natural a los algoritmos de data mining sin estar obligado a ser un conocedor del mundo de sistemas.

5.- AMBIENTE DE PROGRAMACION ORIENTADO A UN DOMINIO

5.1. DEFINICION DE UN LENGUAJE DE ESPECIFICACION DE DOMINIOS.

A continuación se desarrollará un lenguaje de especificación, denominado KDD Language, capaz de describir un dominio determinado, a los efectos de brindarle conocimiento y en consecuencia posibilitar el mejor resultado de los algoritmos de Data Mining.

Dicho lenguaje contiene primitivas a las que el usuario no tendrá acceso, no conocerá ni usará. Serán generadas con el uso de una herramienta cuya interfaz será diseñada y presentada en esta tesis en el apartado 6.

Es posible agruparlas en tres componentes bien diferenciadas:

1.- Primitivas de Descripción del Dominio:
A partir de la interfaz que interactuará con el usuario, éste podrá definir todo el contexto de su investigación y problema a estudiar, de manera natural y sin estar obligado a conocer nada acerca de sistemas, bases de datos, etc.. Será la compilación de dicha interfaz la que generará la especificación del dominio a partir del uso de las primitivas de descripción de dominios del KDD Language.

2.- Primitivas de Sistemas:
Permiten la obtención de la información real descripta en el dominio. También en este caso serán el producto de la compilación de la interfaz diseñada, pero a diferencia de las anteriores, requieren del conocimiento de las bases de datos y tablas del negocio.

3.- Primitivas de Descripción de Acciones:
En este punto la interfaz irá guiando al usuario en todas las tareas posibles del proceso de KDD, sugiriéndole un orden pero permitiéndole también la elección de uno propio y como resultado de la traducción de esa elección, se generarán las primitivas cuya especificación definirá las acciones a realizar.

5.1.1. PRIMITIVAS DE DESCRIPCION DEL DOMINIO

CONCEPTOS: **concepto** (NomConcep).

Son elementos u objetos que constituyen el basamento del negocio que se pretende describir y sobre los cuales se trabajará para especificar sus características, cómo están relacionados unos con otros, etc., y de esa manera, proceder a describir todo el dominio del problema.

Si por ejemplo se trata de un dominio bancario, seguramente será necesario definir:
concepto (caja_de_ahorro **).** **concepto (** cuenta _corriente **).**

Si se está representando el dominio de un supermercado, habrá que definir:
concepto (producto **).** **concepto (** comestibles **).** **concepto (** proveedor **).**

También podrán definirse nuevos conceptos a partir de la generación de reglas, lo que será explicado en el apartado correspondiente a "reglas".

PROPIEDADES: **propiedades** (NomConcep, [Propiedades]).

Describen a los conceptos. Son características que van a distinguirlos unos de otros y que el usuario deberá definir.

En el caso del dominio bancario, seguramente se podrán definir las siguientes propiedades:
propiedades (caja_de_ahorro, **[** numero, titular, debito, credito, saldo**]).**

La idea es permitirle al usuario la posibilidad de expresar conocimiento personal sobre el problema y es por ello que las propiedades no responden solamente a atributos de bases de datos reales, sino que también podrá tratarse de agregados que sólo el usuario conozca.

Por ejemplo, el usuario podría conocer en la descripción del dominio de un supermercado, cuáles bebidas son alcohólicas y cuáles no lo son, y que en la tabla de Bebidas de la base de datos ésto no figure. El presente lenguaje, le permitirá al usuario la inclusión de este conocimiento en el estudio, a través de las propiedades de las bebidas. Quedando expresado de la siguiente manera:
concepto (bebida **).**
propiedades (bebida, **[** codigo, nombre, descripcion, tipo **]).**
Conteniendo la tabla Bebidas, los atributos correspondientes a los códigos, los nombres y la descripción, es decir si son lata o botella y la cantidad expresada en cm3, pero no los tipos de bebidas de acuerdo a su contenido de alcohol.

Las propiedades jugarán un rol importante en la herencia, lo cual será explicitado en el punto correspondiente a herencias.

RELACIONES: **relacion** (Tiporel, NomConcep1, NomConcep2).

Vinculan conceptos entre sí. Pueden existir diferentes tipos de vínculos, aunque el más frecuente es el "es un".

Las relaciones permiten agrupar conceptos.

La relación "es un", permite reconstruir las taxonomías presentes en los conceptos.

Se muestra cómo es posible expresar que la carne es un comestible y que está en oferta:
relacion (esun, carne, comestible **).**
relacion (esta_en, carne, oferta **).**

Parte del árbol de taxonomías de los productos quedará representado de la siguiente manera:
relacion (esun, pollo, carne **).**
relacion (esun, vaca, carne **).**
relacion (esun, carne, comestible **).**
relacion (esun, comestible, producto **).**
relacion (esun, ropa, producto **).**
Pollo y Vaca son carnes y por lo tanto cuelgan de dicho nodo. La carne es un comestible y como tal es un nodo "hijo" del nodo comestible. A su vez, los comestibles son productos al igual que lo es la ropa, por lo tanto ambos nodos cuelgan como "hijos" del nodo raíz productos.

HERENCIA: **herencia** (Tipoher, NomConcep, Tiporel).

Permite que los conceptos "hereden" las propiedades de otros conceptos con los cuales están vinculados según la relación especificada.

No hay que confundir con el concepto de herencia de programación orientada a objetos. No se presentan problemas asociados.

Existen 3 tipos de herencia:

"agregacion": agrega al concepto mencionado en la herencia, todas las propiedades de todos los conceptos vinculados con éste a través de la relación Tiporel.

herencia ("agregacion", carne, esun **).**
herencia ("agregacion", comestible, esun **).**

A las propiedades definidas para los comestibles, se les agregan las de los productos y lo mismo sucede con la carne, recibiendo como agregadas todas las propiedades de los comestibles.

A su vez, la carne puede formar parte del árbol de los perecederos, además de pertenecer al de los comestibles:

relacion (esun, carne, comestible **).**
relacion (esun, carne, perecedero **).**

herencia ("agregacion", carne, esun **).**

Se mostró con este ejemplo, que un concepto puede heredar propiedades, a través de la relación esun, de más de un concepto. En este caso, son todas agregadas, quedando como trabajo para el usuario la previsión de las inconsistencias.

"sustitucion": sustituye todas las propiedades del concepto mencionado en la herencia, por todas las propiedades definidas para todos los conceptos vinculados con éste a través de la relación Tiporel.

herencia ("sustitucion", carne, esta_en **).**

Por un lado, la carne heredó las propiedades de los comestibles, pero por estar en oferta, debe sustituir aquellas propiedades que se superpongan con las de las ofertas, es así que su precio será reemplazado por el de la tabla de ofertas.

"nula": no hereda ninguna propiedad del concepto vinculado a través de la relación Tiporel. En general se la utiliza al modificar un dominio definido con anterioridad.

Las herencias están directamente relacionadas con el diccionario de datos, las define el usuario y sirven para identificar el lugar físico de donde se deben obtener los valores correspondientes a las propiedades de los conceptos.

REGLAS: **regla** (NuevoConcepto, [Antecedentes]).

Las reglas permiten generar nuevos conceptos. En los cuales la conclusión será ese nuevo concepto y los antecedentes serán condiciones que deberán cumplir algunos conceptos ya existentes para conformar el nuevo.

Una vez determinado ese nuevo concepto, los antecedentes pasarán a ser sus propiedades. Será el lenguaje por sí solo quien se encargará de dejarlo expresado de esta manera. En el caso que el antecedente se componga por dos o más conceptos, entonces el lenguaje generará las relaciones "esun" y las herencias correspondientes a los efectos de que el nuevo concepto creado, posea todas las propiedades de todos los conceptos intervinientes en el antecedente.

En el siguiente ejemplo se mostrará cómo pueden participar dos conceptos en el antecedente de una regla, transformándose luego en las propiedades del nuevo concepto y cómo quedarían generadas las relaciones y herencias planteadas.

Se estudiará el caso en el que se quieran identificar a los buenos ayudantes de segunda que a su vez son muy buenos alumnos en una Facultad:

regla (buen_docente_alumno, **[[** docente, [cargo, "=", segunda], [concurso, "=", regular], [encuesta, "=", excelente] **], [** estudiante, [promedio, ">=",8 **]).**

Donde ya existían definidos:
concepto (docente **).**
concepto (estudiante **).**
propiedades (docente, **[** nombre, legajo, fecha_nacimiento, sexo, cargo, concurso, dedicacion, encuesta **]).**
propiedades (estudiante, **[** nombre, lu, fecha_nacimiento, sexo, promedio **]).**

En este caso acaba de crearse:
concepto (buen_docente_alumno **).**
propiedades (buen_docente_alumno, **[** nombre, legajo, fecha_nacimiento, sexo, lu, cargo = segunda, concurso = regular, dedicacion, encuesta = excelente, promedio >= 8] **).**
relacion (esun, buen_docente_alumno, docente **).**
relacion (esun, buen_docente_alumno, estudiante**).**
herencia ("agregacion", buen_docente_alumno, esun **).**

5.1.2. PRIMITIVAS DE SISTEMAS

DICCIONARIO DE DATOS:
diccionario (Concepto, Propiedad, Basededatos.Tabla.Atributo).

El diccionario permite relacionar los conceptos y propiedades descriptos en el dominio con los datos reales de las bases de datos.

Para cada propiedad de cada concepto es necesario explicitar en qué lugar físico se encontrarán sus valores, o sea con qué atributo de la realidad se corresponde y dónde se lo localiza.

Esta tarea corresponde realizarla al área de sistemas por ser la que conoce la ubicación física de los datos: bases de datos, tablas, atributos, operadores posibles, valores posibles de los atributos, claves, relaciones, etc. .

Por ejemplo, en la descripción de un dominio determinado, se definió lo siguiente:

concepto (comestible **).** **concepto (** carne **).**
propiedades (comestible, **[** codigo, nombre, categoria, precio **]).**
propiedades (carne **[** codigo, nombre, corte, descripcion, precio **]).**
relacion (esun, carne, comestible **).**
herencia (carne, comestible, esun **).**

Y se conoce que la organización de la base de datos **Negocion**, está compuesta, entre otras, por las siguientes tablas:

Comestibles:
Atributos: Cod, NomProd, Cat, Costo, Precio.
Carnes:
Atributos: Cod, NomProd, Corte, Descripción, Precio.

Por lo que será necesario definir el siguiente diccionario de datos:

diccionario (comestible, codigo, $Negocion.Comestibles.Cod$ **).**
diccionario (comestible, nombre, $Negocion.Comestibles.NomProd$ **).**
diccionario (comestible, categoria, $Negocion.Comestibles.Cat$ **).**
diccionario (comestible, precio, $Negocion.Comestibles.Precio$ **).**

diccionario (carne, corte, $Negocion.Carnes.Corte$ **).**
diccionario (carne, descripcion, $Negocion.Carnes.Descripcion$ **).**
El resto de las propiedades de las carnes no necesitan ser descriptas en el diccionario porque son heredadas de los comestibles y por lo tanto sólo es necesario recurrir al diccionario de los mismos.

En este punto será necesario aclararle al usuario que para lograr la consistencia en las herencias, deberá trabajar con tablas indexadas. Para ello se le mostrará, durante la interfaz, cuáles son las claves de cada una de las tablas y si no

las hay, se le indicará que será asumido como tal, el primer atributo de cada una de ellas.

5.1.3. PRIMITIVAS DE DESCRIPCION DE ACCIONES

Estas primitivas serán el resultado de la compilación de las tareas que defina el usuario a través de la interfaz diseñada a tal efecto.

Se corresponden con las etapas del proceso de KDD.

El KDD Language permite que el usuario pueda seleccionar el orden en que desee ejecutarlas e inclusive también podrá optar por la no ejecución de algunas y la ejecución reiterada de otras. Pero al mismo tiempo, tiene la facultad de ir guiando al usuario en todo el proceso, sugiriendo un orden definido de acciones a desarrollar.

Esto último está íntimamente ligado al estudio realizado con el objetivo de proponer una metodología de trabajo propia y en tal sentido, serán expuestas a continuación todas las acciones posibles de realizar con el KDD Language y en el orden en que se sugiere ejecutarlas:

- filter
- select
- clean
- enrich
- code
- data mining
- report

Cada una de estas etapas constituirá una acción. La aplicación sucesiva del filter, select, clean, enrich y code, generarán la tabla que contendrá los datos preparados para la ejecución del método de data mining elegido, el cual deberá ser determinado en la función "objetivo". Una vez ejecutado dicho algoritmo, será convocado automáticamente el report, a los efectos de transformar la estructura obtenida, en la salida deseada.

Etapa de FILTERING: **filter** (BD.Tbl, [cond1, ..., condn], Salida).

BD.Tbl = Base de datos.Tabla de donde se obtendrá los valores con los cualres se trabajará.
cond1, ..., condn = describen las condiciones que deberán cumplir los valores de los atributos.
Salida = tabla de salida con las condiciones filtradas.

En esta etapa el usuario puede comenzar a elegir grupos de datos con los que le interesará trabajar.

Se trata de exigirles condiciones a las propiedades de los conceptos definidas en el dominio. Y su utilidad radica en reducir la cantidad de datos sobre la cual se aplicarán los métodos de data mining. No tendría sentido, por ejemplo, quedarse con

todos los estudiantes si lo que se está estudiando es el rendimiento en los estudios de aquellos que trabajan más de 35 hs. semanales, o quedarse con todos los docentes si lo que se está evaluando es la influencia de las docentes madres en los grupos de investigación de la Facultad.

Es una acción de filtrado por fila, vale decir que el resultado de esta etapa será la reducción en registros de la base de datos.

La implementación de esta etapa es ejecutada como un nexo entre las componentes descriptivas del lenguaje y las típicas de sistemas, ya que por un lado el usuario es quien elige qué conceptos y propiedades usará, pero para seleccionarlos realmente con sus valores reales, es necesario acudir a la información provista por sistemas.

Por ejemplo, en la idea de estudiar la influencia de las docentes madres en las investigaciones universitarias, resultaría útil filtrar a los docentes varones, para quedarse sólo con las mujeres, lo cual será traducido por la herramienta al siguiente formato:

filter ($UBA.Docentes$, **[** $sexo$, "=", $masculino$ **]**, $DtesMujeresTbl$ **).**

Quedará generada la tabla DtesMujeresTbl que contendrá los mismos atributos que la tabla Docentes pero con los registros correspondientes a las docentes mujeres, es decir que en el atributo sexo aparecerá un único valor, "femenino".

Etapa de DATA SELECTION:

select (BD.Tbl, [atributo1, ..., atributon] , Salida).

BD.Tbl = Base de datos.Tabla de donde se obtendrá los valores con los cuales se trabajará.
atrib1, ..., atribn = atributos que seleccionará el usuario.
SALIDA = tabla de salida con los atributos seleccionados.

En esta etapa el usuario elegirá las propiedades de los conceptos definidos con las que le interesará trabajar. Aquí orienta su búsqueda, ya que si busca, por ejemplo, reordenar las góndolas de un supermercado, no le interesa tanto saber si el cliente fue varón o mujer, con lo cual puede descartar el sexo.

Es una acción de selección por columna, es decir que el resultado será la reducción de atributos con los que se va a trabajar.

Esta acción hay que aplicarla sobre el resultado de la etapa anterior.

Continuando con el ejemplo de las mujeres docentes, si se desea conservar sólo el cargo, concurso, dedicación y encuesta de las mujeres docentes, la herramienta partirá de la tabla generada en la etapa de filtering y se quedará con los atributos requeridos, generando el siguiente código:

select ($UBA.DtesMujeresTbl$, **[** $legajo$, $cargo$, $concurso$, $dedicacion$, $encuesta$ **]**, $CargoMujeresTbl$ **).**

Quedará generada la tabla CargoMujeresTbl cuyo contenido serán los valores correspondientes a los atributos seleccionados de la tabla DtesMujeresTbl.

Es imprescindible recordar que se está trabajando sobre grandes volúmenes de datos, por lo cual resulta aconsejable reducir primero la cantidad de registros con los que se va a trabajar y sobre los cuales se irán ejecutando las sucesivas acciones. Es por ello que se sugiere el filtrado de los datos en primer término y luego sobre esa tabla generada, con menor cantidad de registros, entonces sí proceder a seleccionar los atributos deseados.

Etapa de CLEANING: **clean** (BD.Tbl, [cond1, ..., condn] , %, Salida).

BD.Tbl = Base de datos.Tabla de donde se realizará la limpieza.
cond1, ..., condn = describen las condiciones que deberán cumplir los valores de los atributos.
% = parámetro de salida que indicará el porcentaje de valores correctos encontrados en la tabla de entrada.
Salida = Podrá ser la convocatoria al procedimiento VistaPrc o al NullPrc.

El usuario podrá determinar los valores que considera incorrectos para las propiedades de los conceptos definidos y el porcentaje de la base que desea conservar.

Será informado acerca del porcentaje de registros limpios encontrados y entonces el usuario decidirá si acepta esa limpieza. En el caso de ser aceptada, deberá pedir la salida deseada:

- Armar vistas de la base de datos real con los registros con datos válidos. Salida = VistaPrc.
- Reemplazar por valor "null" los datos inválidos. Salida = NullPrc.

Siguiendo con el ejemplo anterior, se procederá a efectuar el cleaning sobre los concursos, para ello se consideran correctos los valores R correspondientes a los concursos regulares o I correspondientes a los interinos.

En función de esta especificación, será generado el siguiente código:

clean ($UBA.CargoMujeresTbl$, **[** $Concurso$, "<>", R, I **]**, %, "VistaPrc" **).**

Se generará una vista de la tabla CargoMujeresTbl, que contendrá sólo los datos correctos al mismo tiempo que se indicará qué porcentaje representa esa vista limpia del total de los datos.

Si la vista representa un porcentaje aceptable de la tabla total, se decide efectuar la limpieza y para ello se procede a reemplazar los valores inválidos por $ null $, convocando nuevamente a la acción de clean.

clean ($UBA.CargoMujeresTbl$, **[** $Concurso$, "<>", R, I **]**, %, "NullPrc" **).**

Por último se procede a la eliminación de los registros con valores $ null $, lo cual es realizado por la acción filter.

filter ($UBA.CargoMujeresTbl$, **[** $Concurso$, "=", $null$ **]**, "ListaTbl" **).**

Ahora sí quedó generada una tabla apropiada para trabajar en las acciones que se sucederán, ListaTbl, sin datos incorrectos.

Esta etapa es muy importante ya que influye de manera directa sobre el resultado de la aplicación del mining. La suciedad en los datos implica mayor probabilidad de obtener basura en el resultado de la extracción de la información. Es por ello que su ejecución es altamente recomendada a los efectos de garantizar buenos resultados.

No es aconsejable ejecutar esta etapa antes que el filtrado o la selección, ya que carece de sentido limpiar datos que luego no serán utilizados.

Etapa de ENRICHMENT:

enrich (BD.Tbl1.atributo_clave, BD.Tbl2, **[** atributo1, ..., atributon **]**, Salida **).**

BD.Tbl1.atributo_clave = Base de datos.Tabla que se desea enriquecer. El atributo_clave se corresponde con la clave de dicha tabla, que deberá ser indexada.
BD.Tbl2 = Base de datos.Tabla evocada a los efectos de obtener información de ella para enriquecer la anterior
atrib1, ..., atribn = atributos seleccionados para enriquecer la base anterior.
SALIDA = tabla de salida. Compuesta por los atributos de la tabla a enriquecer más los atributos agregados.

El usuario podrá necesitar información que él sabe que no existe en su base de datos, pero que sí podría obtenerla de otras áreas de la organización. Esta etapa le permitirá plantear su necesidad. Entonces podrá seleccionar de otras bases qué atributos incorporar a su tabla de trabajo. Deberá expresar cuál es la clave de su tabla a enriquecer porque necesariamente deberá coincidir con la de la tabla nueva, de lo contrario se le informará que no será posible ejecutar el enriquecimiento.

Para continuar con el ejemplo anterior, se decide incorporar si son investigadoras del CONICET y como esa información existe en las tablas de esa institución, se la requiere de la siguiente manera:

enrich ($ListaTbl.legajo$, $Conicet.Investigadores$ **[** $rango$, $nombre_proyecto$ **]**, $ListaConicetTbl$ **).**

En este momento, habiendo atravesado las diferentes etapas descriptas, se cuenta con la tabla ListaConicetTbl, cuyos atributos son: legajo, cargo, concurso, dedicación, encuesta, rango, nombre_proyecto, de los cuales el correspondiente a concursos es el que no posee valores incorrectos.

Etapa de CODING: **code** (BD.Tbl, atributo, **[** cond1, ..., condn **]** , Salida**).**

BD.Tbl = Base de datos.Tabla de donde se obtendrá los valores con los cuales se trabajará.
cond1, ..., condn = condiciones que darán cumplir los atributos para ser codificados.
SALIDA = tabla de salida con los atributos codificados.

En esta etapa el usuario definirá cómo quiere dividir los valores posibles de las propiedades de cada concepto. El resultado será la división en subconjuntos de los valores de los atributos codificados.

Por otro lado existe una codificación automática que hará el informático, del tipo 1 para mujeres y 2 para varones, etc..

Si por ejemplo el usuario quiere codificar los resultados de las encuestas en 1,2 = 1, 3,4 = 2, 5,6 = 3, 7 = 4, entonces se generará el siguiente código:

code ($UBA.CargoMujeresTbl$, $Encuesta$, **[** 1,2 = 1, 3,4 = 2, 5,6 = 3, 7 = 4 **],** $CargoMujeresTbl$ **).**

Como resultado de esta etapa quedarán cambiados los valores del atributo encuesta en la tabla CargoMujeresTbl, codificados en 4 grupos según se especificó en el código.

Etapa de DATA MINING:

objetivo (Metodo, **[** Cond1, ..., Condn **],** Estructura **).**

Metodo: corresponde al método de data mining elegido por el usuario para aplicar en el proceso de KDD. Podrá ser Association Rules, Sequential Patterns, Decision Trees, entre otros.
Cond1, ..., Condn: lista de condiciones que requiere el algoritmo de data mining elegido para poder ser aplicado.
Estructura: Es la estructura de salida que devuelve la ejecución del método de data mining elegido.

Aquí se llega una vez especificado y ejecutado todo lo anteriormente detallado.

Es el momento de la definición del objetivo del estudio que se desea realizar, se trata de la elección del algoritmo de data mining que quiera aplicarse. Dicha selección podrá hacerse sobre la oferta de algoritmos disponibles en la programación de la herramienta.

Podría suceder que con el presente lenguaje se intentase especificar solo acciones de preparación de datos, en cuyo caso el método expresado se referirá a la última etapa que se desee desarrollar.

Su implementación consistirá en la convocatoria al algoritmo de DM elegido para la explotación y se le envía como parámetros todo lo descripto en el lenguaje.

Etapa de REPORTING: **report** (Entrada, Salida).

En esta etapa el usuario seleccionará la manera en que desee obtener el reporte de la explotación de los datos realizada.

Podrá optar por gráficos o por la visualización de tablas con los resultados obtenidos.

Esta etapa será ejecutada después de la aplicación de los algoritmos de data mining y es por ello que no tendrá parámetros de entrada que deba brindar el usuario, éstos serán generados automáticamente por la salida de la etapa de data mining.

En cuanto al formato de salida del reporte se podrá optar por gráficos, textos o ambos y en cada caso el lenguaje se encargará de elaborar lo requerido para mostrarlo al usuario y completar de esta forma todo el ciclo del proceso de KDD.

6.- AMBIENTE DE PARAMETRIZACION (INTERFAZ)

6.1. ESTRUCTURA DE DISEÑO

A continuación será presentada una estructura de diseño elaborada a los efectos de mostrar el orden correspondiente a las especificaciones a realizar con el uso de KDD Language.

El usuario será guiado en la especificación de todos los procesos, ya sean de definición del dominio como en la descripción de las acciones.

En cuanto a la descripción del dominio (diseñado con cuadros de línea llena), el orden a seguir será obligatorio, es decir que no estará permitido definir propiedades si previamente no fueron definidos los conceptos, y así con el resto de las tareas.

En el caso de las acciones (diseñado con cuadros de línea punteada), se enseñará el orden establecido como guía para el usuario pero no será necesariamente obligatorio, ya que el lenguaje creado permite modificar el orden de especificación de las acciones.

En segundo término, se verá el diseño de la lógica de trabajo que encierra la estructura de la interfaz planteada.

El usuario podrá optar por continuar con un estudio comenzado previamente o por iniciar uno nuevo. En el primer caso, podrá saltear etapas, si es que habían sido resueltas anteriormente, de lo contrario estará obligado a especificar su dominio en el orden planteado.

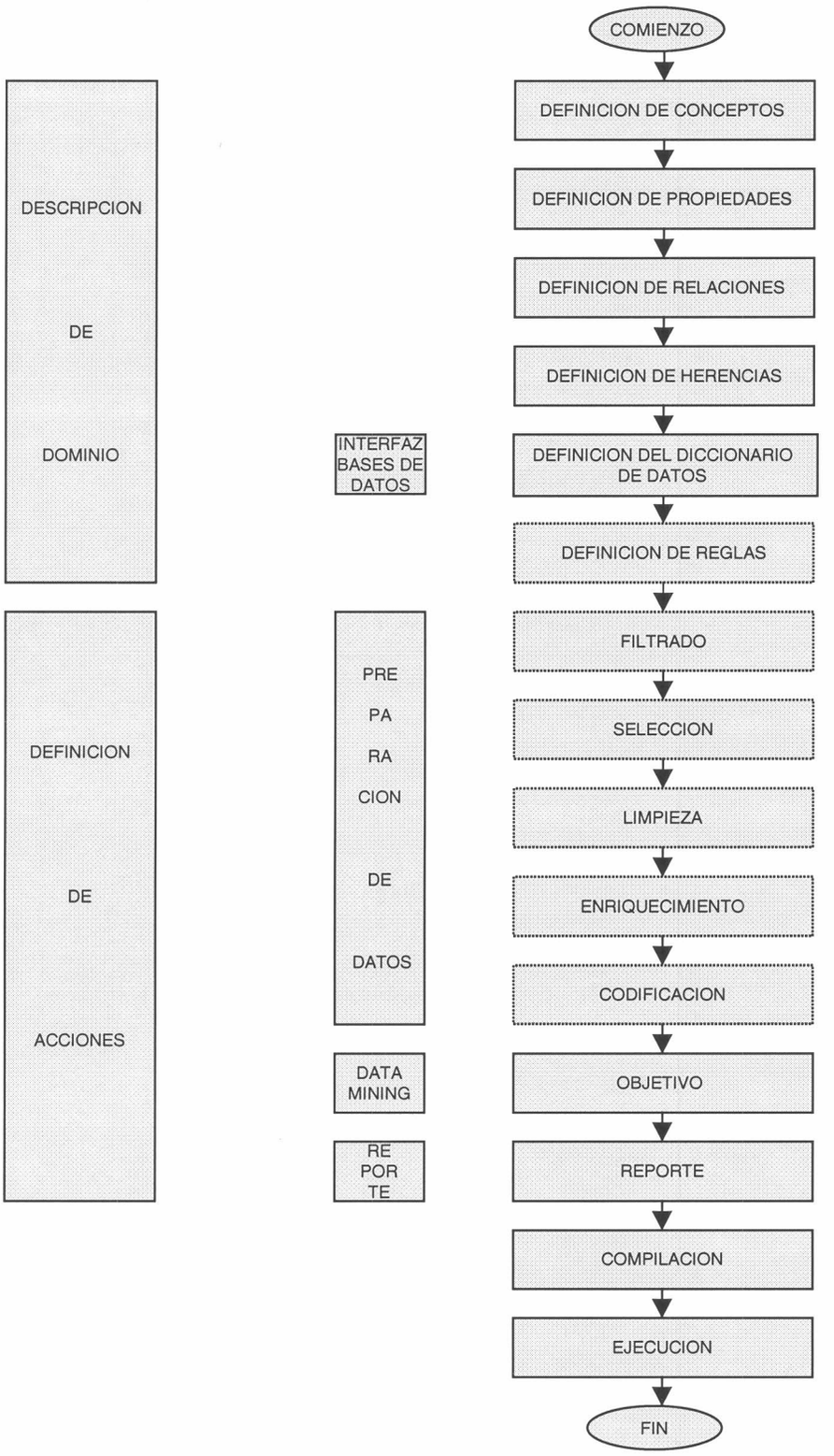
En el caso de la descripción de las acciones, se sugiere con línea de puntos el orden adecuado de su definición para efectuar la preparación de los datos, respondiendo a la metodología de trabajo que encierra el lenguaje creado.

Podría no seguirse el orden sugerido y en ese caso el usuario podría avanzar en el proceso de kdd, a través de las líneas llenas.

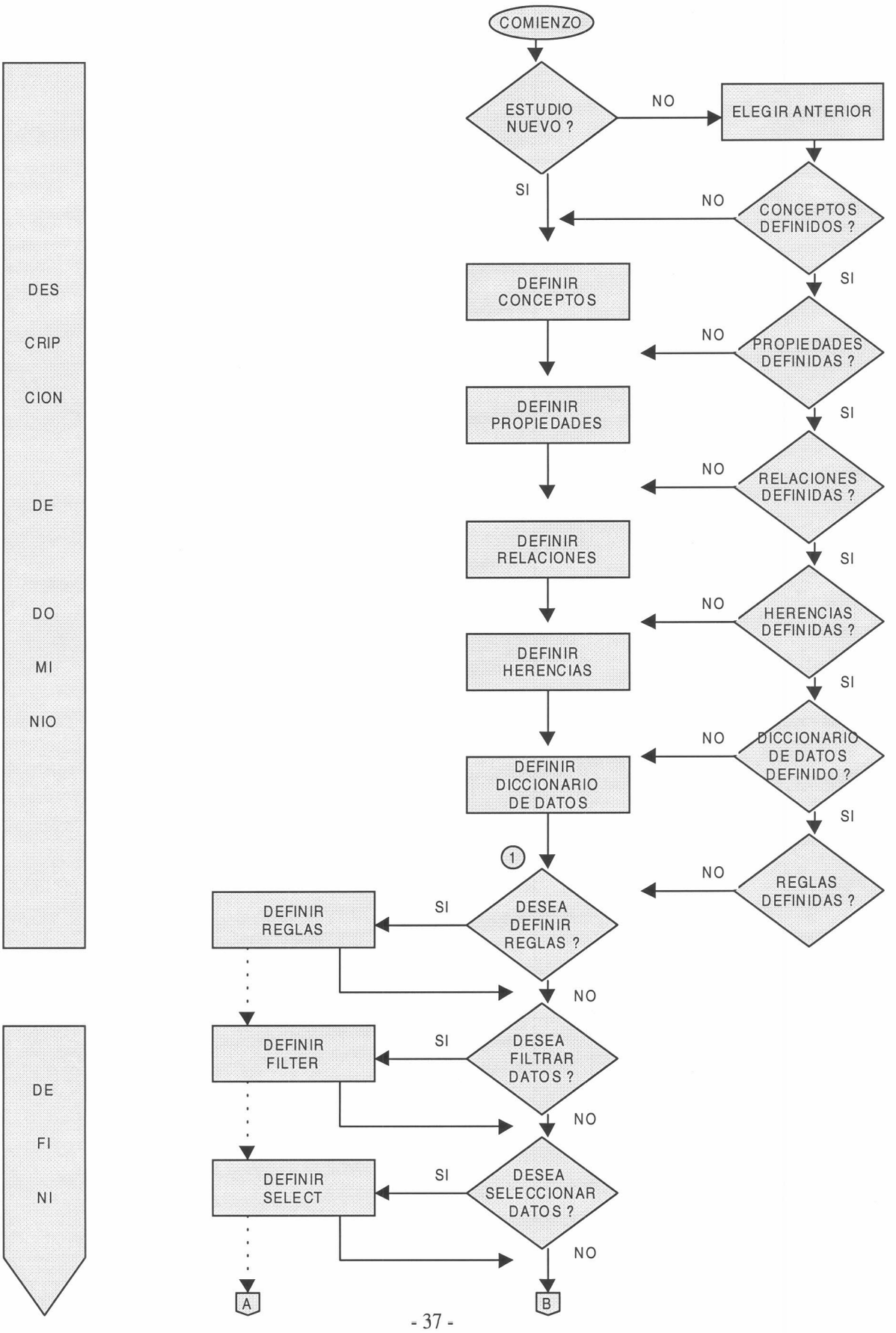
En el final de la especificación, deberá compilarse todo el proceso y finalmente estará listo para su ejecución, momento en el cual se traducirá todo lo especificado al programa correspondiente, encargado de realizar las tareas especificadas, aplicar el algoritmo de data mining elegido y elaborar el reporte final.

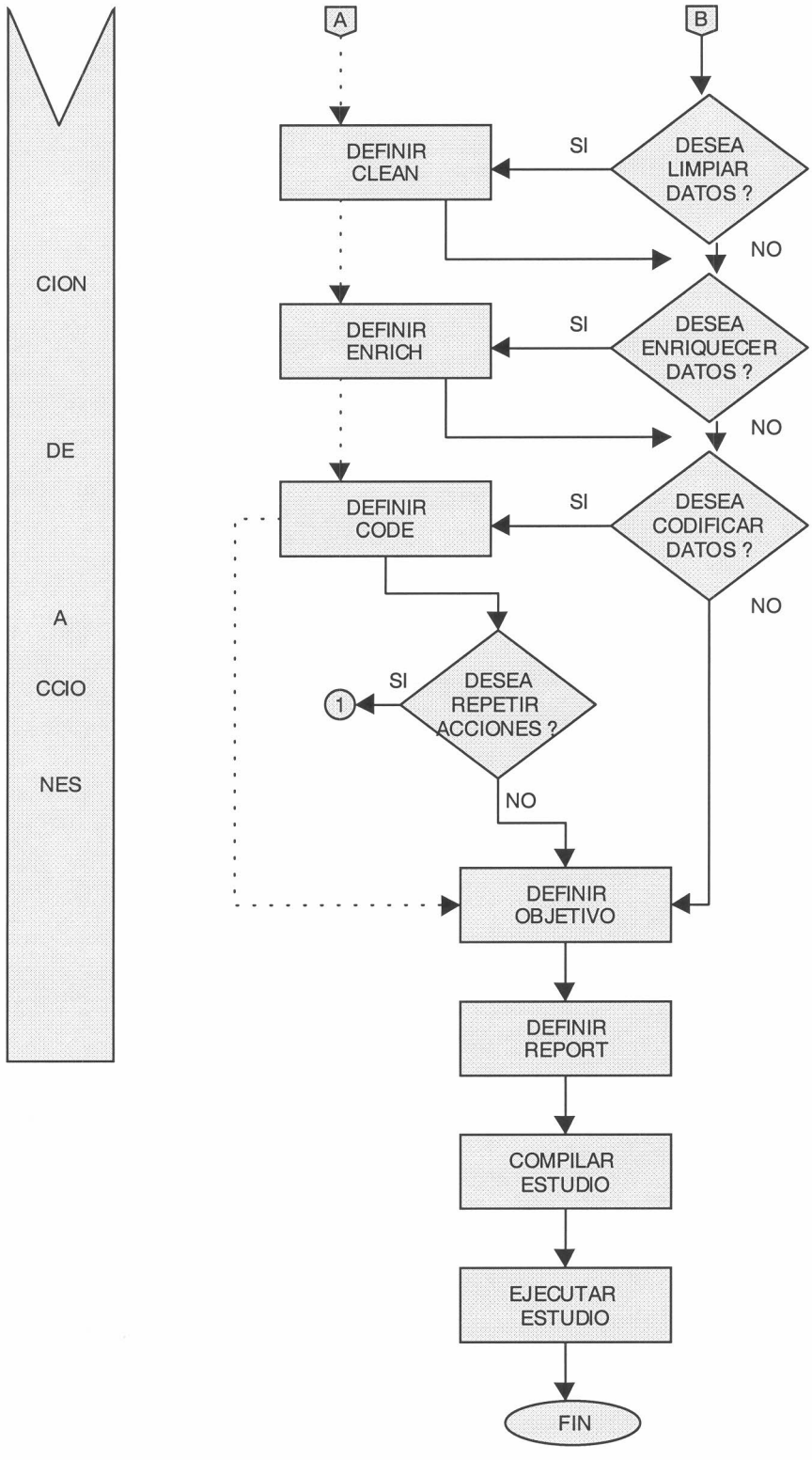
Toda esta estructura deberá corresponderse con el diseño de pantallas que puedan llevar al usuario de manera amigable, a través de todo el proceso, sin que tenga la necesidad de conocer sobre sistemas.

6.1.1. PRESENTACIÓN DE PROCESOS



6.1.2 LOGICA DE TRABAJO




CION
DE
A
CCIO
NES
A
B
DESEA LIMPIAR DATOS ?
SI
DEFINIR CLEAN
NO
DESEA ENRIQUECER DATOS ?
SI
DEFINIR ENRICH
NO
DESEA CODIFICAR DATOS ?
SI
DEFINIR CODE
NO
DESEA REPETIR ACCIONES ?
SI
1
NO
DEFINIR OBJETIVO
DEFINIR REPORT
COMPILAR ESTUDIO
EJECUTAR ESTUDIO
FIN

6.2. ESPECIFICACION

La interfaz para la ejecución del kdd language se plantea haciendo eje en aspectos diferentes, de acuerdo se trate de obtener la mayor participación del usuario conocedor del problema y del estudio a realizar, o del informático conocedor de sistemas y la implementación en sí misma de las diferentes acciones que realizará esta herramienta.

Cuando se trate del diálogo con el usuario, se hará mayor hincapié en el diseño, ya que debe resultar amigable para quien no conoce necesariamente de sistemas.

6.2.1. FASE 1, DESCRIPCION DEL DOMINIO:

Se genera pantalla de inicio con nombre de la herramienta y descripción de su utilidad.

Se solicita al usuario que coloque nombre al estudio que realizará o si prefiere seleccionar un estudio ya hecho para volver a trabajar sobre él.

Si se trata de un estudio nuevo, será guiado a realizar especificaciones de forma ordenada. En la primera instancia, no estará habilitado para definir otra cosa más que los conceptos determinantes de su dominio y luego sus propiedades. Sólo cuando hubiese cumplido con ambas tareas, podrá pasar a definir las relaciones que existan entre los conceptos y recién después estará habilitado para definir herencias. La descripción del diccionario de datos es obligatoria al igual que la de los conceptos y sus propiedades y será necesario especificarlo en este orden, antes de la descripción de reglas, si las hubiere.

Para la definición del diccionario será necesaria la participación de un conocedor de las bases de datos y tablas de la organización, ya que deberá establecer los nexos entre las especificaciones del usuario y el lugar físico donde se encuentren almacenados los datos.

En función de lo recientemente detallado, al comenzar un estudio nuevo, se le sugiere al usuario la confección del listado de conceptos del dominio. Esto se realiza mediante ventanas que le preguntarán cuáles son los elementos que componen su tema de estudio, su negocio o investigación.

En esta instancia se le aclara al usuario que existen tres posibilidades:

- Definir el dominio de todo su universo, para que quede establecido y sirva para efectuar posteriores diferentes consultas de DM, aunque no vaya a utilizar todos los elementos descriptos aquí, en este estudio.
  [Luego se reducirá este universo en la etapa de selección de datos]
- Definir el dominio que abarca sólo la consulta particular de DM que se hará en este estudio.
  [También podrá reducirse luego el dominio definido mediante el filtrado, pero seguramente se obviará la selección de los datos, ya que se supone que ya fue descripto todo lo útil]
- Ampliar la definición de un dominio existente en un estudio previo.

La herramienta se encargará de compilar los datos ingresados, generando conceptos de acuerdo al formato definido en el KDD language. Esto resulta transparente al usuario.
Quedan generados:
concepto (NomConcep 1**).** ... **concepto (**Nomconcep n**).**

Se le explica al usuario que deberá describir las características de cada uno de los elementos definidos, mediante una palabra que identificará a cada una de ellas.

Para ello será ayudado por el diseño de la herramienta que le irá mostrando cada uno de los elementos que definió anteriormente y se le habilitarán ventanas para que coloque las propiedades mencionadas.

La herramienta generará el listado de propiedades de los conceptos de acuerdo al formato definido en el KDD language. Esto resulta transparente al usuario.
Quedan generadas:
propiedades (NomConcep 1, [prop 1, ..., prop n]).

Se le pregunta al usuario qué relaciones existen entre los elementos del dominio. Es necesario explicarle qué es una relación y la utilidad de la expresión "es un".

Para ello se le mostrará el listado de los elementos definidos y se le brindará la posibilidad de unirlos y colocarle nombre a las uniones efectuadas.

El usuario cliqueará dos elementos y luego el diseño le preguntará con qué nombre quiere realizar esa unión.

La herramienta generará las relaciones de acuerdo al formato establecido en el KDD language. Esto resulta transparente al usuario.
Quedan generadas:
relacion (nomRel, NomConcep1, NomConcep2).

Se le muestra al usuario los árboles de taxonomía armados a partir de las relaciones "es un".

Se le indica al usuario que es el momento de establecer las herencias, es decir, de acordar qué características serán compartidas por padres e hijos en las taxonomías y cuáles se mantendrán independientes y serán propias.

Se irán armando pantallas que recorriendo los niveles de las taxonomías desde las hojas hasta la raíz, mostrarán al usuario, para cada concepto, sus propiedades ya definidas, dándole la posibilidad de establecer cuáles son compartidas por padre e hijos y cuáles son propias.

La herramienta generará el listado de herencias de acuerdo al formato definido en el KDD Language.
Quedan generadas:
herencia (agregacion, NomConcep, nomRel).

Se deben dejar especificadas las direcciones físicas para buscar todos los datos del problema. La vinculación entre la especificación del dominio del problema y las BD reales, sus tablas, atributos, etc..

Aquí el usuario necesitará de la acción del informático, quien conoce la composición de las BD reales y quien -teniendo en cuenta las relaciones y herencias definidas previamente por el usuario-, deberá armar el diccionario de tal manera que todos los conceptos y sus propiedades puedan vincularse con los valores de los atributos y registros de la BD real.

Se generará el diccionario de acuerdo al formato definido en el KDD Language. Este diccionario queda transparente al usuario.
Quedan generados:
diccionario (NomConcep, Prop, BD.Tabla.Atributo).

Para aquellas características no existentes en la realidad, la herramienta, a través de su diseño deberá avisarle al usuario que no puede obtener esos datos y que él deberá generarlos, dándole la posibilidad de hacerlo mediante el armado de reglas o decidiendo si resultará más conveniente el agregado de atributos a la BD real y el llenado de todos sus registros en este nuevo campo.

Si el usuario se decide por el armado de reglas, se le consultará qué nombre se le asignará al consecuente, es decir al nuevo concepto y luego se le mostrarán los conceptos ya definidos con sus respectivas propiedades, para que el usuario pueda elegir los antecedentes de la regla, cliqueando en las propiedades seleccionadas y definiendo las condiciones que deberá cumplir cada una de ellas.

Aparecerá una pantalla indicándole al usuario que ha concluido la primera fase de su estudio, la definición preliminar del dominio de su problema y que ya está preparado para el comienzo de la segunda fase del estudio, la fase de las acciones o etapas.

6.2.2. FASE 2, DESCRIPCION DE ACCIONES

Se le mostrará una pantalla con la descripción de todas las etapas existentes, sus nombres y breve descripción de lo que hace cada una.

El usuario deberá elegir el orden en que desea convocarlas, definirlas y ejecutarlas. Cliqueará el nombre de cada una de ellas a medida que requiera su uso. También tendrá la posibilidad de dejarse guiar por la herramienta quien irá avanzando en un orden establecido por default. (será el orden en que se presentan descriptas en este documento).

Etapa de FILTERING:
(si esta etapa no es elegida en primer término, sus tablas de entrada serán el resultado de la Salida de la última etapa ejecutada anteriormente).

Se le brindará al usuario una descripción más detallada de lo que significa esta etapa y la importancia de su ejecución.

La herramienta buscará en la BD real, de acuerdo al diccionario, relaciones y herencias, los diferentes valores posibles para cada propiedad de cada concepto definido en la fase anterior del estudio, y mostrará al usuario para que pueda elegir el filtrado y en consecuencia achicar el universo. También dará al usuario la posibilidad de elegir qué salida desea obtener con ese filtrado, dándole las siguientes opciones:

- Agregar un campo a los registros filtrados, marcándolos. La herramienta aconsejará optar por esta salida. Salida = MarcarPrc.
- Armar tablas nuevas que posean sólo los registros filtrados. Salida = *Tbl.
- Armar una vista de las tablas con los registros filtrados. Salida = VistaPrc.

La herramienta traducirá al formato definido en el KDD Language, los filtering definidos por el usuario.

Quedan generados:
filter (BD.Tabla, [cond1, ..., condn], Salida).

A partir de ahora las tablas sobre las que se ejecutarán las siguientes etapas serán las generadas por la Salida de esta etapa.

Etapa de DATA SELECTION:
(si esta etapa no es elegida en primer término, sus tablas de entrada serán el resultado de la Salida de la última etapa ejecutada anteriormente).

Se le brindará al usuario una descripción más detallada de lo que significa esta etapa y la importancia de su ejecución.

La herramienta mostrará todos los conceptos y para cada uno de ellos todas sus propiedades definidas en la fase anterior del estudio, para que el usuario pueda elegir quedarse con algún subgrupo de ellas y resumir de esta manera el universo de trabajo. También dará al usuario la posibilidad de elegir qué salida desea obtener con esa selección, dándole las siguientes opciones:

- Armar tablas nuevas que posean sólo los atributos seleccionados. La herramienta aconsejará optar por esta salida. Salida = *Tbl.
- Armar una vista de las tablas con los atributos seleccionados. Salida = VistaPrc.

La herramienta traducirá al formato definido en el KDD language, los select definidos por el usuario.

Quedan generados:
select (BD.Tbl, [atributo1, ..., atributon], Salida).

A partir de ahora las tablas sobre las que se ejecutarán las siguientes etapas serán las generadas por la Salida de esta etapa.

Etapa de CLEANING:
(si esta etapa no es elegida en primer término, sus tablas de entrada serán el resultado de la Salida de la última etapa ejecutada anteriormente).

Se le brindará al usuario una descripción más detallada de lo que significa esta etapa y la importancia de su ejecución.

La herramienta irá mostrando al usuario uno a uno todos los conceptos definidos y para cada uno de ellos todas sus propiedades, entonces le irá consultando si desea limpiarlo.

Si el usuario elige no limpiarlo pasará a la propiedad o concepto siguiente.

Si el usuario opta por la limpieza deberá indicar los valores incorrectos de esa propiedad.

La herramienta informará al usuario qué porcentaje de registros limpios encontró y entonces el usuario decidirá si acepta esa limpieza. En el caso de ser aceptada, deberá pedir la salida deseada:

- Armar vistas de la base de datos real con los registros con datos válidos. Salida = VistaPrc.
- Reemplazar por valor "null" los datos inválidos. Salida = NullPrc.

La herramienta buscará para el concepto solicitado, a través de la herencia, las relaciones y el diccionario de datos definidos, los datos en la base de datos real y recorrerá todos los valores de ese concepto, limpiándolo y generando la salida pedida.

La herramienta traducirá al formato definido en el KDD language, los cleaning definidos por el usuario.

Quedan generados:
clean (BD.Tbl, [cond1, ..., condn], Salida).

A partir de ahora las tablas sobre las que se ejecutarán las siguientes etapas serán las generadas por la Salida de esta etapa.

Etapa de ENRICHMENT:

(si esta etapa no es elegida en primer término, sus tablas de entrada serán el resultado de la Salida de la última etapa ejecutada anteriormente).

Se le brindará al usuario una descripción más detallada de lo que significa esta etapa y la importancia de su ejecución.

La herramienta le consultará al usuario si desea completar los datos que posee con otros correspondientes a otras bases de datos. Para que pueda elegir nuevos atributos, se le irán presentando todos los posibles, teniendo en cuenta que las claves de ambas tablas, enriquecedora y a enriquecer, deben coincidir.

La salida de esta etapa será el agregado de los atributos mencionados a la tabla vigente o la generación de una tabla nueva. Salida = *Tbl.

enrich (BD.Tbl1.atributo_clave, BD.Tbl2, [atributo1, ..., atributon], Salida).

A partir de ahora las tablas sobre las que se ejecutarán las siguientes etapas serán las generadas por la Salida de esta etapa.

Etapa de CODING:
(si esta etapa no es elegida en primer término, sus tablas de entrada serán el resultado de la Salida de la última etapa ejecutada anteriormente).

Se le brindará al usuario una descripción más detallada de lo que significa esta etapa y la importancia de su ejecución.

La herramienta consultará al usuario sobre los conceptos definidos para saber si desea codificarlos, es decir, dividir sus valores en grupos.

La herramienta traducirá al formato definido en el KDD language, los cleaning definidos por el usuario.

Quedan generados:
code (BD.Tbl, atributo, [cond1, ..., condn], Salida).

La salida será la generación de nuevas tablas con los datos divididos en grupos según solicitó el usuario. Salida = *Tbl.

Luego, la herramienta recodificará automáticamente asignando números a los diferentes n conjuntos de datos, comenzando por el 1 hasta n.

Etapa de DATA MINING:
(sus tablas de entrada serán el resultado de la salida de la última etapa ejecutada anteriormente)

Si esta etapa es elegida en primer término, se informará al usuario que no ha especificado acciones de preparación de los datos, explicándole las desventajas de no hacerlo y todas las posibles acciones que podría llevar a cabo para optimizar los resultados de su investigación.

Aquí es cuando deberá definir el objetivo de su estudio, deberá elegir qué método de data mining desea aplicar y bajo qué condiciones, para ello le serán mencionados todos los posibles y una breve síntesis del accionar de cada uno de ellos, de manera tal que el usuario pueda discernir si para su estudio le conviene obtener reglas, secuencias, árboles de decisión, agrupamientos de los datos, etc..

Dentro de los métodos de posible aplicación, le serán expuestos al usuario, los correspondientes a cada una de las etapas del kdd, puesto que podría utilizar esta herramienta para realizar alguna de esas tareas, sin pretender el mining de los datos. Podría por ejemplo, usarse solamente para limpiar los datos.

Se generará la siguiente primitiva:
objetivo (Metodo, **[**Cond1, ..., Condn **]**, Estructura **).**

Etapa de REPORTING:

A esta etapa se llega con todas las acciones ya realizadas.

Toda la estructura generada por la aplicación del algoritmo de data mining elegido, constituirá la entrada de esta etapa y en función del formato de salido elegido por el usuario, se elaborarán los reportes de salida.

Se le enseñarán al usuario posibles formas de representación de los resultados, tablas y gráficos, para que éste elija la más conveniente para su posterior análisis.

Internamente se generará la siguiente especificación:
report (Entrada, Salida **).**

Una vez especificada la etapa de reporting, se generará la compilación de todo lo especificado y finalmente, a partir de la invocación de la ejecución por parte del usuario, comenzará a correr toda la herramienta, ejecutándose de manera sucesiva todas las acciones correspondientes a las especificaciones realizadas.

7.- DESARROLLO DE EJEMPLOS TIPICOS

A continuación se desarrollarán dos ejemplos de aplicación del KDD Language a los efectos de mostrar su funcionamiento completo en situaciones de la realidad y poder comparar la manera de encarar los estudios así como también los beneficios que implica su uso.

En el primer caso, se trata de un ejemplo abordado en [Adr96], en su capítulo 4 denominado "The Knowledge Discovery Process". En dicho libro se lo utiliza para enseñar con un caso concreto y consistente, todo el proceso de extracción de conocimiento de una base de datos, las diferentes etapas y los resultados y cambios que se van generando en los datos al ejecutar cada una de ellas.

El segundo ejemplo estará basado en [Sri95], donde se aplican algoritmos de Association Rules sobre una tabla de transacciones de ítems vendidos en un supermercado.

7.1. APLICACIÓN DEL KDD LANGUAGE SOBRE LA BASE DE DATOS DE UN NEGOCIO DE VENTA DE SUSCRIPCIONES DE REVISTAS

Se trata de un negocio de venta de suscripciones de revistas que comenzó a funcionar en enero de 1990. Su nombre: "LEER MAS"

Este negocio ofrece la suscripción a 5 tipos de revistas: de historietas, de autos, de música, del hogar y deportivas.

Las revistas tienen su respectivo título o nombre y son editadas por una editorial y provistas a medida que son requeridas por los clientes a través de las suscripciones.

Los clientes tienen un número de código que los individualiza.

La empresa cuenta con una base de datos, Leer Más, que organizó y mantiene un asistente informático.

En el área de marketing decidieron realizar un estudio profundo sobre las vinculaciones existentes entre los clientes y los tipos de revistas que suscriben, generando secuencias de suscripciones de revistas a lo largo del tiempo.

Se mostrará a continuación cómo resultaría la aplicación del KDD Language en dicho estudio.

ORGANIZACIÓN DE LA BASE DE DATOS "LEER MAS"

Tabla: **REVISTAS**
Atributos: **CodigoRevista - NombreRevista - Autor - Editorial - Tipo - Precio**

Tabla: **EDITORIALES**
Atributos: **NumeroEditorial - NombreEditorial - Direccion - Te**

Tabla: **CLIENTES**
Atributos: **NumeroCliente - NombreCliente - Direccion - Te**

Tabla: **SUSCRIPCIONES**
Atributos: **NumeroSuscripcion - NombreCliente - NombreRevista - PrecioSuscripcion - FechaSuscripcion**

NOMBRE DEL ESTUDIO A REALIZAR: "ANALISIS CLIENTES"

CONCEPTOS:

concepto (revista).
concepto (suscripcion).
concepto (proveedor).
concepto (cliente).

PROPIEDADES:

propiedades (revista, [codigo, titulo, autor, tipo, precio, editorial, proveedor]).
propiedades (cliente, [numero, nombre, direccion, te]).
propiedades (proveedor, [numero, nombre, direccion, te]).
propiedades (suscripcion, [numero, fecha, revista, cliente]).

RELACIONES:

relacion (proveen, proveedor, revista).
relacion (realiza, cliente, suscripcion).
relacion (vende, suscripcion, revistas).

HERENCIAS:

herencia (agregacion, cliente, realiza).
herencia (agregacion, suscripcion, vende).
herencia (agregacion, proveedor, provee).

DICCIONARIO:

diccionario (revista, codigo, $ LeerMas.Revistas.CodigoRevista $ **).**
diccionario (revista, titulo, $ LeerMas.Revistas.NombreRevista $ **).**
diccionario (revista, autor, $ LeerMas.Revistas.Autor $ **).**
diccionario (revista, tipo, $ LeerMas.Revistas.Tipo $ **).**
diccionario (revista, precio, $ LeerMas.Revistas.Precio $ **).**
diccionario (revista, editorial, $ LeerMas.Revistas.Editorial $ **).**
diccionario (revista, proveedor, $ LeerMas.Revistas.Editorial $ **).**

diccionario (cliente, numero, $ LeerMas.Clientes.NumeroCliente $ **).**
diccionario (cliente, nombre, $ LeerMas.Clientes.NombreCliente $ **).**
diccionario (cliente, direccion, $ LeerMas.Clientes.Direccion $ **).**
diccionario (cliente, te, $ LeerMas.Clientes.Te $**).**

diccionario (proveedor, numero, $ LeerMas.Editoriales.NumeroEditorial $ **).**
diccionario (proveedor, nombre, $ LeerMas.Editoriales.NombreEditorial $ **).**
diccionario (proveedor, direccion, $ LeerMas.Editoriales.Direccion $ **).**
diccionario (proveedor, te, $ LeerMas.Editoriales.Te $ **).**

diccionario (suscripcion, numero, $ LeerMas.Suscripciones.NumeroSuscripcion $ **).**
diccionario (suscripcion, fecha, $ LeerMas.Suscripciones.FechaSuscripcion $ **).**
diccionario (suscripcion, revista, $ LeerMas.Suscripciones.NombreRevista $ **).**
diccionario (suscripcion, cliente, $ LeerMas.Suscripciones.NombreCliente $ **).**

ACCIONES:

El cliente decide realizar en primer término la selección de datos, eligiendo como salida la creación de una tabla nueva conteniendo los datos seleccionados. Esta tabla nueva es llamada "Nueva".

select ($LeerMas.Suscripciones$, **[** $NombreCliente$, $FechaSuscripcion$, $NombreRevista$ **]**, $NuevaTbl$ **).**

select ($LeerMás.Clientes$, **[** $ NumeroCliente $, $ NombreCliente $, $ Direccion $ **]**, $ NuevaTbl $ **).**
select ($ LeerMas.Revistas $, **[** $ NombreRevista $, $ Tipo $ **]**, $ NuevaTbl $ **).**

El resultado de la ejecución de esta etapa es la tabla "Nueva", cuyos atributos son:
NombreCliente, FechaSuscripcion, NombreRevista, NumeroCliente, Direccion, Tipo.

Luego decide realizar la limpieza de los datos, optando por reemplazar por "null" los valores incorrectos. Esto lo realiza el procedimiento NullPrc.

Como la entrada de esta etapa es la salida de la ejecución anterior, la limpieza se realizará sobre la tabla NuevaTbl.

clean ($ LeerMas.NuevaTbl $**, [** $ FechaSuscripcion $, "<", $ 01-01-90 $ **],** %**,** "NullPrc" **).**
clean ($ LeerMas.NuevaTbl $**, [** $ NumeroCliente $, " >", $ 23100 $ **],** %**,** "NullPrc" **).**

El resultado de esta ejecución es la tabla NuevaTbl, con valores null en las fechas y número de cliente inválidos según los criterios definidos.

Ahora el usuario quiere enriquecer su tabla NuevaTbl con datos ajenos a su negocio, agregará a su tabla la fecha de nacimiento de sus clientes, sus ingresos, ahorros y si son dueños de auto y/o casa.

enrich ($LeerMas.NuevaTbl.NombreCliente$**, [** $Nacimiento$, $Ingresos$, $Ahorros$, $Auto$, $Casa$ **],** $NuevaTbl$ **).**

El resultado será el agregado de los atributos recién definidos a la tabla NuevaTbl.

Como se agregaron datos que pueden estar sucios, se decide limpiar la fecha de nacimiento de los clientes, invocándose para ello nuevamente la acción clean.

clean ($LeerMas.NuevaTbl$, **[** $Nacimiento$, "<", $01-01-20$ **],** %, "NullPrc" **).**

El usuario puede elegir qué atributos aceptar con valor null y cuáles no, para ello ejecuta la acción de filtrado.

filter ($LeerMas.NuevaTbl$, **[** $Nacimiento$, "=", $null$ **]**, "Nueva1Tbl" **).**

filter ($LeerMas.NuevaTbl$, **[** $Ingresos$, "=", $null$ **]**, "Nueva1Tbl" **).**

filter ($LeerMas.NuevaTbl$, **[** $Ahorros, "=", $null$ **]**, "Nueva1Tbl" **).**

filter ($LeerMas.NuevaTbl$, **[** $Auto$, "=", $null$ **]**, "Nueva1Tbl" **).**

filter ($LeerMas.NuevaTbl$, **[** $Casa$, "=", $null$ **]**, "Nueva1Tbl" **).**

Esta ejecución armará una nueva tabla: Nueva1Tbl conformada por los registros de la anterior, Nueva, pero sin aquellos registros con valor null en los atributos Nacimiento, Ingresos, Ahorros, Auto ó Casa.

También podría completarse la limpieza con la convocatoria a un algoritmo que elimine registros que contengan igual o mayor cantidad que "n" nulls, con n definido por el usuario, sin importar a cuáles atributos correspondan.

En este momento el usuario cuenta con la tabla Nueva1Tbl, que contiene los datos por él seleccionados, enriquecidos como él quiso y limpios de acuerdo a los criterios de limpieza que él eligió. En este momento decide codificarlos para poder manipularlos y comprenderlos mejor. Es así que convoca a la ejecución del coding.

Le resultará más sencillo trabajar con la edad de los clientes que con sus fechas de nacimiento, es por ello que el procedimiento EdadPrc transformará ese dato.

code ($ LeerMas.Nueva1Tbl $, $ Nacimiento $**, [** "EdadPrc" **],** $ Nueva2Tbl $ **).**

En cuanto a la dirección, le resulta más útil dividir el mapa de la ciudad en áreas o regiones y codificar cada domicilio de acuerdo a la zona que corresponda. Lo hará el procedimiento RegionesPrc.

code ($LeerMas.Nueva1Tbl$**,** $Direccion$**, [** "RegionesPrc" **],** $Nueva2Tbl$ **).**

Con la fecha de suscripción podrá obtener la antigüedad de cada cliente, esto le servirá luego para establecer series de tiempo. La idea es transformar cada fecha en un número de mes de antigüedad, contando a enero del 90' como el n° 1, febrero como el n° 2, y así sucesivamente. Todo esto lo hará el procedimiento AntigüedadPrc.

code ($LeerMas.Nueva1Tbl$**,** $Fecha$**, [** "AntiguedadPrc" **],** $Nueva2Tbl$ **).**

Con respecto a los ingresos y los ahorros, le resultará más sencillo para trabajar el tenerlos expresados en miles de pesos, con lo cual debe dividir todos esos datos, para lo cual convoca a los procedimientos Div1000Prc.

code ($LeerMas.Nueva1Tbl$**,** $Ingresos$**, [**"Div1000Prc"**],** $Nueva2Tbl$ **).**

code ($LeerMas.Nueva1Tbl$, $Ahorros$, ["Div1000Prc"], $Nueva2Tbl$).

En cuanto a la posesión de casa o auto, convendrá transformar el sí por 1 y el no por 2.

code ($LeerMas.Nueva1Tbl$, $Auto$, [si, "=", 1, no, "=", 2], $Nueva2Tbl$).

code ($LeerMas.Nueva1Tbl$, $Casa$, [si, "=", 1, no, "=", 2], $Nueva2Tbl$).

Llegó el momento de la especificación del método de data mining elegido, para ello es evocada la función objetivo:

objetivo ("sequential patterns", ["sp", ">", 0.25], Estructura).

Cuando se ejecute la herramienta, será convocado el algoritmo de sequential patterns que posea la misma y será aplicado sobre la tabla Nueva2, que fue la última generada en este proyecto.

La aplicación del algoritmo de data mining generará una estructura de salida acorde al algoritmo seleccionado, la cual será tomada por la función report (Entrada, Salida)., elaborando los reportes correspondientes y sobre los formatos seleccionados por el usuario.

A través de la aplicación del algoritmo elegido, se generará la estructura resultado que será del tipo:

Estructura = (secuencia [[auto], [hogar], 0.50]).
(secuencia [[musica], [auto], 0.40]).
(secuencia [[deportiva], [auto], 0.30]).
(secuencia [[historieta], [musica], 0.40]).

Quedaron establecidas las secuencias de suscripciones de revistas a lo largo del tiempo. El 50% de los clientes suscriben revistas de automovilismo y tiempo después suscriben revistas del hogar y de esta manera se pueden leer el resto de las secuencias generadas.

Luego se ejecutará la acción report ( )., y quedará generado el reporte que recibirá el usuario, pudiendo ser por ejemplo, de la siguiente forma:

Secuencias temporales que han superado el mínimo soporte requerido: 0.25

Soporte	1° suscripción	2° suscripción
0.5	auto	hogar
0.4	música	auto
0.4	historieta	musica
0.3	deportiva	auto

7.2.- APLICACIÓN DEL KDD LANGUAGE SOBRE LA BASE DE DATOS DE UN SUPERMERCADO.

Se trata del Supermercado "SUR" en el cual se venden diversos productos clasificados por categorías: comestibles, electrodomésticos, ropa, bazar, perfumería y limpieza.

Dentro de los comestibles se venden carnes, verduras, frutas y envasados (son los comestibles no frescos: yerba, fideos, arroz, café, etc.). Las carnes se dividen en vaca, pollo, cerdo y pescado.

Mensualmente se ofrecen diferentes ofertas que van rotando por rubros.

Los productos son adquiridos a diferentes proveedores de acuerdo al producto.

En el supermercado trabajan empleados encargados de diversas tareas: administrativos, repositores, cajeras, limpieza y seguridad.

Existe un área de sistemas encargada de organizar y mantener actualizados todos los datos vinculados al negocio en una base de datos: SUR

Se realizará la representación en el KDD Language de un estudio de marketing con el objetivo de preparar los datos y convocar el accionar de un algoritmo de data mining, "mining generalized association rules", a los efectos de analizar los conjuntos de productos que se vendieron simultáneamente durante el último mes de octubre, investigando la implicancia que tuvo la venta de determinados productos sobre la venta de otros, teniendo en cuenta el árbol de taxonomías existentes de manera tal de permitir un estudio que cruce los diferentes niveles desde las hojas hasta su raíz.

ORGANIZACIÓN DE LA BASE DE DATOS: SUR

Tabla: **Productos**
Atributos: **Codigo – Nombre – Categoria – Existencia – Costo - PrecioVenta, Proveedor**

Tabla: **Proveedores**
Atributos: **Numero – Nombre – Direccion – Te – FormaPago – PlazoPago**

Tabla: **Comestibles**
Atributos: **Codigo – Tipo - Descripcion**

Tabla: **Carnes**
Atributos: **Codigo – Animal – Corte**

Tabla: **Electrodomesticos**
Atributos: **Codigo – Nombre - Marca - Modelo**

Tabla: **Ropa**
Atributos: **Codigo - Nombre - Marca – Modelo**

Tabla: **Bazar**
Atributos: **Codigo – Nombre - Marca**

Tabla: **Perfumeria**
Atributos: **Codigo – Nombre - Marca**

Tabla: **Limpieza**
Atributos: **Codigo – Nombre - Marca**

Tabla: **Ventas**
Atributos: **Fecha - NumeroVenta - CodigoProducto – Nombre - Cantidad – Importe**

Tabla: **Empleados**
Atributos: **Legajo – Nombre – Sector – Direccion - Te - Salario – FechaNacimiento - DNI – CalificacionAnual – HsTrabajo**

Tabla: **Ofertas**
Atributos: **Mes - Rubro - CodigoProducto – Nombre – PrecioVenta**

NOMBRE DEL ESTUDIO A REALIZAR:
"VENTAS DE PRODUCTOS"

CONCEPTOS:

concepto (producto).
concepto (comestible).
concepto (electrodomestico).
concepto (ropa).
concepto (prodperf).
concepto (prodbazar).
concepto (prodlimp).
concepto (proveedores).
concepto (provcomest).
concepto (provelectrodom).
concepto (provropa).
concepto (provperfume).
concepto (provbazar).
concepto (provlimp).
concepto (ventas).
concepto (sectorvta). *# es el lugar donde se vende: almacén, carnicería, etc.*
concepto (carne).
concepto (verdura).
concepto (fruta).
concepto (paquete). *# son los comestibles no frescos: yerba, fideos, café, etc.*
concepto (vaca).
concepto (pollo).
concepto (cerdo).
concepto (pescado).
concepto (oferta).

concepto (empleados).
concepto (empadmin).
concepto (repositor).
concepto (cajera).
concepto (emplimp).
concepto (empseguridad).

PROPIEDADES:

propiedades (producto, **[** codigo, nombre, categoria, proveedor, costo, preciovta, existencia **]).**

propiedades (comestible, **[** codigo, nombre, proveedor, costo, preciovta, existencia **]).**

propiedades (electrodomestico, **[** codigo, nombre, marca, modelo, proveedor, costo, preciovta, existencia **]).**

propiedades (ropa, **[** codigo, nombre, marca, proveedor, costo, preciovta., existencia **]).**

propiedades (prodperf, **[**codigo, nombre, marca, proveedor, costo, preciovta, existencia **]).**

propiedades (prodbazar, **[**codigo, nombre, marca, proveedor, costo, preciovta, existencia **]).**

propiedades (prodlimp, **[**codigo, nombre, marca, proveedor, costo, preciovta, existencia **]).**

propiedades (carne, **[** codigo, animal, corte, proveedor, costo, preciovta, existencia **]).**

propiedades (vaca, **[** codigo, corte, proveedor, costo, preciovta, existencia **]).**

propiedades (pollo, **[** codigo, corte, proveedor, costo, preciovta, existencia **]).**

propiedades (cerdo, **[** codigo, corte, proveedor, costo, preciovta, existencia **]).**

propiedades (pescado, **[** codigo, corte, proveedor, costo, preciovta, existencia **]).**

propiedades (verdura, **[** codigo, nombre, proveedor, costo, preciovta, existencia **]).**

propiedades (fruta, **[** codigo, nombre, proveedor, costo, preciovta, existencia **]).**

propiedades (paquete, **[** codigo, nombre, proveedor, costo, preciovta, existencia **]).**

propiedades (proveedores, **[** nombre, direccion, telefono, formapago, plazopago, producto **]).**

propiedades (provcomest, **[**nombre, direccion, telefono, formapago, plazopago, producto **]).**

propiedades (provelectrodom, **[**nombre, direccion, telefono, formapago, plazopago, producto **]).**

propiedades (provropa, [nombre, direccion, telefono, formapago, plazopago, producto]).

propiedades (provperfume, [nombre, direccion, telefono, formapago, plazopago, producto]).

propiedades (provbazar, [nombre, direccion, telefono, formapago, plazopago, producto]).

propiedades (provlimp, [nombre, direccion, telefono, formapago, plazopago, producto]).

propiedades (ventas, [fechvta, numerovta, codigoproducto, nombre, cantidad, importe]).

propiedades (sectorvta, [nombre, producto]).

propiedades (empleado, [legajo, nombre, direccion, telefono, fechnac, dni, lugartrab, califanual, hstrab, salario]).

propiedades (empadmin, [legajo, nombre, direccion, telefono, fechnac, dni, lugartrab, tarea, califanual, hstrab, salario]).

propiedades (repositor, [legajo, nombre, direccion, telefono, fechnac, dni, lugartrab, tarea, califanual, hstrab, salario]).

propiedades (cajera, [legajo, nombre, direccion, telefono, fechnac, dni, lugartrab, tarea, califanual, hstrab, salario]).

propiedades (emplimp, [legajo, nombre, direccion, telefono, fechnac, dni, lugartrab, tarea, califanual, hstrab, salario]).

propiedades (empseg, [legajo, nombre, direccion, telefono, fechnac, dni, lugartrab, tarea, califanual, hstrab, salario]).

propiedades (oferta, [mes, categoria, codigo, nombre, preciovta]).

RELACIONES:

relacion (esun, comestible, producto).

relacion (esun, electrodomestico, producto).

relacion (esun, ropa, producto).

relacion (esun, prodperf, producto).

relacion (esun, prodbazar, producto).

relacion (esun, prodlimp, producto).

relacion (esun, carne, comestible).

relacion (esun, verdura, comestible).

relacion (esun, , fruta, comestible).

relacion (esun, paquetes, comestible).

relacion (es, vaca, carne).

relacion (es, pollo, carne).

relacion (es, cerdo, carne).

relacion (es, pescado, carne).

relacion (esun, provcomest, proveedores).

relacion (esun, provelectrodom, proveedores).

relacion (esun, provropa, proveedores).

relacion (esun, provperfume, proveedores).

relacion (esun, provbazar, proveedores).

relacion (esun, provlimp, proveedores).

relacion (proveen, proveedores, productos).

relacion (sevenden, producto, sectorvta.).

relacion (esun, empadm, empleado).

relacion (esun, repositor, empleado).

relacion (esun, cajera, empleado **).**

relacion (esun, emplimp, empleado **).**

relacion (esun, empseg, empleado **).**

relacion (asignadoa, repositor, sectorvta **).**

relacion (asignadoa, emplimp, sectorvta **).**

relacion (estanen, productos, oferta **).**

HERENCIAS:

herencia (agregacion, comestible, esun).

herencia (agregacion, electrodomestico, esun).

herencia (agregacion, ropa, esun).

herencia (agregacion, prodperf, esun).

herencia (agregacion, prodbazar, esun).

herencia (agregacion, prodlimp, esun).

herencia (agregacion, carne, esun).

herencia (agregacion, verdura, esun).

herencia (agregacion, fruta, esun).

herencia (agregacion, paquetes, esun).

herencia (agregacion, vaca, esun).

herencia (agregacion, pollo, esun).

herencia (agregacion, cerdo, esun).

herencia (agregacion, pescado, esun).

herencia (agregacion, proveedores, proveen).

herencia (agregacion, provcomest, esun).

herencia (agregacion, provelectrodom, esun).

herencia (agregacion, provropa, esun).

herencia (agregacion, provperfume, esun).

herencia (agregacion, provbazar, esun).

herencia (agregacion, provlimp, esun).

herencia (agregacion, empadm, esun).

herencia (agregacion, repositor, esun).

herencia (agregacion, cajera, esun).

herencia (agregacion, emplimp, esun **).**

herencia (agregacion, empseg, esun **).**

herencia (sustitucion, producto, estanen **).**

DICCIONARIO DE DATOS:

diccionario (producto, codigo, $Sur.Productos.Codigo$).
diccionario (producto, nombre, $Sur.Productos.Nombre$).
diccionario (producto, categoria, $Sur.Productos.Categoria$).
diccionario (producto, proveedor, $Sur.Productos.Proveedor$).
diccionario (producto, costo, $Sur.Productos.Costo$).
diccionario (producto, preciovta, $Sur.Productos.PrecioVenta$).
diccionario (producto, existencia, $Sur.Productos.Existencia$).

diccionario (comestible, codigo, $Sur.Comestibles.Codigo$).

diccionario (electrodomesticos, codigo, $Sur.Electrodomesticos.Codigo$).
diccionario (electrodomesticos, nombre, $Sur.Electrodomesticos.Nombre$).
diccionario (electrodomesticos, marca, $Sur.Electrodomesticos.Marca$).
diccionario (electrodomesticos, modelo, $Sur.Electrodomesticos.Modelo$).

diccionario (ropa, codigo, $Sur.Ropa.Codigo$).
diccionario (ropa, nombre, $Sur.Ropa.Nombre$).
diccionario (ropa, marca, $Sur.Ropa.Marca$).

diccionario (prodperf, codigo, $Sur.Perfumeria.Codigo$).
diccionario (prodperf, nombre, $Sur.Perfumeria.Nombre$).
diccionario (prodperf, marca, $Sur.Perfumeria.Marca$).

diccionario (prodbazar, codigo, $Sur.Bazar.Codigo$).
diccionario (prodbazar, nombre, $Sur.Bazar.Nombre$).
diccionario (prodbazar, marca, $Sur.Bazar.Marca$).

diccionario (prodlimp, codigo, $Sur.Limpieza.Codigo$).
diccionario (prodlimp, nombre, $Sur.Limpieza.Nombre$).
diccionario (prodlimp, marca, $Sur.Limpieza.Marca$).

diccionario (carne, codigo, $Sur.Carnes.Codigo$).
diccionario (carne, animal, $Sur.Carnes.Animal$).
diccionario (carne, corte, $Sur.Carnes.Corte$).

diccionario (proveedores, nombre, $Sur.Proveedores.Nombre$).
diccionario (proveedores, direccion, $Sur.Proveedores.Direccion$).
diccionario (proveedores, telefono, $Sur.Proveedores.Te$).
diccionario (proveedores, formapago, $Sur.Proveedores.FormaPago$).
diccionario (proveedores, plazopago, $Sur.Proveedores.PlazoPago$).

diccionario (ventas, fechvta, $Sur.Ventas.Fecha$).
diccionario (ventas, numerovta, $Sur.Ventas.NumeroVenta$).
diccionario (ventas, producto, $Sur.Ventas.CodigoProducto$).
diccionario (ventas, nombre, $Sur.Ventas.Nombre$).
diccionario (ventas, cantidad, $Sur.Ventas.Cantidad$).
diccionario (ventas, importe, $Sur.Ventas.Importe$).

diccionario (empleado, legajo, $Sur.Empleados.Legajo$ **).**
diccionario (empleado, nombre, $Sur.Empleados.Nombre$ **).**
diccionario (empleado, direccion, $Sur.Empleados.Direccion$ **).**
diccionario (empleado, telefono, $Sur.Empleados.Te$ **).**
diccionario (empleado, fechnac, $Sur.Empleados.FechaNacimiento$ **).**
diccionario (empleado, dni, $Sur.Empleados.DNI$ **).**
diccionario (empleado, lugartrab, $Sur.Empleados.Sector$ **).**
diccionario (empleado, califanual, $Sur.Empleados.CalificaciónAnual$ **).**
diccionario (empleado, hstrab, $Sur.Empleados.HsTrabajo$ **).**
diccionario (empleado, salario, $Sur.Empleados.Salario$ **).**

diccionario (oferta, mes, $Sur.Ofertas.Mes$ **).**
diccionario (oferta, categoría, $Sur.Ofertas.Rubro$ **).**
diccionario (oferta, codigo, $Sur.Ofertas.CodigoProducto$ **).**
diccionario (oferta, nombre, $Sur.Ofertas.Nombre$ **).**
diccionario (oferta, preciovta, $Sur.Ofertas.PrecioVenta$ **).**

ACCIONES:

Sólo interesan las ventas del mes de octubre, por lo tanto es necesario filtrar todas aquellas que no correspondan a dicho mes. Con esos datos se decide armar una nueva tabla llamada VentasOctubreTbl.

filter ($Ventas$, [$Fecha$, "<>", $*-10-98$], $VentasOctubreTbl$ **).**

El resultado de la ejecución del filter será la tabla VentasOctubreTbl con los mismos atributos que la tabla Ventas pero con los registros correspondientes a octubre de 1998.

Es necesario seleccionar con qué atributos se trabajará y para ello es convocada la ejecución del select.

select ($VentasOctubreTbl$, **[** $NumeroVenta$, $Nombre$ **]**, $TransacOctubreTbl$ **).**

El resultado de la ejecución del select será la tabla TransacOctubreTbl cuyos valores serán los correspondientes a los atributos NumeroVenta y Nombre de la tabla VentasOctubreTbl.

Como próxima etapa se realizará la limpieza de los datos de la tabla de transacciones y se pedirá una vista de la misma para decidir si se efectúa la limpieza o no. Se consideran datos válidos a los número de ventas entre 1000 y 2000 y a los código de productos menores que 20000.

clean ($TransacOctubreTbl$, **[[**1000, ">", $NumeroVenta$, ">", 2000 **],** %**,** "VistaPrc" **).**

Como la vista representa un porcentaje aceptable de la tabla total, se decide efectuar la limpieza y para ello se procede a reemplazar los valores inválidos por "null", convocando nuevamente a la acción de clean.

clean ($TransacOctubreTbl$, **[[** 1000, ">", $NumeroVenta$, ">", 2000 **],** %**,** "NullPrc" **).**

Por último se procede a la eliminación de los registros con valores "null", lo cual es realizado por la acción filter.

filter ($TransacOctubreTbl$, **[[**$NumeroVenta$, "=", $null$ **]**, $ListaTbl$ **).**

Quedó conformada la tabla ListaTbl, con los datos necesarios y limpios para efectuar el data mining.

Es importante notar que no fueron ejecutadas todas las etapas definidas en la interfaz del KDD Language. Esto demuestra que es importante permitir al usuario la

elección de las acciones y el orden en que desea ejecutarlas para llegar al objetivo deseado.

Se procede entonces a la definición de la función objetivo, para convocar la aplicación del algoritmo de mining generalized association rules y obtener las reglas correspondientes a las implicancias de los productos vendidos simultáneamente durante octubre de 1998.

objetivo ("MiningGeneralizedAssociationRules", [conclusion, [$ListaTbl.Nombre$], ["sp", "=", 0.6], ["cf", "=", 0.4]], Estructura).

El algoritmo generará las reglas correspondientes a los productos vendidos y como no se trata del algoritmo standard de association rules, sino que tiene en cuenta las taxonomías existentes en los conceptos, deberá reconstruirlas a partir de las relaciones definidas en el dominio.

Quedarán generadas reglas cuyo soporte supere el 0.6 y cuya confianza sea mayor que el 0.4. El consecuente de las reglas estará conformado por los nombres de los productos vendidos en cada transacción o por sus antecesores, de acuerdo a las relaciones establecidas. El algoritmo pivoteará sobre el atributo mencionado como conclusión.

Un ejemplo de la estructura de salida podría ser el siguiente:

```
Estructura = ( regla [ [ cuadril ], [ tomate ], 0.65, 0.48 ] ).
             ( regla [ [ lechuga ], [ tomate ], 0.65, 0.50 ] ).
             ( regla [ [ jabon ], [ tomate ], 0.70, 0.50 ] ).
             ( regla [ [ desodorante ], [ tomate ], 0.70, 0.55 ] ).
             ( regla [ [ yerba ], [ azucar ], 0.72, 0.50 ] ).
             ( regla [ [ platos ], [ azucar ], 0.77, 0.55 ] ).
             ( regla [ [ manzanas, jabon ], [ azucar ], 0.80, 0.58 ] ).
             ( regla [ [ lomo, yerba ], [ azucar ], 0.72, 0.50 ] ).
             .....
             ( regla [ [ prodperf ], [ verdura ], 0.72, 0.55 ] ).
             ( regla [ [ carne ], [ electrodom ], 0.80, 0.55 ] ).
             ( regla [ [ prodbazar ], [ comestible ], 0.80, 0.57 ] ).
             ( regla [ [ prodlimp ], [ comestible ], 0.85, 0.55 ] ).
```

Esta estructura será tomada como entrada por la función report ( )., la cual generará un reporte compresible para el usuario acorde al formato por él especificado en dicha acción.

Un ejemplo de dicho reporte podría ser el siguiente:

A partir de la aplicación del algoritmo de association rules, fueron generadas las siguientes reglas, con soporte y confianza superiores a los requeridos, 0.6 y 0.4 respectivamente:

Si los clientes que compraron:	también compraron:
cuadril	tomate
lechuga	tomate
jabon	tomate
desodorante	tomate
yerba	azucar
platos	azucar
manzana, jabon	azucar
lomo, yerba	azucar
...	
prodperf	verdura
carne	electrodomesticos
prodbazar	comestibles
prodlimp	comestibles

Basándose en la misma especificación de dominios, correspondiente al supermercado SUR, podría encararse otro estudio, y querer investigar por ejemplo, cómo influyen las ofertas en las ventas globales de un producto.

Se analizarán las ventas de tomate. Se sabe que los tomates estuvieron en oferta durante julio de 1998. Se desea averiguar si el total vendido en ese mes se debe a su condición de ofertados o si se hubiera vendido la misma cantidad en caso contrario. Para enriquecer el estudio, se analizará la influencia que tuvo la oferta de lechuga del mes anterior.

Para llegar a una conclusión sólida, se obtendrán las ventas de tomates de julio de 1998 y se las comparará con las siguientes:

- junio de 1998, mes en que estaba en oferta la lechuga
- agosto de 1998, mes cercano a julio y sin ofertas
- julio de 1997, igual mes del año anterior y sin ofertas

A continuación se describirán las acciones efectuadas para obtener la tabla de cantidades e importes de las ventas de tomates efectuadas durante el mes de julio de 1998, mes en que estuvieron en oferta los tomates:

filter ($Sur.Ventas$**, [[** $fecha$, "<>", $*-07-98$ **], [** $nombre$, "<>", $tomate$ **]],** $Tojul98Tbl$ **).**

filter ($Sur.Ofertas98$**, [** mes, "<>", $julio$ **], [** $nombre$, "<>", $tomate$ **]],** $Ofjul98Tbl$ **).**

select ($Tojul98Tbl$, **[** $cantidad$, $importe$ **],** $Tojul98Tbl$ **).**

select ($Ofjul98Tbl$, **[** $precio_de_venta$ **]).**

Se decide limpiar los posibles errores de tipeo correspondientes a los importes de venta, para lo cual se define la función clean.

clean ($Tojul98Tbl$, **[** $importe$, "=", "CalculoPrc" **]**, %, "VistaPrc" **).**

CalculoPrc = se trata de un procedimiento que calcula lo siguiente: Tojul98Tbl.cantidad * 0.55 (precio del kg. de tomates durante julio98, obtenido de Ofjul98Tbl.precio_de_venta.

Se supone que el resultado % fue 99%. Se lo considera bueno, como para no realizar más cleaning.

Se obtuvo la tabla Tojul98Tbl que posee las cantidades y los importes de los kg. de tomates vendidos durante el mes de julio de 1998. Con los valores chequeados.

Acciones efectuadas para obtener la tabla de cantidades e importes de las ventas de tomates efectuadas durante el mes de julio de 1998, mes en que estuvo de oferta la lechuga:

Se realizarán acciones sobre los datos a los efectos de comparar con el mes de junio de 1998, mes en el cual estuvo en oferta la lechuga y se presume su influencia en la venta de tomates.

En primer término se hará un filter sobre la tabla de las ventas. Se filtrarán los registros cuya fecha sea <> junio de 1998 y cuyos nombres de productos vendidos no incluyan a la lechuga y a los tomates. Todo se guardará en la tabla de salida LeTojun98Tbl.

En segundo lugar se chequeará la correctitud de los cálculos, tal cual fue realizado en el análisis anterior, teniendo en cuenta en este caso que el precio de venta del tomate será obtenido de la tabla de productos, mientras que el de la lechuga de la tabla de ofertas.

En tercer lugar, una vez efectuadas las correcciones y la limpieza pertinente, se procederá a quedarse sólo con los tomates, por lo que se filtra sobre LeToJun98Tbl, los nombre <> lechuga, generándose la tabla Tojun98Tbl.

Por último se procede a seleccionar los atributos que interesan, que en este caso son la cantidad y el importe.

select ($ToJun98Tbl$, **[** $cantidad$, $importe$ **]**, $ToJun98Tbl$ **).**

Llegando así al final del análisis correspondiente a la influencia de la oferta de la lechuga, habiendo generado la tabla Tojun98Tbl cuyo contenido es la cantidad y el importe de cada venta de tomate efectuada durante el mes de junio de 1998.

Acciones efectuadas para obtener la tabla de cantidades e importes de las ventas de tomates efectuadas durante el mes de agosto de 1998, cercano a julio y sin ofertas en las verduras.

Se realizarán acciones sobre los datos a los efectos de comparar con el mes de agosto de 1998, mes cercano en el calendario al de julio y en el cual no hubo verduras en oferta.

Se efectuará un procedimiento similar a los anteriores en cuanto a los filtrados y selecciones de registros y atributos, la diferencia estará en que no se utilizará la tabla de las ofertas.

En primer lugar se filtrará de la tabla de Ventas, aquellas cuya fecha sea <> agosto de 1998 y cuyo nombre de producto sea <> tomate, generándose como salida la tabla ToAgo98Tbl.

Se efectuarán los cálculos de correctitud tal cual se hicieron antes y finalmente se seleccionarán los atributos cantidad e importe, quedando generada y verificada la tabla ToAgo98Tbl.

Acciones efectuadas para obtener la tabla de cantidades e importes de las ventas de tomates efectuadas durante el mes de julio de 1997:

Se elige este mes a los efectos de mostrar el comportamiento en la venta de tomates durante un mes en el cual no estuvieron en oferta y que coincide con la estacionalidad del mes en que sí estuvieron en oferta (julio98).

El procedimiento a seguir es similar al descripto recientemente, para el estudio de agosto de 1998. La única diferencia radica en que se debe filtrar por la fecha <> julio de 1997 y el resto se desarrollará de la misma manera.

Al finalizar esta parte del estudio, se habrá generado la tabla ToJul97Tbl, verificada y en la cual constarán las cantidades e importes de los tomates vendidos durante julio de 1997, mes en que no hubo oferta de verduras.

Por último y sintetizando, cabe destacar la generación de cuatro tablas resultado:

- ToJul98 = venta de tomates durante julio de 1998.
- ToJun98 = venta de tomates durante junio de 1998.
- ToAgo98 = venta de tomates durante agosto de 1998.
- ToJul97 = venta de tomates durante julio de 1997.

Con la preparación de los datos resuelta, es el momento de definir la función objetivo para cada una de las tablas y poder ejecutar los programas correspondientes para analizar las influencias de cada caso.

Tal vez en este ejemplo en particular, no resulte tan ventajoso el resultado del algoritmo de data mining, quizás hasta pueda ser reemplazado por consultas SQL, pero sin embargo es importante notar la potencia que dio al estudio toda la preparación previa de los datos: los filtros, las selecciones y los chequeos permitieron simplificar el estudio de mining posterior al mismo tiempo de garantizar resultados más eficientes, ya que no es lo mismo convocar un algoritmo con los atributos de las tablas originales como parámetros, que convocarlo con los datos previamente tratados y limpios.

Si se decidiera no aplicar ningún algoritmo de data mining, la función objetivo quedaría expresada de la siguiente forma:

objetivo (“cleaning”, **[ ]**, Estructura **).**

En la ejecución del motor correspondiente, el objetivo se traducirá en ejecutar hasta la etapa de cleaning del proceso de KDD, especificada en último término con el KDD L. Al no existir condiciones impuestas desde el objetivo, se tomarán las descriptas durante la especificación de la etapa señalada. La salida será la correspondiente al clean definido, que seguramente se tratará de una tabla.

Como en este ejemplo se prepararon cuatro tablas, deberían definirse cuatro funciones objetivos a los efectos de que cada una de ellas tenga como entrada en su ejecución, cada una de las tablas preparadas.

Si las tablas generadas conservaran los nombres de todos los productos vendidos simultáneamente, podrían aplicarse algoritmos de data mining a cada una de ellas para proceder a comparar luego, los diferentes grupos de ventas, según las condiciones dadas de cada mes en estudio.

7.3. PROBLEMAS ASOCIADOS Y PLANTEO DE SOLUCIONES

Si se compara la manera de resolver la sucesión de actividades del proceso de KDD en [Adr96] con la aplicación de la herramienta definida en esta tesis en 6.1., pueden citarse numerosas conclusiones importantes.

El KDD Language guió al usuario de manera automática, y a través de una interfaz amigable, en todo el proceso de preparación de los datos.

En primer término posibilitó una descripción en lenguaje natural de todo el dominio del problema, definiendo el objetivo del negocio, los elementos primordiales, sus características salientes y las vinculaciones existentes entre ellos.

Por otra parte, fue posible la libre elección del orden en el cual ejecutar cada una de las tareas, permitiendo incluso, optar por la no ejecución de algunas o convocar en más de una oportunidad la ejecución de otras.

Todo esto permitió llegar a la etapa de Data Mining con el conocimiento necesario incorporado al proceso y dentro de un accionar indisoluble entre la preparación de los datos y la extracción de la información.

Por último es indispensable destacar que mientras en [Adr96] la sucesión de tareas es realizada "a mano", sobre la organización real de los datos y con la necesaria presencia del conocedor de sistemas, en 6.1. se concreta un enfoque absolutamente diferente, cumpliendo con todas las necesidades planteadas en el apartado 4.- de esta tesis como superación de la manera en que hoy se encaran estos procesos.

Fue realizado un único estudio abarcativo de todas las etapas del KDD, tanto la preparación de los datos como la aplicación de un método de mining elegido.

Quedó generada toda la documentación correspondiente a las acciones especificadas, metodología importante a los efectos de poder encarar estudios futuros sin duplicar tareas y sin desperdiciar lo útil de lo realizado en el presente trabajo.

Se seleccionaron los datos, limpiaron, enriquecieron y codificaron, concluyendo con la especificación del objetivo del estudio, para lo cual se decidió mostrar una posible aplicación del método de sequential patterns con las condiciones deseadas para su ejecución.

Nótese que podría haberse aplicado cualquier otro método de data mining, así como también podrá efectuarse en el futuro, ya que los datos quedan preparados y listos para que en cualquier momento se especifique sólo la función objetivo con el algoritmo que se desee aplicar.

Cabe realizar a continuación un análisis acerca de la manera de encarar la aplicación del algoritmo de Mining Generalized Association Rules presentado en [Sri95], con el uso de la herramienta presentada en esta tesis y cuyo ejemplo se mostró en el apartado 6.2..

En el paper mencionado, como en la mayoría de los estudios realizados sobre métodos de data mining, se observa que son abordados sin preparación previa de los datos, se trabaja directamente sobre una tabla de la organización real de los mismos, en este caso sobre la de Transacciones.

Esto conlleva a permitir un resultado poco útil si se tiene en cuenta la posible suciedad de los datos o su inadecuada representación y por lo tanto es absolutamente necesario trabajar previamente sobre ellos en todo el proceso de preparación, codificación y limpieza, a los efectos de garantizar que el resultado de la ejecución del método elegido, en este caso las reglas de asociación, sea realmente útil y confiable.

Por otra parte, podría quererse trabajar sólo sobre una porción de la tabla, por ejemplo en este caso, solo sobre las ventas realizadas durante el mes de octubre, o también podría ser sobre los comestibles vendidos y no sobre todos los productos. Esto seria imposible de especificar si no se contara con un lenguaje con las características del definido en esta tesis, o de lo contrario habría que convocar a un especialista en informática para la reorganización de los datos, lo cual resultaría muy poco práctico para encarar estudios futuros en los cuales las porciones de datos seleccionadas para el análisis fueran variables.

En el ejemplo descripto, a partir de la acción de filtrado, se obtuvieron sólo las ventas de octubre, luego se procedió a la limpieza de los datos y entonces sí quedaron los datos descriptos en un lenguaje natural, ordenados, seleccionados y limpios de acuerdo al interés momentáneo del usuario.

En la especificación de la función objetivo ( ). Queda expresado el nexo indispensable entre toda la preparación de los datos y los algoritmos de data mining.

En este ejemplo particular, se mostró cómo el KDD L efectúa ese nexo entre toda la preparación de y la ejecución del algoritmo de data mining elegido. Los parámetros de entrada para dicho algoritmo fueron en este caso, la tabla ListaTbl, que contiene el número de venta y el nombre de los productos vendidos durante el mes de octubre, en lugar de la de Transacciones del paper.

Por otra parte, es importante destacar que con el KDD L, pueden especificarse problemas para la ejecución de etapas del KDD que no culminen necesariamente con la aplicación del mining.

Esto último quedó expresado en el ejemplo que estudió la influencia de las ofertas en las ventas globales de un producto, donde fue utilizado el KDD L para especificar un dominio en el cual se proyectaron efectuar acciones como el filtrado, la selección y la limpieza, resultando tan potente estas aplicaciones que no resultó necesario aplicar el mining. La generación de las cuatro tablas resultó suficiente a los efectos de investigar el propósito planteado, bastaría con aplicar sentencias SQL para completar el análisis.

8.- CONCLUSIONES

Quedó plasmado un lenguaje de especificación de dominios capaz de superar las deficiencias enumeradas con relación a los procesos de KDD y data mining estudiados.

El KDD Language permite al usuario realizar una descripción de su problema en un ambiente natural de trabajo, parametrizando los algoritmos de data mining sobre diferentes dominios, de manera de poder aprovechar al máximo el conocimiento incorporado en cada caso y complementar de esta forma las herramientas existentes en el mercado, para tornarlas inteligentes y lograr en consecuencia mejores resultados.

Fue diseñado para abordar conjuntamente la totalidad del proceso de KDD, de manera indisoluble la preparación de los datos, la extracción de la información y el reporte de los resultados obtenidos.

Por último es importante destacar que con su ejecución se sugiere con una metodología de trabajo tal que reduce al mínimo posible las tareas manuales, además de posibilitar la conservación de toda la documentación pertinente a cada etapa del proceso.

9.- FUTUROS TRABAJOS Y POSIBLES EXTENSIONES DEL LENGUAJE

9.1. FUTUROS TRABAJOS

A los efectos de generar una herramienta completa y para poder lograr su aplicación, queda pendiente la realización de tareas, que si bien algunas fueron estudiadas para elaborar este trabajo, merecen una investigación profunda que supera los alcances de la presente tesis.

En primer lugar resulta necesaria la programación de una interfaz con el usuario, para lo cual se estuvo estudiando Visual Basic y analizando diversos diseños posibles. Dicha interfaz debería responder a un ambiente de trabajo que resulte sencillo de comprender para el usuario y natural a los efectos de permitirle expresar el dominio de su investigación con la metodología definida en este documento.

En segundo término habrá que programar el nexo entre la interfaz y el KDD Language, motor que deberá traducir el conocimiento expresado por el usuario en sucesivas pantallas de trabajo, al formato de especificación definido como primitivas del lenguaje propuesto.

Luego restará programar la ejecución de las primitivas de especificación, para lo cual será necesario un motor capaz de tomar toda la especificación descripta y transformarla en acciones sobre las bases de datos poseedoras de los datos sobre los cuales desee obtenerse la información oculta.

Por último queda pendiente la profundización de los reportes posibles que deberán ser entregados al usuario de manera clara y comprensible para ayudarlo en el objetivo final de todo su estudio, ya sea la predicción de comportamientos de entidades de su dominio o la fijación de políticas de acción futura.

9.2. POSIBLES EXTENSIONES DEL LENGUAJE

A lo largo de toda la investigación realizada, surgieron diferentes inquietudes que recibieron un tratamiento particular, pero que en algunos casos no fue posible generar propuestas de cómo encarar sus soluciones.

- En cada etapa del proceso de KDD, debería estar contemplada la posibilidad de una ejecución automática, que la herramienta, a partir de la aplicación de heurísticas, pueda ayudar al usuario en las diferentes acciones que éste deba concretar.
  Por ejemplo, cuando se trate de la selección de atributos, debería poder indicarle cuáles parecen más importantes y para ello podría analizar automáticamente la entropía de los datos: (1 - P (exista un determinado valor)).
  En el caso del coding, habría que estudiar la posible generación de un coding automático, por ejemplo con la aplicación de un algoritmo de renumeración que agrupe los valores iguales y avise a manera de "warning" cuando halla encontrado n valores diferentes.

- Al finalizar cada etapa, el usuario debería poder convocar la ejecución de un algoritmo que le informara acerca de la diversidad de datos de la tabla que envíe como parámetro. Es decir, si por ejemplo quiere saberlo con respecto a una tabla "Nueva", después de la selección de los datos, el algoritmo le devolverá para cada propiedad, cuántos valores diferentes tiene y también podrá querer ver cuáles son esos valores. La idea de esto es brindarle al usuario un panorama de sus datos de tal manera que pueda ver si le sirve o no lo que hizo o si desea seleccionar nuevamente datos, o filtrarlos con algún criterio.

- Durante la interfaz, el usuario debería tener la posibilidad de crear una etapa propia, que sería ejecutada tomando como base las acciones definidas en el KDD-L. Debería darle un nombre y definir las tareas que desea desarrollar con ella.

- Podría contemplarse la posibilidad de ejecutar el filtrado sobre condiciones que abarquen más de un registro. Por ejemplo sobre las ventas globales.

- Habría que estudiar el posible agregado de una etapa inicial, de categorización de los datos, la cual podría ejecutarse previamente al filter y debería estar basada en un pseudo clustering.

- La etapa de enrichment podría contemplar la posibilidad de que la tabla enriquecedora no este organizada bajo la relación 1 a 1 y por lo tanto en su ejecución, genere, por ejemplo, 10 registros en la tabla enriquecida (por tener 10 registros con la misma clave). Esto no sirve para data mining. Habría que pensar en transformar primero la tabla enriquecedora.

- La etapa cleaning debería ser encarada desde una visión más global, permitiendo recuperar datos, reemplazando los incorrectos por valores correctos.
  En este punto se describirá un estudio realizado sobre la etapa de cleaning que supera la concepción de limpieza enfocada en el KDDL. En el lenguaje creado, la limpieza es encarada a los efectos de erradicar los valores incorrectos, pudiéndoselos transformar en “null” para luego eliminarlos.
  En este apunte se aborda el estudio desde la perspectiva de la verificación y posible corrección de los valores incorrectos.

  Se analizaron errores de tipeo que aparecen frecuentemente, como por ejemplo:
  - las fechas, que suelen transformarse en inválidas o encontrarse fuera de las cotas definidas.
  - variables numéricas reales como en el siguiente caso:

  se sabe que 20.00 <= X <= 30.00 y en un registro se encuentra X = 2.803, se lo podría corregir porque seguro que el dato correcto es 28.03.
  - valores muy fuera de rango, como por ejemplo si existe la variable X definida de la siguiente manera: 10 <= X <= 15, y de golpe aparece un X = 80, muy probablemente se hayan equivocado de tabla o de atributo y tal vez se lo pueda corregir.

  En [Adr96] se plantea el uso de algoritmos de reconocimiento de patrones.

  Producto de todo el estudio realizado se propone en primer término el uso de una primitiva “verify”, mediante la cual se obtendrá el porcentaje de valores correctos

según la cota descripta en dicha primitiva. La idea es poder discernir si se trata de un porcentaje importante o no, como para encarar la corrección de los valores fuera de cota o simplemente eliminarlos.

verify (BD.Tbl, **[** cond1, ..., condn**]**, % **).**

BD.Tbl = Tabla de la Base de Datos de la cual se verificarán los valores.
cond1, ..., condn = condiciones correctas que se desean verificar sobre los valores.
% = parámetro de salida que devolverá el porcentaje de valores correctos.

A partir del porcentaje obtenido se decidirá si se desea corregir los datos incorrectos o eliminarlos directamente.
En el caso de corregirlos, habrá que individualizarlos para poder reconocer su error. Será necesario trabajar con porciones chicas de las tablas para no recorrer millones de datos a cada rato.
En función de ello se propone filtrar los valores correctos y agrupar los equivocados en una tabla que podría ser llamada MalaTbl.
Luego habría que seleccionar sólo el atributo en cuestión, de esa tabla MalaTbl y finalmente proceder a corregirlos.

filter (BD.Tbl, **[** cond1, ..., condn **]**, MalaTbl **).**

select (MalaTbl, **[** X **]**, XmalaTbl **).**

En la tabla XmalaTbl se guardaron los valores incorrectos del atributo X, por lo cual ya se puede proceder a corregirlos.

replace (XmalaTbl, ValorIncorrecto, ValorCorrecto **).**

Cómo funcionaría esto para un ejemplo concreto:
Se trata de facturas pagas al contado que figuraban con " " en los registros correspondientes al pago de "contado".

filter ($BD.Facturas$, **[** $pago$, "<>", $ $ **]**, $PagoTbl$ **).**

select ($PagoTbl$, **[** $pago$ **],** $Pago1Tbl$ **).**

replace ($Pago1Tbl$, $ $, contado **).**

A continuación se sugerirá cómo proceder si se decidiera eliminar los registros incorrectos en lugar de corregirlos:

verify ($BD.Tbl$, **[** 100, "<=", X **]**, % **).**

Como en este caso devuelve un % que se acepta, se decide borrar de la base los valores erróneos sin efectuar un análisis particular de cada caso.

replace ($BD.Tbl$, **[** X, ">", 100 **]**, "null" **).**

filter (BD.Tbl, **[** $ X $, "=", "null"**]**, BuenaTbl **).**

Borra los que tienen valor "null" y devuelve la tabla de salida sin los null, que podría ser la misma o cambiarle el nombre.

- Sería importante considerar la posibilidad de agregar nuevos conceptos, contemplando el caso de pretender crear conceptos ya existentes, con propiedades similares o diferentes.

  Si por ejemplo se tiene:

  concepto (cliente).
  propiedades (cliente, [numero, nombre, direccion, telefono]).

  Y se desea agregar el e-mail:

  A través de la interfaz, habría que volver a la pantalla de definición de propiedades y definir:

  propiedades (cliente, [e-mail]).

  Mediante el testeo en la pantalla de definición de propiedades habría que poder acceder a las propiedades ya definidas de clientes y allí observar que no estaba incluido el e-mail, entonces habría que proceder como antes. Será traducido al lenguaje de especificación y quedará agregado el e-mail a las propiedades de los clientes.

  Al testear nuevamente, debería observarse que cliente posee todas las propiedades mencionadas. En la traducción al lenguaje será:

  propiedades (cliente, [numero, nombre, direccion, telefono, e-mail]).

  Podría existir el siguiente caso:

  Con la creación de un concepto nuevo, a partir del uso de reglas, podría definirse un concepto ya existente pero con diferentes propiedades.

  En algún momento fue definido:

  regla (buendocente, [docente, [concurso, "=", $ regular $], [encuesta, "=", 10]]).

  Acaban de crearse los buendocente con las propiedades de los docentes pero con concurso = regular y encuesta =10 .

  Más tarde se desea crear a los buendocente pero con otras características:

  regla (buendocente, [docente, [asistencia, "=", $ perfecta $]]).

  En este caso la herramienta le avisará al usuario que el concepto buendocente ya fue definido y le enseñará sus propiedades, así como le consultará si desea agregarle la asistencia como nueva propiedad o si desea crear un concepto nuevo.

Si el usuario elige agregar la propiedad asistencia entonces deberá definir:

propiedades (docente, [asistencia]).

Y por el mecanismo anterior se incorporará la asistencia a los docentes y luego por herencia y relación esun, se le agregará la asistencia a los buendocente.
Pero si el usuario decidiera crear un concepto diferente para los perfectos asistentes, entonces deberá cambiar la regla, o sea cambiar el nombre del nuevo concepto y definirlo con las propiedades que para este caso desee.

- Al aplicarse una regla, la herramienta debería avisar cuál fue el resultado, por ejemplo decir que no se aplicó en el x % de los casos, para permitirle al usuario definir si esa aplicación le sirve o si prefiere replantear la regla enunciada. En la aplicación de reglas se puede tener en cuenta, en algún punto, la confianza, soporte u otras medidas que se quieran usar a los efectos de determinar y sugerirle al usuario si ese nuevo concepto tiene el suficiente sustento estadístico como para ser usado y/o priorizado.
  En este punto fue estudiada la existencia de diferentes medidas estadísticas que ayudan a tomar decisiones en función de las necesidades de cada caso. [Car]

- Podría encararse el estudio de jerarquías dentro de los conceptos: dependiendo del sustento estadístico y si existiera la posibilidad de diferenciar niveles de importancia, los nuevos conceptos podrían tener mayor jerarquía que los iniciales (salvo que el usuario indicara lo contrario), pues al constituirse a partir de condiciones establecidas sobre los anteriores, resultaría lógico suponer que su importancia a la hora de explotar los datos deba ser mayor. Estos conceptos jerárquicos serían tenidos en cuenta por las heurísticas en la búsqueda de propuestas automáticas para las diferentes acciones.

- Sería importante encarar un estudio sobre diferentes posibilidades de sampling para el caso en que la herramienta no pueda correr sobre bases de datos muy grandes. Existen muchas investigaciones realizadas en este tema y diferentes propuestas de algoritmos de data mining que lo tienen en cuenta. Constituye un punto realmente importante y que merece un estudio profundo, por lo cual podría encarárselo como temas propios de futuras tesis.

10.- APENDICE

Si bien los algoritmos de data mining tienen comportamientos individualizados para cada problema, resulta importante estudiarlos a los efectos de poder conformar grupos de acuerdo a algunas de sus características y permitir ayudar al usuario en su elección del método a aplicar, según el tipo de estudio que deba realizar.

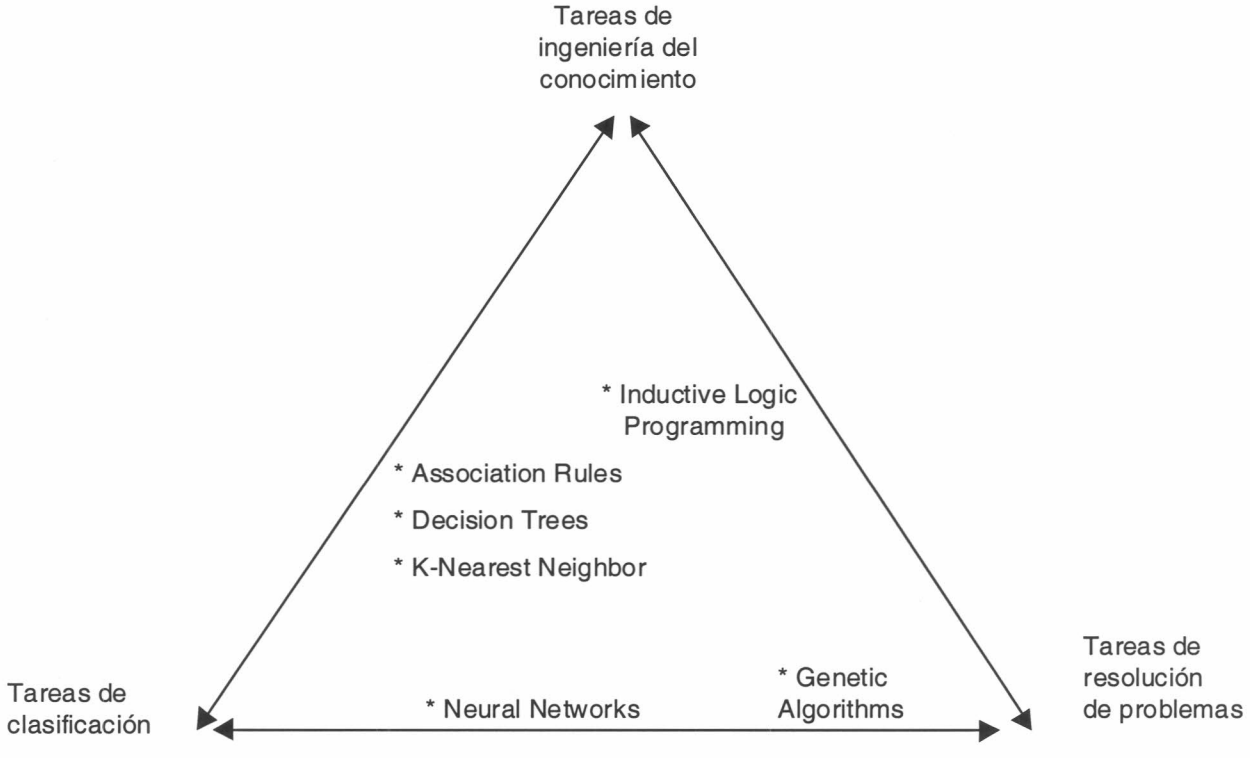
Existen investigaciones realizadas acerca de las vinculaciones existentes entre los diferentes tipos de problemas a resolver con las diferentes técnicas de búsqueda de la información.

Se expondrá aquí uno de los estudios existentes, obtenido de [Adr96], basado en encuestas a los usuarios de métodos de data mining.

En primer término, se analizan las tareas posibles de realizar, dividiéndolas en tres grupos fundamentales:

- ingeniería del conocimiento
- clasificación
- resolución de problemas

Para cada una de estas tareas se ha estudiado el comportamiento de algunos de los métodos de data mining y se representa el resultado obtenido en el siguiente gráfico:



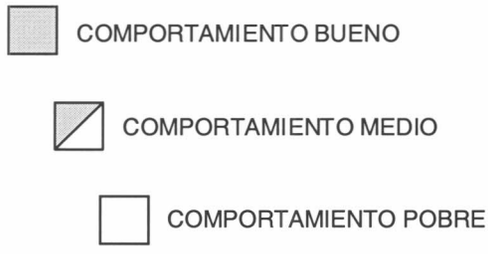
Las redes neuronales se comportan mejor para realizar tareas de clasificación que los algoritmos genéticos, los cuales resultan más apropiados para la resolución de problemas, en tanto que la programación lógica inductiva, es más apropiada para la resolución de tareas sobre ingeniería del conocimiento.

Los métodos de Association Rules, K-Nearest Neighbor y Decision Tree, son los apropiados para encarar clasificaciones.

Pero este gráfico muestra sólo la superficialidad de un estudio más profundo, que consistió en tomar cinco de los métodos de data mining existentes y analizarlos desde su comportamiento en tres aspectos que resultan importantes para la elección del algoritmo a aplicar:

- calidad del input
- calidad del output
- performance

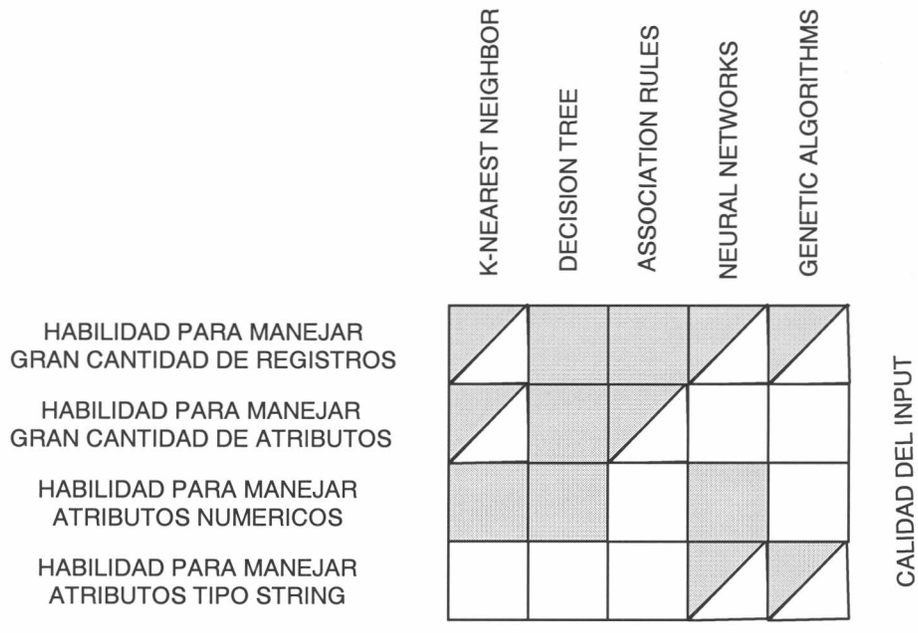
El análisis de cada uno de ellos se visualizará en diferentes gráficos que muestran el resultado de las comparaciones realizadas, correspondiendo la siguiente lectura:



Se plantea como aspecto importante dentro de la calidad del input, la cantidad de registros, ya que implica variaciones en el comportamiento de los algoritmos. No todos responden con la misma eficiencia frente a grandes cantidades.

Por otro lado, también merece atención la cantidad de atributos. Por ejemplo, la performance de las redes neuronales como la de los algoritmos genéticos se ve deteriorada con el aumento de atributos.

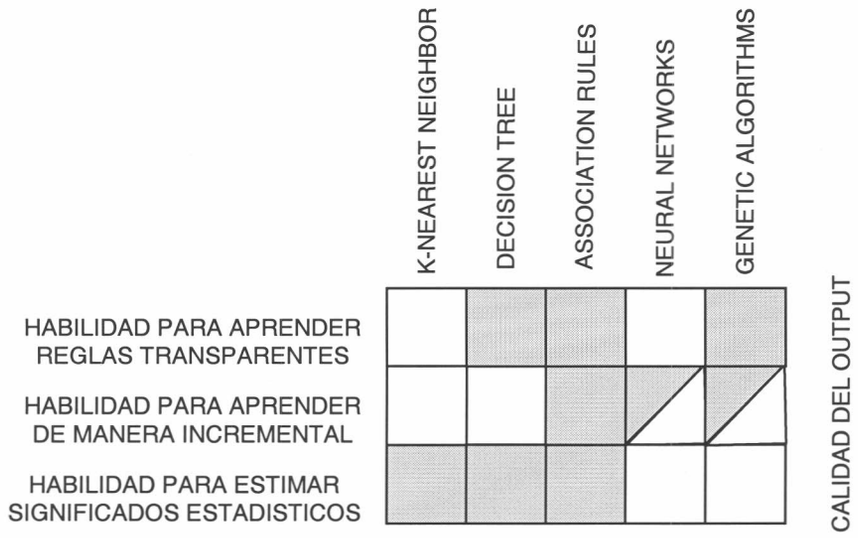
Y por último, también influye el tipo de atributos ya que no todos los algoritmos son buenos en el manejo de variables tipo string o numéricas y esto puede jugar un rol decisivo en la elección del método de mining a aplicar.



En cuanto al output, es necesario tener en cuenta si el algoritmo es capaz de aprender reglas. Algunos, como el k-nearest neighbor o las redes neuronales, dan un sí o no como respuesta sin explicitar el por qué.

En otros casos podría ser importante seleccionar algoritmos capaces de aprender incrementalmente cuando la nueva información se torna disponible, éstos son aptos para revisar sus teorías, lo que implica que no necesitan recomenzar el proceso de aprendizaje cada vez que surge nueva información y esto resulta de vital importancia si se trata de grandes volúmenes de datos.

Otro punto interesante de analizar, es la capacidad de estimar el significado estadístico de los resultados encontrados. En el caso de las redes neuronales o los algoritmos genéticos es a menudo muy difícil de evaluar el resultado desde un punto de vista estadístico. Y para algunas organizaciones, éste podría ser un factor decisivo para seleccionar el algoritmo de data mining a aplicar.

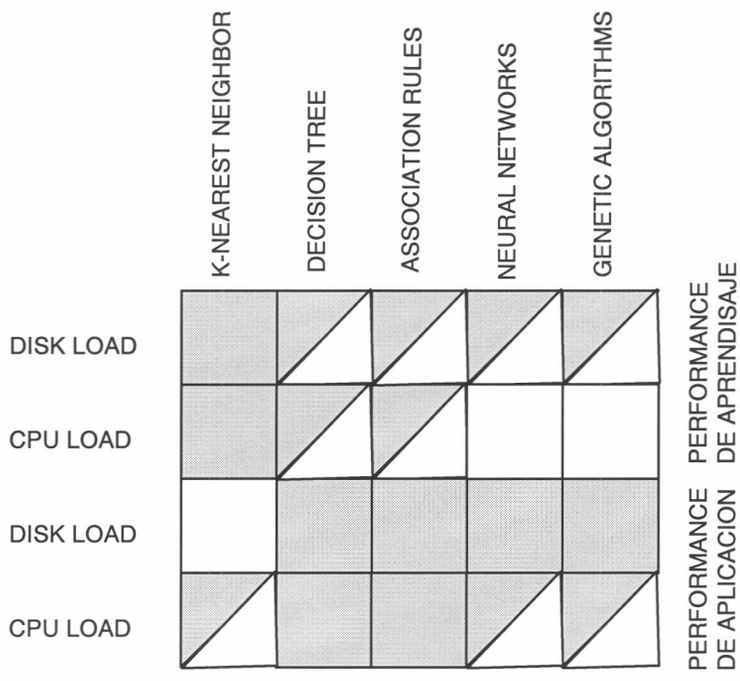


El otro aspecto evaluado es la performance. Analizada en dos situaciones diferentes:

- la etapa de aprendizaje
- la etapa de aplicación real del algoritmo

En promedio, los algoritmos que aprenden rápidamente son algo lentos en la etapa de aplicación y viceversa.

En el siguiente gráfico se dan estimaciones para la carga en disco y en CPU para ambas etapas. Notar que un algoritmo se considera bueno, si ambas etapas son bajas.



Con una mirada rápida podría observarse el porqué de la popularidad de los métodos de Association Rules y Decision Trees: el análisis arroja resultados buenos tanto en el manejo del input como del output como en su performance.

Toda esta investigación podría profundizarse, tanto en la cantidad de algoritmos analizados como en los puntos de estudio y comparación efectuada. En esta presentación se intentó dar un pantallazo general, teniendo en cuenta los aspectos más salientes de los métodos más utilizados.

ESTUDIOS COMPARATIVOS

Continuando con el estudio de las características de los métodos de data mining y agregando a dicho análisis algunas de las herramientas comerciales existentes hoy en el mercado, resultan de suma importancia las comparaciones que se pueden efectuar entre ellos y contraponiendo el comportamiento del KDD Language creado en la presente tesis.

A continuación se presentarán las comparaciones efectuadas sobre las posibles acciones a llevar a cabo con algunos de los algoritmos de data mining: Association Rules, K-Nearest Neighbor, Decision Tree y Sequential Patterns.

Association Rules:

Los algoritmos de Association Rules procuran encontrar implicancias entre atributos de las bases de datos. Por ejemplo, pueden determinar que el 80% de los clientes usuarios de tarjetas de crédito, son propietarios de por lo menos un inmueble y un automóvil.

Las reglas de asociación son útiles en la medida que se tenga una idea sobre la búsqueda que se desea realizar, ya que no existe ningún algoritmo que devuelva automáticamente todas las reglas interesantes presentes en las bases de datos. Si la búsqueda de reglas no se realiza con algún tipo de guía, seguramente sucederá que junto con reglas útiles, se hallen muchas asociaciones inútiles y por el contrario si se generan búsquedas muy limitadas, seguramente el resultado obviará reglas interesantes.

Como en el caso de los árboles de decisión, resulta conveniente encarar la búsqueda de asociaciones bajo un entorno de atributos interesantes que actúen a manera guía en el estudio que se desee realizar.

Los algoritmos de AR, tienen en general la parametrización correspondiente a las medidas de probabilidad que indican el umbral aceptable para la generación de las reglas: soporte, confianza o alguna otra definida con el mismo propósito.

Con la modificación de alguna de estas medidas se puede aumentar o disminuir la cantidad de reglas resultado generadas, pero esto no significa que se brinde conocimiento al algoritmo, o que se lo torna inteligente. Solo se trata de un parámetro que mide o indica la representatividad del resultado.

Los diferentes algoritmos de Association Rules revisados [Sri95], utilizan el soporte para delimitar el conjunto de ítems interesantes, candidatos a integrar las reglas resultados, pero ésto siempre se desarrolla en el campo de las estadísticas y en el trabajo concreto sobre las tablas y bases de datos. Si en cambio se pretende trabajar sobre el nivel del conocimiento, será necesario recurrir al uso de un lenguaje de representación, capaz de brindar conocimiento a los algoritmos mencionados.

Decision Trees:

Los algoritmos que generan árboles de decisión parten de tablas con datos sobre el comportamiento de determinadas variables.

El proceso consiste en determinar cuál es el atributo más influyente en la determinación del comportamiento que se quiere investigar y encontrar para él un valor umbral adecuado que permita separar en dos conjuntos a todos sus valores, seguir con este mismo procedimiento para cada uno de los atributos restantes y armar de esta forma un árbol de decisión para toda la base de datos. Existen muchos algoritmos capaces de armar estos árboles de manera automática y son en general muy efectivos, con grado de complejidad n(logn).

Estos algoritmos funcionan muy bien sobre grandes bases de datos y otra de sus ventajas es que son capaces de brindar una clara descripción de todo el proceso de decisión.

A pesar de poseer muchas ventajas, no siempre constituyen el método más apropiado.

Los algoritmos que generan árboles de decisión tienen como parámetros la profundidad deseada, la definición del nodo raíz, que se corresponde con el atributo que se considera más influyente en el camino del análisis a efectuar y el atributo objetivo o columna target, que describe el objetivo de la clasificación. Pero si en cambio se quisiera agregar conocimiento a estos algoritmos para hacerlos inteligentes, se necesitaría desarrollar la preparación previa de los datos de manera de guiar la clasificación posterior. Y con el uso de un lenguaje de especificación, el usuario, totalmente ajeno al mundo de los sistemas de computación, podría describir todo el conocimiento que, producto de su experiencia cotidiana, posee sobre el tema de investigación.

K-Nearest Neighbor:

El método K-Nearest Neighbor [Adr96] interpreta a los registros como puntos en el espacio de datos y trabaja con el concepto de vecindad: los registros que están vinculados entre sí, viven en el mismo vecindario.

La filosofía básica que encierra este método es "te comportarás como tus vecinos lo hacen". Si se busca predecir el comportamiento de determinado individuo, debe comenzarse por identificar el comportamiento de por ejemplo 10 individuos que vivan cerca suyo en el espacio de datos. Habrá que calcular la media de sus valores y esa será la predicción para el individuo investigado.

En estos algoritmos se parametriza el k = cantidad de vecinos que se desea investigar para establecer el comportamiento de la entidad deseada.

No es una técnica de aprendizaje, sino un método de búsqueda.

Tiene complejidad cuadrática, o sea que sólo se lo utiliza con muestras pequeñas o conjuntos de datos de tamaño limitado.

Tiene problemas para manejar gran cantidad de atributos, por lo que se han estudiado estrategias para mejorarlo, como por ejemplo calcular la importancia relativa de cada atributo para priorizarlos en la investigación.

Pero estos estudios se desarrollan analizando estadísticamente los valores almacenados en las tablas o bases de datos. Con el uso de un lenguaje de especificación, trabajando sobre el nivel del conocimiento, podría mudarse el escenario de acción en la faz previa a la ejecución del algoritmo y permitir al usuario la posibilidad de expresar todo su saber sobre el estudio particular, con el objetivo de priorizar atributos según su influencia conocida, producto de la experiencia, y no debido a su representatividad estadística.

Sequential Patterns:

Los algoritmos de Sequential Patterns descubren secuencias en el comportamiento de los datos.

En general son usados para el reconocimiento de secuencias temporales, es decir, para determinar patrones que cumplen los datos de manera sucesiva a lo largo del tiempo.

Son parametrizados con el soporte, medida que determina la proporción de representatividad de las secuencias generadas como resultado.

El algoritmo básico [Agr95], [Sri96] se encarga de transformar la tabla de transacciones en una de secuencia de clientes, con la identificación del cliente como clave principal y el tiempo transcurrido como clave secundaria. Luego se encuentran los litemsets = itemsets con mínimo soporte. A continuación se transforma cada secuencia de cliente en una representación tal que cada transacción es reemplazada por el conjunto de todos los litemsets contenidos en esa transacción. Si una transacción no contiene litemsets, entonces no es conservada en la nueva representación. Luego se generan las secuencias, para lo que se pueden utilizar diferentes métodos y finalmente se determinan las secuencias maximales, las que constituyen el resultado buscado.

Nuevamente se trata de métodos que se aplican sobre las tablas o bases de datos, sin poder usufructuar el conocimiento que el usuario puede tener sobre el estudio a realizar y definiendo elementos interesantes solo en función de cálculos estadísticos.

Con la propuesta de uso de un lenguaje de especificación, no se plantea el reemplazo de los cálculos estadísticos, sino el agregado de conocimiento externo que permitirá ejecutar de manera inteligente los algoritmos y por lo tanto la obtención de mejores resultados.

Ejemplos:

Se trabajará sobre dos ejemplos que mostrarán cómo influye el conocimiento que pueda brindar el experto en el tema a investigar y su posibilidad de incorporarlo al estudio con el uso del KDDL.

1.- Se propone trabajar en primer término sobre la definición de un dominio que permitirá establecer vinculaciones existentes entre la caracterización de la superficie y los riesgos de contaminación del agua subterránea.

El objetivo de este análisis es el de poder priorizar en la práctica, los análisis de riesgo de contaminación del agua, asociando los pozos correspondientes a cada área geográfica con sus características de superficie y previendo, a partir de las reglas resultado, el tipo de contaminación del agua subterránea más posible de hallar en cada zona.

Para ello, el experto en calidad del agua define:

concepto (pozo **).**
concepto (superficie **).**

propiedades (pozo**, [** nombre, area_geografica, profundidad, caudal, residuos_salinos, nitratos, cromo, hidrocarburos **]).**

propiedades (superficie, **[** area_geografica, topografia, morfologia, ocupacion_industrial, ocupacion_residencial, ocupacion_agricola **]).**

Junto con el informático se definirá el diccionario de datos, vinculando la descripción dada con los datos almacenados en las bases de datos.

En cuanto a las acciones, el experto en el tema de aguas define las siguientes:

select ($Calidad.PozosTbl$, **[** $id-pozo$, $area_geo$, $profundidad$, $caudal$, res_sal, $nitratos$, $cromo$, $hidrocarburos$ **]**, $RiesgosTbl$ **).**

Y como los datos correspondientes a las propiedades de la superficie deben ser extraídos de bases de datos externas, se procede a enriquecer la tabla Riesgos creada.

enrich ($RiesgosTbl.area_geografica$, $IGM.Topografia$, **[** $topografia$, $morfologia$ **]**, $RiesgosTbl$ **).**

enrich ($RiesgosTbl.area_geografica$, $Saneamiento.Industrias$, **[** $ocupacion_industrial$ **]**, $RiesgosTbl$ **).**

enrich ($RiesgosTbl.area_geografica$, $Censos.Habitantes$, **[** $ocupacion_residencial$ **]**, $RiesgosTbl$ **).**

enrich ($RiesgosTbl.area_geografica$, $INTA.Agricultura$, **[** $ocupacion_agricola$ **]**, $RiesgosTbl$ **).**

Como los datos enriquecidos tienen gran diversidad de valores, son codificados a los efectos de acotar los rangos posibles y obtener parámetros más manejables y claros para obtener las reglas de asociación.

A manera de ejemplo se mostrará la codificación de dos de los atributos: nitratos y ocupacion_residencial.

code ($RiesgosTbl$, $nitratos$, **[** "<=", 25, "=", $bajo$, ">", 25, "^", "<=", 50, "=", $medio$, ">", 50, "^", "<=", 100, "=", $alto$, ">", 100, "=", muy_alto **]**, $RiesgosTbl$ **).**

code ($RiesgosTbl$, $ocupacion_residencial$, **[** "<=", 100, "=", muy_baja, ">", 100, "^", "<=", 500, "=", $baja$, ">", 500, "^", "<=", 800, "=", $media$, ">", 800, "<=", 1500, "=", $alta$, ">", 1500, "=", muy_alta **]**, $RiesgosTbl$ **).**

Con todo el conocimiento agregado, quedó generada la tabla **Riesgos** que posee los siguientes atributos: **Id-pozo - area_geografica – profundidad – caudal - res_sal – nitratos – cromo – hidrocarburos – topografia – morfologia – ocup_indus – ocup_resid – ocup_agric.**

Esta tabla será enviada como parámetro al algoritmo de Association Rules definido en el motor de la herramienta y con todo el conocimiento incorporado en ella, se generarán reglas más útiles e inteligentes que si hubiesen sido encaradas solamente desde el área de sistemas.

Y quedará definida la función objetivo de la siguiente manera:

objetivo ("AssociationRules", **[** conclusion, **[** RiesgosTBL.res_sal, RiesgosTBL.nitratos, RiesgosTBL.cromo, RiesgosTBL.hidrocarburos **]**, **[** "sp", "=", 0.6 **]**, **[** "cf", "=", 0.4 **]]**, Estructura **).**

2.- En segundo término, se procede a definir parte de un dominio bancario:

concepto (caja_ahorro **).**
concepto (cuenta_corriente **).**
concepto (tarjeta_credito **).**
concepto (prestamo **).**
concepto (cliente **).**
concepto (transaccion **).**

propiedades (cliente, **[** numero, nombre, servicio, fecha_alta, fecha_baja **]).**
propiedades (transaccion, **[** tipo, servicio, cliente **]).**

relacion (efectua, cliente, transaccion **).**

Si el banco quisiera hacer data mining con el propósito de prever futuros ex clientes para intentar contenerlos, podría encarar una investigación acerca del comportamiento que hayan tenido los ex clientes durante sus últimos seis meses de vinculación con el banco.

Con alguno de los algoritmos de Sequential Patterns, podrían obtenerse secuencias de actitudes que los identifiquen.

Pero si este estudio se realizara sólo desde el área de sistemas, se perdería información útil como por ejemplo el hecho de que la tarjeta de crédito Xcard, haya cerrado durante ese período. Este es un hecho ajeno al banco y por lo tanto los clientes que se han ido por este motivo deben ser excluidos del análisis, para evitar desvirtuarlo.

Entonces, habría que plantear, entre otras acciones:

filter ($Northbank.Excli$, **[** $tarjeta$, "=", $Xcard$ **]**, Excli2 **).**

Y con esta tabla resultado, Excli2, que contendrá los datos preparados y enriquecidos con el conocimiento del experto en el tema, se convocará a la ejecución del algoritmo de Sequential Patterns definido.

objetivo ("SequentialPatterns", **[** "sp", ">", 0.3 **]**, Estructura **).**

Si en cambio se hubiese utilizado cualquiera de las herramientas comerciales existentes hoy, no se hubiese podido representar el dominio de acción, perdiéndose la incorporación del conocimiento del sector financiero del banco.

A continuación se mostrará un cuadro comparativo del comportamiento de estos cuatro algoritmos de data mining y el KDD Language en la realización de tareas que proveen conocimiento para la búsqueda de la información oculta.

Algoritmos de Data Mining vs. KDD Language:

Referencia	Descripción	Association Rules	Decision Trees	K-Nearest Neighbor	Sequential Patterns	KDD Language
Especificación del dominio	Descripción de las componentes del problema	NO	NO	NO	NO	Definición de conceptos, propiedades relaciones y herencias
Filtrado de datos	Posibilidad de filtrar registros con información inservible	NO (1)	NO	NO	NO (1)	Etapa de Filtering filter ( ).
Selección de atributos	Posibilidad de encarar la búsqueda hacia atributos que brindan información, descartando otros	NO	SÍ	NO (2)	NO	Etapa de Data Selection select ( ).
Limpieza de datos	Posibilidad de verificar la base de datos para trabajar sobre valores correctos	NO	NO	NO	NO	Etapa de Cleaning clean ( ).
Agregado de información	Posibilidad de enriquecer la información con datos ajenos a la base de datos	NO	NO	NO	NO	Etapa de Enrichment enrich ( ).

Referencia	Descripción	Association Rules	Decision Trees	K-Nearest Neighbor	Sequential Patterns	KDD Language
Codificación de valores	Posibilidad de codificar la información de manera de orientar el resultado a valores de mejor comprensión	NO	NO	NO	NO	Etapa de Coding code ( ).
Reporte de resultados	Posibilidad de elegir formato de reporte de resultados	No se Aplica	No se Aplica	No se Aplica	No se Aplica	Etapa de Reporting report ( ).

(1): La variación del soporte podría hacer modificar los resultados, descartando información (registros), pero no se debe al agregado de conocimiento que se corresponde con la etapa del filtering del KDD Language.

(2): El algoritmo de K-Nearest Neighbor tiene problemas para manejar grandes cantidades de atributos, por lo cual, entre las estrategias investigadas para mejorarlo, se encuentra la de calcular la importancia relativa de cada uno y focalizar su aplicación sólo sobre los atributos de mayor importancia.

Se estudiaron diversas herramientas comerciales y de su comparación con el KDD Language surgió el armado del siguiente cuadro. Tiene por objeto mostrar las tareas posibles de realizar con el soft existente en el mercado y el ambiente de trabajo generado con el lenguaje propuesto en esta tesis.

Herramientas comerciales vs. KDD Language:

Referencia	Descripción	Herramientas Comerciales	KDD Language
Especificación del dominio	Descripción de las componentes del problema	NINGUNA	Definición de conceptos, propiedades relaciones y herencias
Filtrado de datos	Posibilidad de filtrar registros con información inservible	POCAS	Etapa de Filtering filter ( ).
Selección de atributos	Posibilidad de encarar la búsqueda hacia atributos que brindan información, descartando otros	CASI TODAS	Etapa de Data Selection select ( ).
Limpieza de datos	Posibilidad de verificar la base de datos para trabajar sobre valores correctos	NINGUNA	Etapa de Cleaning clean ( ).
Agregado de información	Posibilidad de enriquecer la información con datos ajenos a la base de datos	POCAS	Etapa de Enrichment enrich ( ).
Codificación de valores	Posibilidad de codificar la información de manera de orientar el resultado a valores de mejor comprensión	NINGUNA	Etapa de Coding code ( ).

Referencia	Descripción	Herramientas Comerciales	KDD Language
Reporte de resultados	Posibilidad de elegir formato de reporte de resultados	TODAS	Etapa de Reporting report ( ).

Por último y a manera de ejemplo, se cita una de las herramientas de data mining más completas y comercializadas en la actualidad, Business Miner, y con el mismo enfoque comparativo anterior, se describen sus particulares características en cuanto a acciones del proceso de KDD que puede desarrollar.

Business Miner vs. KDD Language:

Referencia	Descripción	Business Miner	KDD Language
Especificación del dominio	Descripción de las componentes del problema	NO	Definición de conceptos, propiedades relaciones y herencias
Filtrado de datos	Posibilidad de filtrar registros con información inservible	SÍ	Etapa de Filtering filter ( ).
Selección de atributos	Posibilidad de encarar la búsqueda hacia atributos que brindan información, descartando otros	SÍ	Etapa de Data Selection select ( ).
Limpieza de datos	Posibilidad de verificar la base de datos para trabajar sobre valores correctos	NO	Etapa de Cleaning clean ( ).
Agregado de información	Posibilidad de enriquecer la información con datos ajenos a la base de datos	NO	Etapa de Enrichment enrich ( ).
Codificación de valores	Posibilidad de codificar la información de manera de orientar el resultado a valores de mejor comprensión	NO	Etapa de Coding code ( ).
Reporte de resultados	Posibilidad de elegir formato de reporte de resultados	SÍ	Etapa de Reporting report ( ).

Quedaron conformados diversos cuadros, los que podrían permitir al usuario tomar una decisión más fundada acerca cómo encarar el mining en su investigación.

Dando una vista rápida sobre ellos podría identificar opciones que le permiten encarar todo el proceso de KDD, de manera más o menos completa según sus características particulares.

Tendrá que conocer cuáles son las necesidades de su estudio, el grado de conocimiento de informática de quienes vayan a desarrollarlo así como también la calidad y cantidad de los datos almacenados.

Todo este relevamiento le resultará indispensable para poder determinar la utilidad de ejecutar un algoritmo de data mining sin la previa preparación de los datos, la elección de alguna herramienta comercializable pero incompleta para el desarrollo de todo el proceso de KDD y generalista en la aplicación del data mining, ó bien la determinación de encarar el estudio a través del KDD Language, bajo un ambiente de trabajo natural, sin la necesidad de ser un conocedor de informática y pudiendo enfocar la búsqueda de la información desde la especificación del dominio, pudiendo describir todas las acciones del proceso de KDD y con la posibilidad de aplicar un mining parametrizado de manera de efectuarlo en forma inteligente, habiéndole proporcionado al método general, el conocimiento particular necesario.

11.- BIBLIOGRAFIA

11.1. CITAS:

[Fra] William Frawley, Gregory Piiatetsky-Shapiro, Christopher Matheus "Artículo Introductorio sobre Data Mining"

[Adr96] Pieter Adriaans, Dolf Zantinge "Data Mining"

[Fay96] Usama Fayyad, Gregory Piiatetsky-Shapiro, Padhriac Smith "Knowledge Discovery and Data Mining, Towards a Unify Framework"

[Cas97] Paul Casarin "Using Data Mining Techniques in Auditing", Control Journal, Volume V, 1997.-

[Agr95] Rakesh Agrawal, Ramakrishnan Srikant "Mining Sequential Patterns"

[Sri95] Ramakrishnan Srikant, Rakesh Agrawal "Mining Generalized Associattion Rules"

[Hei96] Oskari Heinonen, Heikki Mannila "Attribute-Oriented Induction and Conceptual Clustering"

[Sri96] Ramakrishnan Srikant, Rakesh Agrawal, "Mining Sequential Patterns: Generalizations and Performance Improvements"

[Fay97] Usama Fayyad "Data Mining and Knowledge Discovery", An International Journal, Volume ¡, Issue 2, 1997

[Car] Colin Carter, Howard Hamilton, Nick Cercone "Share Based Measures for Itemsets"

11.2. MATERIAL DE CONSULTAS

Adriaans P. & Zantinge "Data mining" Addison-Wesley, [1996]
Agrawal R. & Srikant R. "Mining Sequential Pattern" IBM Research, [1995]
Agrawal & Imielinski & Swami "Database Mining: A Performance Perspective" IBM Research, [1993]
Ali K. & Manganaris S. "Partial Classification using Association Rules" IBM Research
Apte Ch. & June Hong S. "Predicting Equity Returns from Securities Data with Minimal Rule Generation" MIT Press [1996]
Bell David et Al "Distributed Database Systems" Addison-Wesley Publishing Company [1992]
Brachman R. "The process of Knowledge Discovery in Databases" MIT Press [1996]
Carnota Raúl & Teszkiewicz Alberto "Sistemas Expertos y Representación del Conocimiento" III E.B.A.I. [1988]
Carter & Hamilton & Cercone "Share Based Measures for Itemsets" Dep. Computer Science, University of Regina, Canadá
Ceri & Gottlob & Tanca "Logic Programming and Databases" Springer-Verlag [1990]
Date C. J. "A guide to SQL standard" Addison Wesley [1994]
De Raedt & Blockeel "Using Logical Decision Trees for Clustering" Dep. Computer Science Katholieke Universiteit Leuven, Belgium
Dehaspe & De Raedt "Mining Association Rules in Multiple Relations", Dep. Computer Science Katholieke Universiteit Leuven, Belgium
Dzeroski S. "Inductive Logic Programming and Knowledge Discovery in Databases" MIT Press [1996]
Fayyad U. & Uthurusamy R. "Data Mining and Knowledge Discovery in Databases" Comunication of ACM - November 1996/Vol 39 N 11
Fayyad U. & Shapiro G. & Smyth P. "From Data Mining to knowledge Discovery" MIT Press [1996]
Fayyad U. & Shapiro G. & Smyth P. "The KDD Process for extracting Useful Knowledge from volumes of Data" Comunication of ACM - November 1996/Vol 39 N 11
Fayyad U. & Shapiro G. & Smyth P. "Knowledge Discovery and Data Mining: Towards a Unifying Framework", Second International conference of KDD and DM, Oregón 1996.
Fayyad U. & Haussler D. & Stolorz P. "Mining Scientific Data" Comunication of ACM - November 1996/Vol 39 N 11
González O. "UBAMiner: A knowledge based Data mining" KDD & Data Mining (Kluger Publisher) (1998)
González O. "Proyecto UBAMiner" Reporte Interno Departamento de Computación F.C.E.N [1997]
Heinonen O. & Mannila H. "Attribute Oriented Induction and Conceptual Clustering" Dep. Computer Science University of Helsinki (Finlandia)
Helman P. "The Science of Database Management" IRWIN [1994]
Kumar Das S. "Deductive Databases and Logic Programming" Addison-Wesley Publishing Company [1992]
Mitra Sitansu "Principles of Relational Database Systems" Prentice-Hall International editions [1991]
Nilson N. "Problem-Solving Methods in Artificial Intelligence" McGraww-Hill [1971]
Parsaye Kamran et Al "Intelligence Databases" Willey Professional Computing [1989]
Srikant & Agrawal "Mining Sequential Patterns: Generalizations and Performance Improvements" IBM Research

Srikant & Agrawal "Mining Generalized Association Rules" IBM Research
Ullman J. "Principles of Database and Knowledge-base Systems" Vol I, II Computer Science Press [1995]

AGRADECIMIENTOS:

A mi director de tesis, Lic. Osvaldo Gonzalez, por tolerar todos los cambios sucedidos durante el trabajo, brindandome apoyo académico y motivación en todo momento.

A mi marido Elio por colaborar activamente en la presentación de la tesis y por sus valiosas sugerencias.

A toda mi familia, padres, suegros y tíos, por su permanente disposición solidaria.

A Ricardo Rodriguez y Juan Santos por sus críticas y aportes sustanciosos.