Departamento de Investigación Operativa
Instituto de Computación
Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay

Proyecto de Grado
Informe Final

Minería de Datos aplicada a problemas de Investigación Operativa

Mónica Gamarra Barrios

Julio 2006

Tutor: Ing. MSc. Omar Viera

Índice general

Índice de figuras

Resumen

La necesidad de información es inherente a todas las personas y organizaciones sin importar su actividad. La Investigación Operativa trabaja con el objetivo de proporcionar a personas y organizaciones, modelos útiles para el proceso de toma de decisiones, basándose para ello de toda la información que le es posible recabar. Las técnicas de Minería de Datos pueden ser incorporadas por la Investigación Operativa para expandir la cantidad y variedad de problemas que hoy le es posible solucionar y también para contar con otra herramienta a utilizar en los problemas que le son comunes. La Minería de Datos se puede definir rápidamente como el descubrimiento de información en grandes volumenes de datos. Es un campo en el cual se reúnen técnicas de variadas áreas de investigación, como estadística, inteligencia artificial, aprendizaje de máquina, reconocimiento de patrones y base de datos; esto le permite enfrentar enorme variedad de problemas. La Minería de Datos tienen como objetivo extraer más y mejor información de los datos disponibles. Cuando se aplica Minería de Datos a problemas de Investigación Operativa esta información proporciona más elementos con los cuales el investigador puede mejorar sus modelos y por tanto puede proporcionar a las organizaciones de mejores modelos con los cuales guiar la toma de decisiones. La mayor ventaja que presenta la Minería de Datos son las mejoras a tecnologías previamente existentes y desarrollo de nuevas tecnologías para afrontar eficientemente problemas en que se manejan grandes volumenes de datos. En la actualidad la mayoría de las organizaciones dedican importante tiempo y recursos en la acumulación de datos concernientes a sus actividades. Se espera que estos datos se transformen en información a ser usada para la toma de decisiones. Esta realidad de abundancia de datos es la que afrontan los investigadores de Investigación Operativa cuando una organización les presenta un problema. La incorporación de Minería de Datos dentro de sus técnicas de trabajo le permite avanzar rápidamente en este sentido, ya que Minería de Datos tiene años de experiencia que aportarle en el manejo de grandes volumenes de datos.

El objetivo de este proyecto de grado es realizar una primera aproximación entre Minería de Datos e Investigación Operativa, en la cual sea posible concluir si existen posibilidades de aplicar técnicas de Minería de Datos en Investigación Operativa. El proyecto contempla el estudio del Estado del Arte de Minería de Datos y de la aplicación de sus técnicas en el área de Investigación Operativa en base a un caso de estudio. El estudio del Estado del Arte de Minería de Datos comprende desde el punto de vista teórico la

investigación de procesos, métodos y algoritmos de Minería de Datos. Desde el punto de vista práctico muestra aplicaciones conocidas de técnicas y algoritmos de Minería de Datos a problemas del mundo real. Para ejemplificar el uso de la Minería de Datos se plantea un caso de estudio en el cual se aplican algunas de las técnicas y algoritmos presentados. Al mismo tiempo que se plantean problemas de Investigación Operativa para el mismo caso de estudio. Esta comparación de enfoques intenta mostrar como puede un investigador de operaciones aprovecharse de la Minería de Datos en sus proyectos. Luego se presenta el desarrollo de mejoras a un algoritmo de Minería de Datos, con el motivo de mostrar la facilidad con que se pueden realizar aportes a la base de algoritmos existentes en el área. Finalmente el algoritmo desarrollado se aplica al mismo caso de estudio con el que se probaron algoritmos conocidos de Minería de Datos y se comparan los resultados obtenidos.

Palabras clave: Data mining, Minería de Datos, Investigación Operativa.

Parte I

Aplicación de Minería de Datos a Investigación Operativa

Capítulo 1

Introducción

1.1. Motivación

El departamento de Investigación Operativa del Instituto de Computación de la Facultad de Ingeniería (Universidad de la República) está interesado en ahondar en la aplicación de técnicas de Minería de Datos en problemas de su área de trabajo. Al día de hoy se han utilizado ocasionalmente técnicas de Minería de Datos, como por ejemplo técnicas de clustering para resolver problemas de Ruteo de Vehículos con Múltiples Depósitos y con Múltiples Depósitos y Ventanas de Tiempo. Estas experiencias, si bien han sido exitosas, son todavía casos esporádicos. Se desea generar documentación sobre Minería de Datos, en particular sobre sus técnicas y algoritmos, para considerarlas dentro de los métodos de solución posibles ante los problemas de Investigación Operativa que se plantean. Por otra parte, los problemas de Investigación Operativa, como los problemas que se dan en la actualidad en otras áreas, involucran cada día mayor volumen de datos. Minería de Datos propone técnicas interesantes de manejo de grandes volumenes que resultan atractivas para su aplicación a problemas de Investigación Operativa.

1.2. Objetivos

Los objetivos de este proyecto de grado son :

- Investigar y documentar el Estado del Arte de Minería de Datos. La documentación de Minería de Datos se divide en dos partes. La primera, que involucra el estudio del surgimiento del área, su evolución desde

sus inicios a hoy, la recopilación de métodos y técnicas, la descripción de los algoritmos más representativos de cada técnica y la evaluación de sus perspectivas a futuro. La segunda, es un relevamiento de las herramientas de software disponibles en el mercado.

- Estudiar y evaluar las posibilidades de aplicación de técnicas de Minería de Datos a problemas de Investigación Operativa. Se desea ejemplificar la aplicación de métodos Minería de Datos a problemas reales por la aplicación de estas a un caso de estudio. El caso de estudio debe mostrar como se aplican las técnicas de Minería de Datos a un conjunto de datos de ejemplo y como se pueden aplicar los resultados obtenidos a problemas de Investigación Operativa que se planteen sobre esos datos.

- Desarrollar un nuevo algoritmo de Minería de Datos. Se desarrolla un nuevo algoritmo con el objetivo de mostrar que es sencillo realizar aportes propios a la gran gama de algoritmos disponibles y buscar de esta forma mejores resultados a problemas específicos de nuestro interés.

1.3. Metodología de trabajo

Este trabajo se separa en tres partes bien definidas, que se corresponden con cada uno de los objetivos planteados:

1. Estado del Arte de Minería de Datos.
2. Estudio de aplicación de Minería de Datos a Investigación Operativa.
3. Desarrollo de un nuevo algortimo de Minería de Datos.

1.3.1. Metodología para la investigación y documentación del Estado del Arte de Minería de Datos.

Como paso inicial se optó por la lectura de tres libros que sirvieran para introducirse en el tema Minería de Datos y familiarizarse con la terminología utilizada. Ya que el estudio de Minería de Datos resulta de interés para distintas áreas profesionales, se eligieron estos tres libros con diferentes enfoques, de forma de tener un acercamiento a la Minería de Datos desde diferentes ángulos.

El libro "Principles of Data Mining"de Hand, Mannila y Smyth[4] analiza temas claves de Minería de Datos desde puntos de vista computacionales, en particular de base de datos y estadísticos.

El libro "Discovering Data Mining from concept to implemetation"de Peter Cabena et. al. [8] es de gran utilidad como introducción al tema desde un punto de vista de negocios y proporciona ejemplos de aplicación de técnicas Minería de Datos a casos reales.

El libro "Predictive Data Mining: A practical guide"de Weiss y Indurkhya [7] proporciona un enfoque matemático-estadístico de las técnicas y métodos de Minería de Datos.

Esta etapa introductoria, se continuó con una etapa de búsqueda y recopilación de información en Internet. Durante esta búsqueda se reunieron libros, artículos, reportes técnicos y documentos en general de sitios especializados en tema de Minería de Datos (especialmente artículos disponibles en estos sitios que aún no han sido publicados). También se investigó el software de Minería de Datos disponible en el mercado, por lo tanto se bajaron de la Web e instalaron versiones tanto de software libre como versiones de prueba de software comerciales. A partir de esta información se realizó la documentación del Estado del Arte de Minería de Datos, tomando para ello los que se consideraron los temas de mayor interés y teniendo en cuenta los aportes más interesantes de cada una de las fuentes de información disponible.

1.3.2. Metodología para el estudio de aplicaciones de Minería de Datos a Investigación Operativa.

Se tomó la decisión de escribir una breve reseña de Investigación Operativa (IO) con el fin de presentar brevemente en este documento los temas de interés de esta área, las técnicas hoy día utilizadas y los problemas que suele tratar. Para ello se tomó como libro guía el libro *Introduction to Operations Researchs* de F. Hillier y G. Lieberman [19]. No es del interés de este proyecto escribir un Estado del Arte de Investigación Operativa, principalmente porque este proyecto de grado está dirigido al Departamento de Investigación Operativa del Instituto de Computación. El lector que esté interesado en profundizar los temas de IO resumidos en este trabajo, puede hacerlo en cualquiera de los textos que tratan el tema, algunos de los cuales se detallan en las referencias bibliográficas.

Luego de escrita la reseña de IO se pasó a la tarea de recopilar información concerniente a la aplicación de Minería de Datos a Investigación

Operativa. Se buscó en Internet cualquier tipo de información que relacionara IO con Minería de Datos, pero se encontró poca información comparado con los enormes volumenes de información disponible de Minería de Datos. No fue posible encontrar un libro que refiera a la aplicación conjunta de IO y Minería de Datos, pero si se encontraron artículos aislados en publicaciones de IO que señalan oportunidades de uso de Minería de Datos en IO y analizan casos de uso concretos. La mayoría de la información se obtuvo de los sitios de sociedades de IO y sus respectivas publicaciones. A continuación se nombran las principales fuentes de información utilizadas:

1. IFORS (International Federation of Operational Research Societies), es una federación que nuclea Sociedades de Investigación Operativa de todo el mundo y actualmente cuenta con la afiliación de sociedades de 48 países [4].

2. The OR Society, es una sociedad de investigadores de IO que tiene miembros en 53 países. Provee entrenamiento, conferencias, publicaciones e información en Investigación de Operaciones tanto a miembros como al público en general. Su publicación mensual JORS (The Journal of the Operational Research Society) publica información variada de IO pero se enfoca especialmente en artículos que ilustren la aplicación de IO a problemas reales [5].

3. INFORMS®(The Institute For Operations Research and The Management Sciences), es el instituto de profesionales de Investigación Operativa y Ciencias de Decisión más grande de Estados Unidos, es miembro de IFORS. INFORMS realiza 12 publicaciones, entre ellas ORMS Today e INFORMS Journal on Computing. ORMS Today publica lo último en Investigación Operativa y Ciencias de Decisión: artículos, casos de estudio, revisiones de software y otros. INFORMS Journal on Computing publica artículos que relacionan Investigación Operativa con computación. Desde fines del año 2004 INFORMS habilitó una zona de su sitio web (http://dm.section.informs.org/) especialmente dedicado a la Minería de Datos, para incentivar el trabajo en Minería de Datos en la comunidad de IO [3].

1.3.3. Metodología para el desarrollo de un algoritmo de Minería de Datos.

En primera instancia se estudiaron y compararon dos opciones: desarrollar el algoritmo como un programa independiente o incorporarlo a alguna

herramientas de Minería de Datos de código abierto. Para ello se estudiaron los códigos fuente de los paquetes de software de código abierto, en los cuales se evaluó especialmente la facilidades que estos proveen para añadir nuevos algoritmos. Luego de haber decidido extender el software libre WEKA [45], se revisaron los algoritmos disponibles en el mismo y se pasó a buscar información para la implementación de un nuevo algoritmo. Para ello se buscó en libros y artículos propuestas de algoritmos innovadores.

1.4. Conclusiones

El Estado del Arte de Minería de Datos que se ha documentado logra presentar la Minería de Datos con buen nivel de detalle. Debido a la amplitud del área de Minería de Datos no se ha entrado en detalles de implementación de cada una de sus técnicas y algoritmos, sino de algunas de ellas. Investigación Operativa por su parte es un área igualmente amplia. La amplitud de ambas áreas dificultó tanto la tarea de realizar una comparación entre ellas, como la de estudiar la aplicación de Minería de Datos en IO. Además la decisión tomada al comienzo de este proyecto de no realizar un Estado del Arte de Investigación Operativa, si bien fue acertada desde el punto de vista de dimensionamiento del proyecto, creó una carencia de información igualmente detallada de IO que hubiera facilitado el trabajo planteado de estudiar la aplicación de Minería de Datos en IO. Fue necesario, para cubrir esta carencia, suplementar el trabajo con la lectura de libros de Investigación Operativa que reseñaran las técnicas aplicadas en el área.

El estudio de la aplicación de técnicas de Minería de Datos al caso de estudio, si bien permite ejemplificar el uso de algunas técnicas, no es posible concluir que la aplicación de estas sea extendible a todos los problemas de IO. Se encuentra entonces que es necesario un estudio más extensivo de los problemas de IO y para cada uno de los tipos de problemas estudiar los algoritmos aplicables. Ese análisis, en este nivel de detalle excede los objetivos del proyecto.

1.5. Organización del documento

El presente documento está separado en dos partes. La Parte I trata sobre Investigación Operativa, el uso de Minería de Datos en problemas de Investigación Operativa, el análisis de un caso de estudio y el desarrollo de

un algoritmo de Minería de Datos. En la Parte II se disponen en forma de apéndices el Estado del Arte de Minería de Datos y un par de anexos con definiciones de interés para la lectura de este documento. Las referencias bibliográficas utilizadas se detallan al final de cada una de las partes.

La Parte I está dividida en 5 capítulos que se describen a continuación:

El Capítulo 1 presenta un resumen del trabajo realizado con motivo de este proyecto. Se explica en él la motivación que lo ha originado, los objetivos planteados, las etapas en las cuales se dividió el trabajo y la metodología aplicada en cada una de ellas. También se proporciona una guía para la lectura de este documento mediante una explicación de la organización general del mismo.

El Capítulo 2 aborda varios temas. Primeramente resume que es Investigación Operativa, los problemas que trata y sus técnicas. Luego expone una comparación de Investigación Operativa y Minería de Datos. Finalmente analiza la aplicación de Minería de Datos a Investigación Operativa y viceversa.

El Capítulo 3 presenta un caso de estudio en el cual con el cual se detalla la aplicación de técnicas de Minería de Datos, se presentan los resultados obtenidos para cada una de estas técnicas y se analiza la información resultante puede ser de utilidad en un trabajo de Investigación Operativa.

El Capítulo 4 trata el desarrollo de un nuevo algoritmo de Minería de Datos. En este capítulo se detallan las decisiones más importantes que debieron tomarse previo el desarrollo, se registra la fundamentación teórica que sirvió de guía para el desarrollo y se analizan los resultados obtenidos al aplicar el nuevo algoritmo.

El Capítulo 5 presenta las conclusiones del trabajo realizado y propone posibles trabajos futuros a partir de este, que pueden realizarse en base a los mismos objetivos de este trabajo o nuevos que a partir del mismo se puedan plantear.

La Parte II está dividida en 5 apéndices que se describen a continuación:

El Apéndice A presenta el Estado del Arte de Minería de Datos. Los principales puntos abordados son la reseña histórica del surgimiento de Minería de Datos y su evolución hasta nuestros días, la comparación de Minería de Datos con otras soluciones y el análisis de metodología que esta propone.

En el Apéndice B se describe gran variedad de algoritmos a partir de una clasificación usual de algoritmos de Minería de Datos. Para cada clase de esta clasificación se identifican algunos de los algoritmos más conocidos

y las mejoras que cada uno de ellos ha propuso sobre los existentes en el momento de su creación.

El Apéndice C parte de una clasificación de herramientas de Minería de Datos para presentar algunos de los productos de software de cada tipo que se encuentran disponibles en el mercado, tomando en cuenta tanto los paquetes de tipo comercial como los de código abierto. Además se proponen aquí algunos elementos que se deben tener en cuenta al momento de decidir usar o comprar un software para Minería de Datos.

El Apéndice D analiza los distintos propósitos que pueden llevar al uso de Minería de Datos en un problema del mundo real y muestra ejemplos interesantes de aplicaciones de Minería de Datos en diferentes áreas como la industria, negocios, mercadeo, seguridad y medicina.

En el Apéndice E se describen particularidades de la herramienta de software WEKA, como ser el formato de los archivos de entrada y la interpretación de la salida. Esta información es de utilidad para la interpretación de resultados que se muestran en los Capítulos 3 y 4 de este documento.

El Apéndice F es un anexo que contiene definiciones matemáticas y estadísticas que se mencionan en este documento.

El Apéndice G es un glosario de términos donde se detallaron aquellos términos con los cuales pudieran no estar familiarizados los lectores de este documento, con el objetivo de dar una definición rápida de los mismos.

Al final de cada Parte se detallan las referencias bibliográficas. Se utiliza para las mismas la siguiente notación según sean referencias externas o internas.

- [**Número de referencia**] para referenciar libros, artículos, reportes técnicos, páginas web.
- **Identificador de sección** para referenciar otras secciones de este mismo documento.

Capítulo 2

Investigación Operativa y Minería de Datos

2.1. Introducción

Como se ha explicado anteriormente en el Capítulo 1, no es la intención de este proyecto de grado realizar un Estado del Arte de Investigación Operativa. Con el objetivo de refrescar la memoria del lector este capítulo contiene una breve reseña de que es Investigación Operativa, cuales son los problemas que pueden ser resueltos con técnicas de Investigación Operativa y cuales son las más importantes de estas técnicas. Luego de esta reseña nos adentramos de lleno en el tema que nos interesa: la aplicación de Minería de Datos en Investigación Operativa.

2.2. Investigación Operativa

La Investigación Operativa es parte de las que se conocen como Ciencias de apoyo a la toma de decisiones [1]. Consiste en la aplicación de métodos científicos, técnicas y herramientas a problemas que involucran las operaciones de un sistema.

"La Investigación Operativa es la aplicación del método científico por equipos interdisciplinarios a problemas que comprenden el control y gestión de sistemas organizados (hombre - máquina); con el objetivo de encontrar

[1] Ver definición de Decision Support en el Glosario

soluciones que sirvan mejor a los propósitos del sistema (u organización) como un todo, enmarcados en procesos de toma de decisiones." [10]

La Investigación Operativa surge como parte de la innovación de la revolución industrial, cuando se intentó aplicar por primera vez el método científico a la administración de una empresa. Sin embargo fue durante la segunda Guerra Mundial que los científicos comenzaron a aplicar la Investigación Operativa a problemas reales. Durante la guerra se usó la Investigación Operativa con fines bélicos, en operaciones tácticas y estratégicas [19]. Los primeros problemas fueron de ordenamiento de tareas, reparto de cargas de trabajo, planificación y asignación de recursos en el ámbito militar. Luego de finalizada la guerra, su uso se extendió al ámbito industrial, académico y gubernamental. Hoy en día, la gama de aplicaciones es extraordinariamente amplia, algunas de las áreas que la utilizan son industria manufacturera, transporte, telecomunicaciones, planeación financiera, cuidado de la salud, milicia y servicios públicos.

"La Investigación Operacional se puede pensar que representa el primer acercamiento de tratar de aplicaciones de sistemas a la actividad humana."[13]

Es muy importante entender que aún cuando el problema es único, existen distintas maneras de definirlo, dependiendo de los objetivos que se planteen. Aplicando el método científico, el investigador construirá uno o más modelos del sistema, con sus operaciones correspondientes y sobre cada modelo se realizará la investigación. La Investigación Operativa es la aplicación del método científico a un mismo problema por diversas ciencias y técnicas, en apoyo a la selección de soluciones, en lo posible óptimas.

2.2.1. Problemas típicos de Investigación Operativa

Los problemas de Investigación Operativa tienen elementos comunes fácilmente identificables:

- un objetivo o función,
- variables de decisión y
- un conjunto de restricciones sobre las variables de decisión.

De los problemas tratados en las diferentes áreas con técnicas de Investigación Operativa, se repiten ciertos problemas con características comunes. A continuación describiremos en forma genérica algunos de estos problemas y daremos algunos ejemplos específicos de problemas reales de cada tipo.

- Problemas de secuenciación u ordenamiento: se ocupan de colocar items en determinado orden.
  Ejemplos [19]

  - Problema de generación de horarios para la secuenciación de salidas y llegadas de trenes a una estación.
  - Problema de secuenciación de proyectos que utilizan los mismos recursos limitados.

- Problemas de transporte o de rutas, se ocupan de la construcción de la ruta óptima de un punto origen a uno destino cuando existen varios caminos posibles.
  Ejemplos [19]

  - Problema del viajante. Un viajante tiene que visitar n ciudades, una a la vez antes de volver a la ciudad origen de donde partió.
  - Reparto de mercadería. Un camión parte a entregar mercadería en n destinos, construir su ruta de reparto óptima.

- Problemas de reemplazamiento, se ocupan de decidir el momento adecuado en el tiempo en el que se debe reemplazar maquinaria o equipos debido a fallas o deterioro.
  Ejemplos [19]

  - Determinar cuando reemplazar maquinaria industrial.
  - Determinar cuando reemplazar una computadora.

- Problemas de inventario, se ocupan de determinar la cantidad ideal de cada uno de los productos que se debe tener disponibles en el lugar de venta de los mismos o los depósitos. Los objetivos de este problema son evitar pérdida de ventas por no disponer del stock necesario y evitar altos costos de almacenamiento por exceso de stock. Por lo tanto se debe encontrar el punto de equilibrio para cada producto, que indica el stock ideal del mismo [19].
  Ejemplos [19]

  - Determinación del inventario óptimo de productos terminados para el fábrica de perfumes.
  - Definir la política de inventario de repuestos para una empresa de reparación de computadoras.

- Problemas de colas o líneas de espera, se ocupan de cualquier problema en que se debe esperar para recibir un servicio. La formación de líneas de espera es un fenómeno común que ocurre cuando la demanda por un servicio excede la capacidad para brindarlo. Proporcionar demasiado servicio genera costos excesivos. Carecer de niveles adecuados de servicio genera colas de espera excesivamente largas, que también significan un costo que puede estar dado por ejemplo por la pérdida de clientes. El objetivo final es balancear el costo del servicio con el de la espera del servicio. Los problemas de colas de espera se definen por la cantidad de tiempo entre los arribos a la cola, los tiempos dedicado al servicio, el tamaño de la sala de espera y la disciplina de la cola (método para seleccionar el siguiente a ser atendido)[19].

  Ejemplos [19]

  - Determinar la cantidad óptima de cajas que deben estar habilitadas en un banco en las horas pico de trabajo para que el tiempo de espera promedio sea de 10 minutos.
  - Determinar una política de atención combinada prioridades versus arribo para un centro de salud.

- Problemas de asignación, tratan la asignación de recursos destinados a la realización de tareas [19].

  Ejemplos [19]

  - Asignación de tareas a los trabajadores de una fábrica.
  - Asignación de destinos a vehículos de reparto de mercadería.

2.2.2. Aplicación de Investigación Operativa

Como su nombre lo indica la Investigación Operativa realiza investigación sobre operaciones; por lo tanto se aplica a problemas que se refieren a la conducción y coordinación de operaciones o actividades dentro de una organización. La naturaleza de la organización no es importante, por lo tanto la gama de aplicaciones es extraordinariamente amplia. [19]

El método usado en Investigación Operativa es el método científico, cuyos pasos son:

1. Planteo y análisis del problema
2. Construcción de un modelo

3. Deducción de las soluciones
4. Prueba de los modelos y evaluación de las soluciones
5. Ejecución y verificación de las soluciones

La resolución de problemas es el objetivo de muchas disciplinas. Para resolver un problema apropiadamente primero debemos representarlo. La representación de un problema es por lo tanto un paso crítico en la resolución de problemas que ayuda a encontrar buenas soluciones rápidamente. Existen muchas formas de representar un problema, en Investigación Operativa la representación del problema es cuantitativa [21]. En general, la elección de una representación apropiada del problema influye en la elección del método adecuado para resolverlo. Es por esto que es necesario elegir el método antes de representar el problema. Esta técnica es difícil y ocasiona errores que hacen que en ocasiones se detecte que no es posible representar el problema de la forma que el método necesita. Una vez que el problema está representado, el siguiente paso es buscar algoritmos que se puedan aplicar y elegir uno o más de los que generen las mejores soluciones. Si existen medios para evaluar las soluciones, entonces interesa la óptima; si no hay medios para realizar la evaluación interesa entonces encontrar una solución [21].

La aplicación de Investigación Operativa al estudio de sistemas y resolución de problemas establece un riesgo, el de tratar de manipular los problemas para buscar que se ajusten a un modelo o técnica específica, en vez resolver el problema utilizando los métodos que proporcionan las mejoras soluciones. Para hacer uso apropiado de la Investigación Operativa, es necesario estudiar detenidamente la metodología, para saber cuándo utilizarla o no según las características del problema. La identificación de problemas comunes que se agrupan según la similitud de los modelos y técnicas de resolución resulta útil para que una vez identificado el problema sea más sencillo identificar las técnicas que es viable aplicar para su resolución. Se espera que el investigador de operaciones seleccione del conjunto de la tecnología actual, la que promete mejores resultados en la tarea a ejecutar. Esta tecnología puede encontrarse en cualquier rama de matemáticas, ciencias o ingeniería; no se reduce solamente a aquellas tecnologías creadas por la Investigación de Operaciones [26].

La Investigación Operativa ha tenido un gran impacto en el mejoramiento de la eficiencia de numerosas organizaciones en el mundo, durante este proceso, la Investigación Operativa ha hecho contribuciones significativas al incremento de la productividad de los países. Aunque la mayoría de los

estudio proporcionan beneficios modestos, los mismos reflejan un aumento dramático en estudios grandes y bien diseñados. Se muestran a continuación algunos ejemplos de aplicación a gran escala de Investigación Operativa [19].

- Desarrollo de políticas nacionales de administración de agua.
- Optimización del corte árboles en productos de madera para maximizar su producción.
- Asignación óptima de recursos hidráulicos y térmicos en el sistema nacional de generación de energía.
- Programación de turnos de trabajo en las oficinas de reservas y en aeropuertos para una aerolínea, para cumplir con las necesidades del cliente a un costo mínimo.
- Optimización de mezcla de ingredientes disponibles para que los productos de gasolina cumpla requerimientos de venta y calidad.
- Optimización del diseño de la red nacional de transporte y programación de rutas de envío.

2.2.3. Clasificación de problemas

Debido a la amplitud y profundidad de la colección, cada vez mayor, de métodos y de los campos de uso, Investigación Operativa se divide en especialidades. Asociar un problema con alguna de las técnicas es una tarea compleja. Esta tarea puede simplificarse si consideramos que los proyectos pertenecen a uno de estos tres grupos de métodos [26] :

1. Optimización
2. Simulación
3. Análisis de datos

Optimización

Existe una enorme variedad de actividades de la vida que pueden ser descritas como sistemas, tanto físicos como teóricos. La operación eficiente de esos sistemas usualmente requiere optimizar índices que miden el desempeño del mismo. La Teoría de la Optimización es matemática por naturaleza, involucra la maximización o minimización de una función que representa un sistema, la cual es a veces desconocida. Esto se resuelve encontrando los valores de las variables que hacen que la función alcance su mejor valor

(tanto si este es mínimo como máximo). Se asume que las variables dependen de factores que en ocasiones están bajo el control del usuario del sistema y pueden ser manipulados para obtener mejores resultados.

Algunos de los problemas de la Teoría de Optimización se pueden resolver por las técnicas clásicas del cálculo avanzado, sin embargo, la mayoría de los problemas no satisfacen las condiciones necesarias para que estas técnicas puedan aplicarse. Muchos de los que no cumplen las condiciones, se resuelven mejor si se utilizan métodos diseñados especialmente para el sistema a tratar. Existen innumerables técnicas de Optimización, algunas de ellas comenzaron a usarse cuando el uso de computadoras lo hizo posible.

Problemas típicos de Optimización

- Optimización de diseño o planificación [3]:
  Ejemplos
  - Dimensionamiento de un equipo con costo mínimo.
  - Determinar la mejor ubicación geográfica para la construcción de una planta de producción de celulosa de papel.
- Optimización de operativa y logística [3]:
  Ejemplos
  - Determinar el punto de operación más rentable.
  - Determinar la ruta más corta para la distribución de productos.
- Optimización de asignación de recursos [3]:
  Ejemplos
  - Minimizar las horas de dedicación de un proyecto de desarrollo de software.
  - Maximizar la cantidad producida de un artículo en una industria manufacturera mediante la asignación de maquinaria de apoyo a las tareas críticas del proceso.

Simulación

La técnica de simulación es muy versátil y puede ser usada para investigar casi cualquier tipo de sistema estocástico [2]. . Esta versatilidad la ha hecho

[2]Ver definición de Sistema Estocástico en el Glosario

la técnica más utilizada para este tipo de sistemas, y su popularidad sigue creciendo [19].

Simulación es la construcción de un modelo abstracto que representa un sistema de la vida real y la descripción en el tiempo del mismo. El modelo se escribe como una serie de ecuaciones y relaciones y desde que tecnología lo permite a través de programas de computadora. Para estudiar realmente un sistema se debe poder experimentar con él, y en general ocurre que no es posible experimentar con el sistema real. Las razones de este impedimento pueden ser que el sistema real aún no exista, que la experimentación con el sistema sea muy costosa o que la experimentación con el sistema sea inapropiada. En algunos de estos casos es posible construir un prototipo y probarlo; pero aún cuando sea posible puede ser costoso o poco práctico. Es por esto que los estudios de sistemas se realizan con modelos computacionales de los mismos. La simulación se ha convertido en la rama experimental del área de Investigación Operativa y la misma ha evolucionado a medida que las computadoras y su utilización lo han hecho.

La simulación en computadoras presenta varias ventajas:

- Permite estudiar y experimentar las relaciones e interacciones que existen en un sistema real.
- Ahorra tiempo y dinero de lo que representaría crear un modelo real.
- Independiza el proceso de análisis de los sistemas de la duración real de los eventos que en ellos ocurren. Esto es posible manipulando las variables de tiempo.
- Permite experimentar con sistemas que no existen en la realidad y con ello determinar la viabilidad de su construcción.
- Permite estudiar como reacciona el sistema a los cambios y al mismo tiempo evita los riesgos que ocasionaría probarlo en la realidad.

Problemas típicos de Simulación

- Simulación de funcionamiento de sistemas de colas. Se usa tanto para estudiar y mejorar el diseño de un sistema existente, como para diseñar nuevos sistemas [19].

  Ejemplos

  - Simulación de sistema de colas de un banco con diferentes para determinar la cantidad de óptima de cajeros.

  - Simulación de la sala de espera de un hospital para determinar la cantidad de óptima de cajeros.

- Simulación de sistemas de inventarios. Se usa generalmente para sistemas muy complejos para los cuales resulta imposible o es muy costosa la aplicación de modelos matemáticos. [19]

  Ejemplos

  - Simulación del crecimiento de plantaciones forestales para determinar el período óptimo de corta .
  - Simulación de la operación de una línea de producción para determinar la cantidad de almacenamiento que se debe proporcionar para materiales en proceso [19].

- Simulación de evolución de un proyecto en el tiempo. Se usa para estimar si se cumplirán las fechas límite con el personal asignado[19].

  Ejemplos

  - Simulación de operaciones de mantenimiento para determinar el tamaño óptimo de las brigadas de reparación [19].

- Simulación de sistemas críticos. Se usa en el diseño de nuevos sistemas, para preveer el funcionamiento de un sistema en el futuro, para planificar y probar cambios a un sistema. Generalemente esto se da en sistemas complejos o críticos en los cuales no puede experimientarse sobre el sistema real o aquellos que son muy costosos deben ser estudiados en detalle antes de su construcción.

  Ejemplos

  - Simulación del vuelo de un avión en un túnel de viento durante el diseño de un nuevo modelo de avión [19].
  - Simulación del efectos climáticos extremos para determinar su impacto en zonas de particularidad debilidad.
  - Simulación de la operación de la cuenca hidráulica de un río para determinar la mejor configuración de represas, plantas de energía y sistemas de irrigación. [19]
  - Simulación de la operación global de una empresa para determinar el impacto de cambios de políticas y proporcionar un ámbito para la capacitación de ejecutivos. [19]

Análisis de datos

Mediante el análisis de datos el investigador de operaciones ayuda a encontrar patrones y conexiones en los datos, útiles en tareas de predicción.

2.3. Comparación de Minería de Datos e Investigación Operativa

2.3.1. Similitudes

La Investigación Operativa y la Minería de Datos tienen muchos puntos en común, tanto en cuanto a sus intereses como en la metodología utilizada. Ambas se enfocan al uso de metodología y tecnología como apoyo en el proceso de toma de decisiones. Comparten el uso de técnicas de modelado, análisis de datos, aplicación de tecnología informática, método científico y optimización de sistemas. A la Minería de Datos y la Investigación Operativa los une además el fundamento teórico que les proporciona la estadística.

Tanto en Investigación Operativa como en Minería de Datos el proceso de recolección y preparación de datos es muy importante y puede en ambos ocupar una porción significativa del tiempo total del proyecto.

Investigación Operativa y Minería de Datos comparten un futuro común, el de desarrollar métodos científicos para proveer a los ejecutivos con bases cuantitativas para sus decisiones [30].

2.3.2. Diferencias

La diferencia esencial de Investigación Operativa y Minería de Datos son sus objetivos. Investigación Operativa tiene como objetivo diseñar modelos para entregar al dueño de los datos que lo ayuden en el proceso de toma de decisiones para los que fueron diseñados. La Minería de Datos es mucho menos ambiciosa, pretende solamente proveer al dueño de los datos con nueva información extraída de los mismos. Visto desde el punto de vista del usuario Investigación Operativa le sugiere acciones a seguir en su problema, no es así en el caso de Minería de Datos. Los resultados de Minería de Datos

no sugiere acciones concretas al usuario; si de la información proporcionada por Minería de Datos se desprende una solución o acciones a aplicar, deberá deducirlo el mismo.

En Investigación Operativa la representación del problema es cuantitativa, en cambio en Minería de Datos la representación depende del modelo usado. Puede ser cuantitativa si proviene de la estadística o puede representarse con un grafo, árbol o red si proviene de la inteligencia artificial.

En el área de Investigación Operativa se intenta recabar la mayor cantidad de información del problema antes de comenzar el proceso de resolución y esto incluye la mayor cantidad de datos e información de los datos. Minería de Datos por su parte cuenta con técnicas que le permiten partir de cero, explorando los datos para obtener relaciones entre ellos, sin el aporte de conocimientos previos. Esta es la clase de problemas que en Minería de Datos se conocen como problemas descriptivos.

Los problemas de Investigación Operativa cuentan con los siguientes elementos: un objetivo o función, variables de decisión y un conjunto de restricciones sobre las variables de decisión. Los problemas de Minería de Datos pueden iniciarse únicamente con el establecimiento de un objetivo. En los problemas de tipo descriptivo el objetivo es encontrar relaciones entre los datos, en los problemas predictivos se establece una o más variables para las cuales se desea predecir su valor. En este último caso las variables equivalen a las variables de decisión para un problema de Investigación Operativa , que serán determinadas a partir de los datos y también las restricciones.

Las ciencias de apoyo a la toma de decisiones, entre las cuales se encuentra la Investigación Operativa, tienden a confiar en el conocimiento que aportan los expertos, mientras Minería de Datos intenta extraer este conocimiento de los datos con mayor o menor libertad según se trate de métodos de aprendizaje supervisado o no supervisado. Es decir, que mientras la Investigación Operativa es completamente dirigida por expertos, la Minería de Datos llega en ocasiones a la menor intervención posible.

2.3.3. Conclusiones

Minería de Datos e Investigación Operativa provienen de ramas teóricas comunes, como por ejemplo la estadística; superponen sus interéses respecto a los problemas que abarcan y comparten los medios que utilizan para obtenerlos, como el método científico. Si bien lo anterior es cierto ambas áreas proponen técnicas y métodos que no se utilizan en la otra y utilizan una filosofía de trabajo completamente distinta; Investigación Operativa espe-

ra encontrar una solución a un problema y Minería de Datos simplemente explora los datos para proporcionar más información al dueño de los mismos. La interacción entre ambas áreas en la resolución de problemas permite atacar los mismos desde diferentes enfoques y con más herramientas que si consideramos sólo las de un área. Al ser Minería de Datos un conjunto de técnicas de exploración de los datos es de esperar que se aplique primero y la información obtenida alimente el problema de Investigación Operativa. Es posible también que Investigación Operativa se aplique en ciertas etapas del proceso de Minería de Datos para refinar sus resultados. Si las técnicas se combinan correctamente y el proceso se conduce apropiadamente, es de esperar que se obtengan mejores soluciones que con la aplicación de técnicas de una única área.

2.4. Aplicación de Investigación Operativa a Minería de Datos

Está claro que Investigación Operativa puede jugar un rol significativo en ambos lados del motor de Minería de Datos. Algunos problemas de Minería de Datos pueden ser vistos como problemas de Optimización a gran escala. Por otra parte, muchos de los objetivos de los algoritmos de Minería de Datos pueden expresarse de manera matemática y ser tratados como problemas de Investigación Operativa. En cuanto a la escalabilidad, la habilidad de manejar grandes volumenes de datos, es una tema muy importante en Minería de Datos y la Investigación Operativa puede jugar un rol significativo [27].

La comunidad de Investigación de Operaciones ha hecho recientemente contribuciones significativas en el área de la Minería de Datos, específicamente en el diseño y análisis de los algoritmos de Minería de Datos [23]. Minería de Datos plantea la necesidad de escalar los algoritmos, aplicaciones y sistemas a conjuntos masivos de datos aplicando tecnología informática de alto rendimiento. Muchos de los algoritmos de Minería de Datos comúnmente usados no funcionan adecuadamente con grandes volúmenes. Uno de los objetivos a corto plazo de Minería de Datos debe ser desarrollar versiones escalables de los algoritmos comúnmente usados y desarrollar nuevos algoritmos que puedan ser utilizados en terabytes de datos [25]. El alto rendimiento en Minería de Datos es un tema reciente y representa un desafío por el estado actual de los algoritmos. Algunos algoritmos usan búsquedas heurísticas que involucran muchas pasadas por los datos. Este tipo de búsquedas no

deterministas hacen que la paralelización de los algoritmos sea muy compleja e incluso en algunos casos imposible sin rehacer los algoritmos [25]. Es por esto que, el desarrollo de nuevos algoritmos de Minería de Datos no sólo puede ampliar su aplicación a conjuntos de datos más grandes, sino que también ayudará en el proceso de mejorar el rendimiento de los algoritmos mediante paralelización.

Aplicaciones posibles de Investigación Operativa a Minería de Datos son

- Formulación matemática de máquinas de vectores usada para selección de atributos y clustering [23].
- Metaheurísticas y métodos evolutivos usados para resolver problemas de Minería de Datos [23].
- Uso de algoritmos de Optimización exactos y heurísticos [23]
- Adaptación de técnicas de programación lineal para entrenamiento más rápido de redes neuronales [27].

2.4.1. Aplicación de Optimización en Minería de Datos

La Optimización puede contribuir con la Minería de Datos al formar parte de un proceso de Minería de Datos más grande o en la construcción de nuevas técnicas de Minería de Datos enteramente basadas en Métodos de Optimización [12]. Muchos problemas de Minería de Datos puedes ser formulados como problemas de Optimización e Investigación Operativa posee vasta experiencia en la resolución de estos problemas. La Optimización es uno de los campos de aplicación más comunes de Investigación Operativa. Los métodos de Optimización se pueden aplicar en muchas etapas del proceso de Minería de Datos. Se puede aplicar Optimización a la salida del proceso de Minería de Datos, para optimizar el objetivo deseado. Se puede aplicar a la selección de atributos, que es intrínsecamente un problema combinatorio de optimización [23]. Se puede aplicar en la definición del modelo de Minería de Datos; o bien definiendo esta tarea del proceso como un problema de optimización, o bien aplicando optimización para elegir el mejor entre un conjunto de modelos. También se puede aplicar en clasificación, clustering y descubrimiento de reglas de asociación [12].

Aplicación de Optimización a la salida del proceso de Minería de Datos La Minería de Datos asiste en el descubrimiento automático de patrones de interés en los datos. Los resultados obtenidos por Minería de Datos no son estructurados y requieren de sustancial interpretación [23]. La eficiencia sólo puede lograrse si se combina la salida de la Minería de Datos con métodos de Optimización. La mayoría de las veces los patrones encontrados son demasiados; es necesario trabajar con ellos tomando en cuenta restricciones del sistema, para obtener mejores resultados. Esta tarea puede realizarse en forma eficiente aplicando técnicas de Optimización a los resultados de la Minería de Datos [11]. Como se mencionó anteriormente Minería de Datos no garantiza que el resultado de su aplicación proporcione al usuario acciones a seguir en el problema planteado, simplemente le proporciona información. La aplicación de Optimización a estos resultados puede convertir los resultados de Minería de Datos en acciones a seguir para el usuario.

Ejemplo: Aplicación de Optimización a la salida del proceso de Minería de Datos.

Una oportunidad de usar Optimización luego de ejecutado el proceso de Minería de Datos es en la identificación de el mejor de un conjunto de patrones identificados por el algoritmo de Minería de Datos. Para ello es necesario construir una función que maximice o minimice el valor que será tenido en cuenta para realizar tal determinación. Un caso común de aplicación es en la construcción de perfiles de clientes en CRM, luego de ejecutado un proceso descubrimiento de reglas con este fin se obtiene un número que es en general inmanejable. La ejecución de optimización al resultado identifica los patrones más importantes del conjunto total de patrones descubiertos [12].

Aplicación de Optimización a la selección de atributos Preprocesar los datos para eliminar atributos redundantes e irrelevantes es uno de los pasos más importantes del proceso de Minería de Datos. La selección minuciosa de atributos puede mejorar los tiempos de ejecución y la calidad de los modelos. Usar pocos atributos conduce a menudo a modelos más simples y fáciles de interpretar; y seleccionar los atributos más importantes puede conducir a mejores resultados [23]. Una buena selección de atributos produce modelos más precisos y procesamiento más rápido de los algoritmos [12]. El objetivo de la optimización es encontrar el mejor subconjunto de atributos tal que la aplicación del algoritmo con estos atributos produce resultados similares al de aplicarlo en el conjunto completo.

La aplicación de Optimización implica la definición de un problema de Optimización, Olaffson y Yang (2002) lo definen de esta forma :

$max/min \qquad f(x = (x_1, x_2, ..., x_n))$

$x_j = 0, 1 \qquad \forall j$

$\sum_{i=1}^{n} x_i <= min(f)$

$\sum_{i=1}^{n} x_i >= max(f)$

donde la determinación de que la función f sea de maximización o minimización depende del algoritmo y la variable de decisión x_j se define de la siguiente forma:

$$x_j = \begin{cases} 1, & si\ el\ atributo\ es\ elegido \\ 0, & sino \end{cases}$$

donde j pertenece a A_0, el conjunto de atributos posibles y la definición de un problema de optimización

Ejemplo: Aplicación de Optimización a la selección de atributos. El procesamiento de datos de comportamiento de usuarios en la Web (click-stream data) en CRM tiene como principal objetivo identificar cuando la sesión de un usuario terminará en una compra. Este tipo de problemas requiere dividir los datos recolectados en sesiones y para ello es necesario especificar un criterio de optimización para maximizar las similitudes intersesión y minimizar las diferencias extrasesión. Una forma de medir similitudes y diferencias es considerar ciertas medidas, como el tiempo que emplea un cliente en leer cada página. Debido a que este tipo de procesamientos se realizan en tiempo real se desea seleccionar la menor cantidad de atributos y por supuesto los mejores para poder obtener una respuesta en tiempos dentro de un período de tiempo adecuado [12].

Aplicación de Optimización al desarrollo de nuevas técnicas de Minería de Datos Muchos algoritmos de Minería de Datos se basan en técnicas heurísticas de búsqueda para optimizar una función objetivo. El desarrollo de nuevos algoritmos de optimización contribuye al diseño de mejores algoritmos de Minería de Datos. Existe la esperanza además de que estos nuevos algoritmos sean más fáciles de usar y entender para los usuarios y que esto fomente el uso de las técnicas de Minería de Datos [11].

Ejemplo: Aplicación de Optimización al desarrollo de algoritmos de clustering.

Los algoritmos de clustering constan de dos componentes: un mecanismo de búsqueda que genera candidatos y una función de evaluación que mide la calidad de los candidatos. Los algoritmos de búsqueda usados en Minería de Datos y en particular en clustering son heurísticas. Las búsquedas heurísticas son búsquedas guiadas, ampliamente usadas en la práctica, pero que no garantizan encontrar la solución óptima. En la mayoría de los casos funciona y produce resultados satisfactorios de alta calidad [21]. La Investigación Operativa puede apoyar a la Minería de Datos en el uso de búsquedas exactas o en el mejoramiento de las heurísticas existentes.

El algoritmo de clustering k-medias es una algoritmo de partición por distancia. Por lo tanto la función de optimización será una función de distancias. Se realiza una modificación al algoritmo de clustering k-medias : en cada iteración se agrega un paso en el cual se calcula el punto dentro del cluster que es el más cercano a todos los demás cluster y se cambia el centroide por este punto. Esta técnica de optimización acelera la convergencia a la solución disminuyendo el número de iteraciones necesarias.

2.5. Aplicación de Minería de Datos en Investigación Operativa

La Minería de Datos puede ser usada en Investigación Operativa para complementar los métodos tradicionalmente usados. La aplicación de la Minería de Datos a problemas de optimización se remonta a la década del 60, pero aún hoy muchos de los problemas planteados requieren más investigación [23]. El hecho que desde 1996 se encuentran referencias en IFORS del estudio de Minería de Datos y publicación de artículos de aplicación de Minería de Datos a Investigación Operativa demuestra que el interés de aplicar Minería de Datos a Investigación Operativa está latente, a pesar de que los esfuerzos realizados en ese sentido no son constantes. Dadas las investigaciones actuales y la dirección de desarrollo de soluciones, se ven dos tendencias claras [11] :

- La Minería de Datos se unirá a la Programación Matemática y la Optimización como una tecnología clave para la construcción de sistemas de decisión.

- El uso de Minería de Datos será en el futuro tan común como el uso de que hoy día se da a las bases de datos y las planillas electrónicas.

La Minería de Datos puede jugar un rol muy importante en muchas aplicaciones de Investigación Operativa donde se generan enormes volúmenes de datos. La Minería de Datos puede ser usada para extraer información relevante de esos conjuntos de datos que aporten información nueva y de valor para la posterior aplicación de métodos de Investigación Operativa. La escasez de datos confiables, o la escacez de datos directamente, es un problema común que enfrentan los investigadores de operaciones que tratan de encontrar un buen modelo que funcione en el mundo real. Este proceso se vuelve crítico cuando los datos deben ser descifrados de terabytes de datos almacenados. Las herramientas de Minería de Datos hacen que el acceso y el procesamiento de los datos sea más sencillo, y proporcionan datos más confiables al investigador de operaciones en su tarea de diseñar un modelo. Los nuevos patrones descubiertos por medio de Minería de Datos pueden permitir a los investigadores de operaciones alterar los modelos existentes de forma de mejorar sus resultados.

Una de las fortalezas de la Minería de Datos es el amplio rango de metodologías que pueden ser aplicadas a un conjunto de problemas [72].

Por ejemplo, nuevas asociaciones descubiertas entre las ventas de diferentes productos pueden ser usadas como parte de una nueva estrategia de mercadeo o en el diseño de una política de inventario más eficiente. Minería de Datos puede usarse también para detectar patrones de comportamiento bajo distintas escenarios, y esta información puede ser usada para crear modelos de Optimización basados en estos escenarios. En base a los resultados de la Minería de Datos se pueden desarrollar mejores procedimientos para programar tareas del personal, entregas, producción, etc. [27]

2.5.1. Aplicación de Minería de Datos en Optimización

Un problema de optimización está dado por

- el espacio sobre el cual se realiza la optimización,
- la función objetivo,
- las restricciones impuestas sobre el espacio.

La Minería de Datos puede ayudar en la determinación de cada uno de estos componentes [12].

Aplicación de Minería de Datos al espacio de optimización. Minería de Datos puede colaborar en la especificación del espacio o dominio de la optimización ayudando a crear nuevas variable (atributos) o seleccionar subconjuntos de variables (atributos) existentes sobre las cuales realizar la optimización. También puede ser usado para reducir el espacio de búsqueda simplificando el problema de optimización.

Ejemplo: Simplificación del problema de optimización.

Consideremos una campaña de mercadeo directo que tiene como objetivo enviar por correo electrónico catálogos a los clientes. Este problema es determinar la secuencia óptima de catálogos que recibe cada cliente y fue abordado por Campbell et al. en 2001 [18]. La aplicación específica consiste de 7 millones de clientes y 40 catalogos. El problema tiene 280 millones de variables de decisión sin considerar el orden de envío de los correos. Aplicando clustering a los clientes se generan grupos similares basados en ciertas características de los clientes obtenidas de datos históricos, como por ejemplo la ganancia esperada. El número de clusters generados a partir de 7 millones de clientes fue de 2000 y se redujo la cantidad de variables de decisión a 80.000 [12].

Aplicación de Minería de Datos para determinar la función objetivo. La función objetivo por lo general tiene parámetros que deben ser estimados. En dominios donde existen datos históricos, se puedan especificar modelos de Minería de Datos predictivos para estimar el valor de estos parámetros [12].

Ejemplo: Determinación de parámetros de la función objetivo.

Una compañía productora de lilas planta cada año más de 3.5 millones de bulbos de más de 50 variedades de lilas en un invernadero de 20.000 metros cuadrados. Es necesario determinar cuantos y que tipo de bulbos plantar en el invernadero cada semana para maximizar los beneficios durante el año. La especificación de la función objetivo require la estimación de ingresos por tipo de flor en el año. Se puede utilizar entonces un modelo predictivo basado en datos de ventas históricos para estimar las ventas por tipo de flor [12].

Aplicación de Minería de Datos para determinar restricciones del espacio. En general, cualquier técnica de Minería de Datos para descubrimiento de patrones sirve para determinar posibles restricciones. El resultado

del descubrimiento de patrones debe ser analizado por expertos que decidan cuales de los patrones vale la pena considerar como restricciones. También puede realizarse este procedimiento sobre un problema de optimización en el cual ya se hayan especificado restricciones con el onjetivo de modificarlas. Las restricciones, al igual que la función objetivo tienen parámetros que deben ser estimados.

Ejemplo: Determinación de parámetros de las restricciones.

Sery et al. (2001) [29] describen una solución a un problema de optimización de la compañía BASF. El problema trata de minimizar los costos de distribución de los bienes y al mismo tiempo mejorando el servicio al cliente. A medidados de 1990, BASF enviaba 1.6 billones de dólares en bienes a clientes de una red de 135 locaciones y necesitaba minimizar los costos de distribución sin afectar el servicio. Una restricción en esta formulación es que la demanda de cada cliente en cada locación es satisfecha. El valor de la demanda de cada lugar es un parámetro en la restricción que es necesario estimar. Los métodods de Minería de Datos predictivos pueden ser usados para estimar este parámetro.

2.6. Conclusiones

Mediante el descubrimiento de patrones, las técnicas de Minería de Datos pueden sugerir nuevos modos de alcanzar viejos objetivos. Pueden permitir la formulación de mejores y más sofisticados modelos al proporcionar nueva información. En general, la ganancia de incorporar nueva información a los modelos es significativamente mayor que la que puede hacerse mejorando soluciones existentes. Por lo tanto, esta opción permite al investigador de operaciones brindar mejor apoyo en la toma de decisiones [27].

Los algoritmos de Minería de Datos conforman un grupo heterogéneo de algoritmos apenas ligados por el objetivo común de generar mejor información. Por su parte Investigación Operativa refiere a hacer el mejor uso de la información disponible [27]. Por lo tanto si los resultados de la Minería de Datos se toman como materia prima para los problemas de Investigación Operativa, los resultados obtenidos por esta serán mejores que si se aplicara en forma aislada.

Existe una clara necesidad de explotar los datos disponibles y desarrollar tecnologías para construir la siguiente generación de aplicaciones comerciales y científicas, que puedan combinar métodos guiados por los datos con

conocimiento específico del área de investigación. Cuando se combina apropiadamente la Minería de Datos con el conocimiento específico de las fuentes de datos, las soluciones de Minería de Datos tienen el potencial de revolucionar el proceso de descubrimiento científico, verificación y predicción [24]. Minería de Datos e Investigación Operativa proveen juntos la combinación óptima de datos y análisis conducido por expertos [24]. Esta combinación está dando forma a lo que se conoce como Tecnologías de Análisis de Información.

Capítulo 3

Caso de estudio

En este capítulo se ejemplifica mediante un caso de estudio la aplicación de técnicas de Minería de Datos y se muestra como puede usarse en un proyecto de Investigación Operativa. Para ejemplificar la aplicación de algoritmos se utilizan algunos de los algoritmos proporcionados por el paquete WEKA y se explica los cálculos realizados por cada algoritmo para obtener una solución. Por otra parte se evalúa el significado de los resultados obtenidos y como pueden ser estos usados en un problema de Investigación Operativa planteado a partir de la fuente de datos propuesta.

A pesar de que Minería de Datos se propone mayoritariamente para trabajar con grandes volumenes de datos, para el caso de estudio se eligieron fuentes pequeñas. Esto se debe a que una fuente pequeña permite realizar paso a paso los cálculos de los algorimos de Minería de Datos con fluidez y hace que los resultados sean más fáciles de interpretar.

3.1. Presentación del caso

El caso de estudio está dado por una fuente de datos de 244 registros con 3 columnas que corresponden a los siguientes datos: día, eventos de zona 1 y eventos de zona 2 (Tabla 3.1). El campo día es el número de día del mes y toma valores entre 1 y 31. Representa el día del mes en que se midieron los eventos de zona 1 y eventos de zona 2. Los datos están ordenados en la fuente de datos por mes y día, aunque el mes no es una campo de la fuente.

Día	Eventos Zona 1	Eventos Zona 2
1	24388	32301
2	27495	21183
...	...	...
31	36600	38573
1	30391	24060
...	...	...

Tabla 3.1: Planilla Original PO.

Desde el punto de vista de Investigación Operativa este caso puede plantearse como

- un problema de pronóstico de los eventos de zona 1 y zona 2 para un día del mes,

- un problema de simulación que permita visualizar como progresan los eventos de zona 1 y zona 2 en el tiempo,

- un problema análisis de decisiones que muestre el comportamiento de los datos y permita al dueño de los mismos tomar decisiones a futuro a partir de ello.

3.2. Uso de WEKA

El paquete de Minería de Datos WEKA se eligió para realizar con él la prueba de aplicación de algoritmos al caso de estudio. Las razones para elegir este paquete de software y no otro son las mismas que se explican en la Sección 4.2.3 y que llevaron luego a extender WEKA y no otro paquete de software.

En el Anexo E se describen los archivos de entrada de WEKA, las medidas proporcionadas en la salida de ejecución de los algoritmos y una guía para la interpretación de los resultados. En la Sección 3.3 se anexan salidas los resultados obtenidos para el caso de estudio con algunos algoritmos de WEKA, por lo cual se recomienda la lectura previa de la guía para la interpretación de los resultados.

3.3. Aplicación de Minería de Datos al caso de estudio

Para aplicar Minería de Datos a este caso de estudio seguiremos paso a paso el proceso de Minería de Datos descrito en la Sección A.6 que está compuesto por los siguientes pasos:

- Definición del problema
- Preparación de los datos
- Construcción de modelos
- Validación de los modelos
- Puesta en producción los modelos
- Administración de los meta datos

3.3.1. Definición del problema

Al aplicar Minería de Datos la primera decisión a tomar es que tipo de minería de datos aplicar: predicción o descripción. Esta decisión acota el universo de algoritmos y métodos útiles para el problema.

Para este caso de estudio se pueden plantear problemas de Minería de Datos de ambos tipos. Un problema de predicción es el de predecir el número de eventos en el futuro. Un problema de descripción es el de describir como los atributos de la fuente de datos se relacionan entre sí. En ambos casos se aplican técnicas de minería de datos que permiten descubrir las interrelaciones entre los atributos de la fuente de datos. En el caso de predicción esta información sirve para expresar un modelo aplicable a datos futuros; en el caso de descripción es esta información el resultado definitivo.

3.3.2. Preparación de los datos

A partir de esta planilla de datos podemos identificar los atributos día, zona y eventos.

Atributo	Valores posibles
Día	1..31
Zona	1,2
Eventos	entero

Organizar los datos La planilla original no tiene la estructura adecuada para efectuar Minería de Datos. Las columnas "cantidad de eventos zona 1" y "cantidad de eventos zona 2" son en realidad la combinación de 2 atributos: la zona y la cantidad de eventos. Para convertir la planilla original en una en que los casos sean las filas y las columnas sean los atributos son necesarias algunas modificaciones. Para ello, se realizan los siguientes cambios y se genera una nueva planilla que está compuesta por las columnas día, zona y cantidad de eventos. Finalmente se obtiene la Tabla 3.2

- Agregar la columna "zona" que toma los valores 1 o 2.
- Transformar las columnas "cantidad de eventos zona 1" y "cantidad de eventos zona 2" en la columna "cantidad de eventos".

Día	Zona	Eventos
1	1	24388
2	1	27495
...	...	...
31	1	36600
...	...	...
1	2	32301
2	2	21183
...	...	...
31	2	38573

Tabla 3.2: Planilla Transformada T1.

A partir de la fuente de datos original podemos dividirla en datos de entrenamiento y datos de prueba separando para entrenamiento por ejemplo los registros correspondientes al primer mes (días del 1..31).

Reducir los datos No se aplica reducción de datos.

Limpiar los datos No es necesario realizar limpieza de datos.

3.3.3. Construcción de modelos

Las técnicas de modelado de árboles de decisión son los métodos lógicos más usados, no es necesario realizar hipótesis dobre los atributos. Se construyen 2 modelos de árboles de decisión para el caso de estudio, uno

para el algoritmo ID3 y otro para su sucesor C4.5. Por otra parte las redes neuronales son interesantes para este caso de estudio en que no se tiene más información de la fuente de datos que la fuente de datos misma. Las redes neuronales no necesitan más datos, se modelan únicamente en base a los datos de entrada que recibe y los de salida que genera. Es por esto que se contruyen 3 modelos de redes neuronales de diferentes estructuras.

Por más información de la construcción de modelos se recomienda la lectura del Anexo A Sección A.6.3.

Modelos de árboles de decisión

Modelo 1: Clasificación por árboles de decisión ID3 El algoritmo ID3 [1] consiste en la construcción iterativa de un árbol de decisión, basándose en información teórica y en la minimización del número esperado de comparaciones. ID3 presena lalimitación de que sólo permite su aplicación en conjunto de datos discretos, por lo tanto para aplicarlo es necesario procesar los datos para que tanto la variable a predecir, cantidad de eventos, como las variables de entrada sea atributos categóricos.

Para convertir el atributo cantidad de eventos en categórico se aplica la técnica de discretización. Se definen rangos de valores de eventos y se asocian los rangos con un valor categórico. Se puede discretizar en tantos intervalos como sea necesario según el problema y la variabilidad de los valores del atributo no categórico. En este caso se definen 5 rangos y los mismos se identifican con las letras de la A a la E. Se elige un número reducidos de rangos para que los algoritmos sean fáciles de aplicar y los resultados sean fáciles de interpretar y además porque no tenemos elementos para determinar la cantidad más adecuada de rangos ya que la naturaleza de los datos es desconocida. Si fuera necesario un análisis más profundo de estos datos se podría volver a este paso y definir una mayor cantidad de rangos y proceder nuevamente en la aplicación de los algoritmos de Minería de Datos.

Clase	Rango de eventos
A	0 ... 5.000
B	5.001 ... 10.000
C	10.001 ... 20.000
D	20.001 ... 30.000
E	más de 30.000

En la Tabla 3.3 se puede ver el resultado de transformar los datos de la Tabla 3.2 según este criterio.

Día	Zona	Eventos
1	1	D
2	1	D
...	...	...
31	1	E
...	...	...
1	2	E
2	2	D
...	...	...
31	2	E

Tabla 3.3: Planilla Transformada T2.

Variables de entrada : Día, Zona
Variables de salida : Eventos

Se continuación se detalla la aplicación del procedimiento de generación planteado en la Sección B.9.1 a este caso de estudio. Este procedimiento permite evaluar la calidad de cada uno de los atributos: día y zona. Para ello se calcula la ganancia de información debida a cada uno de los atributos y se elige en cada iteración del algoritmo el atributo que proporciona mayor ganancia de información para generar el próximo nivel de decisión del árbol.

Para calcular la ganancia de información de un atributo X, dada por la fórmula $Ganancia(X,T) = Info(T) - Info(X,T)$, es necesario calcular $Info(X,T) = \sum_{i=1}^{n} |T_i|/|T| * Info(T_i)$ donde $Info(T) = E(P)$ y $E(P)$ es la entropía de P, cuya respectiva fórmula es $E(P) = -\sum_{i=1}^{n}(p_i * log(p_i))$ siendo P una distribución de probabilidad $P = (p_1, p_2, ..., p_n)$

Clase	Cantidad de casos
A	53
B	48
C	57
D	43
E	43
Total de casos	244

Tabla 3.4: Tabla de casos por clase según casos de entrenamiento.

En primera instacia se halla la distribución de probabilidad de las clases definidas. En función de los datos de la Tabla 3.4 se deduce el vector de distribución $P = (53/244, 48/244, 57/244, 43/244, 43/244)$. Luego se aplica la ecuación de entropía

$E(P) = -\sum_{i=1}^{n}(p_i * log(p_i))$

$E(P) = -[(53/244) * log(53/244) + (48/244) * log(48/244) + (57/244) * log(57/244) + (43/244) * log(43/244) + (43/244) * log(43/244)]$

$E(P) = 2{,}313$

$Info(T) = 2{,}313$

A continuación se toman ambos atributos: zona y día y se calcula la ganancia de información de cada uno de ellos. La ganancia de información se usa para comparar los atributos y con esta información elegir el atributo que proporciona mayor ganancia para usar en el siguiente nivel de construcción del árbol de decisión. La ganancia de información se define como la diferencia entre la información necesaria para identificar un elemento de T y la información necesaria para identificar un elemento de T luego de obtenido el valor del atributo X.

1) Atributo zona

	zona=1	zona=2
Clase A	27	26
Clase B	26	22
Clase C	27	30
Clase D	20	23
Clase E	22	21
Total de casos	122	122

Tabla 3.5: Tabla de casos por clase y zona.

Como se ve en la Tabla 3.5 el atributo zona divide T (el conjunto total de la muestra) en dos subconjuntos; T_1 cuando zona = 1 y T_2 cuando zona = 2. La distribución de probabilidad de la división de clases producida por el atributo zona es para zona = 1 $PT1 = (27/122, 26/122, 27/122, 20/122, 22/122)$ y para zona = 2 $PT2 = (26/122, 22/122, 30/122, 23/122, 21/122)$.

Se aplica entonces la ecuación

$Info(X,T) = \sum_{i=1}^{n} |T_i|/|T| * Info(T_i)$

para calcular la información generada a partir de un atributo.

$Info(Zona, T) = |T_1|/|T| * Info(T_1) + |T_2|/|T| * Info(T_2)$
$Info(Zona, T) = (122/244) * Info(T_1) + (122/244) * Info(T_2)$
$Info(Zona, T) = (1/2)[Info(T_1) + Info(T_2)]$

$Info(T_1) = E(PT1) = 2{,}312$
$Info(T_2) = E(PT2) = 2{,}309$

$Info(Zona, T) = (1/2)(2{,}312 + 2{,}309) = 2{,}3105$
$Ganancia(Zona, T) = Info(T) - Info(Zona, T) = 2{,}313 - 2{,}311 = 0{,}002$

2) Atributo día

De igual forma que para el atributo zona, se calculan los vectores de distribución de probabilidad para el atributo día. Según los datos de la Tabla 3.6 el atributo día divide T en 31 subconjuntos T_1 cuando día = 1, T_2 cuando día = 2, sucesivamente hasta T_{31} cuando día = 31.

Día	01	02	03	04	05	06	07	08	09	10
Clase A	0	0	0	0	0	4	2	3	3	2
Clase B	0	0	0	0	0	1	2	1	3	2
Clase C	0	0	0	0	0	3	4	4	2	4
Clase D	3	3	4	2	3	0	0	0	0	0
Clase E	5	5	4	6	5	0	0	0	0	0
Total de casos	8	8	8	8	8	8	8	8	8	8

Día	11	12	13	14	15	16	17	18	19	20
Clase A	1	3	4	2	4	1	4	2	4	1
Clase B	3	1	2	3	2	5	2	4	3	3
Clase C	4	4	2	3	1	1	2	2	1	4
Clase D	0	0	0	0	1	1	0	0	0	0
Clase E	0	0	0	0	0	0	0	0	0	0
Total de casos	8	8	8	8	8	8	8	8	8	8

Día	21	22	23	24	25	26	27	28	29	30	31
Clase A	0	4	2	4	3	0	0	0	0	0	0
Clase B	2	2	4	1	2	0	0	0	0	0	0
Clase C	6	2	2	3	3	0	0	0	0	0	0
Clase D	0	0	0	0	0	8	3	7	5	4	0
Clase E	0	0	0	0	0	0	5	1	3	4	4
Total de casos	8	8	8	8	8	8	8	8	8	8	4

Tabla 3.6: Tabla de cantidad de eventos por día.

Por la gran cantidad de valores existentes resulta más cómodo expresar los vectores como las columnas de la Tabla 3.7 de la forma $(p_1, p_2, p_3, p_4, p_5)$.

	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}
p_1 (A)	0	0	0	0	0	0,500	0,250	0,375	0,375	0,250
p_2 (B)	0	0	0	0	0	0,125	0,250	0,125	0,375	0,250
p_3 (C)	0	0	0	0	0	0,375	0,500	0,500	0,250	0,500
p_4 (D)	0,375	0,375	0,500	0,250	0,375	0	0	0	0	0
p_5 (E)	0,625	0,625	0,500	0,750	0,625	0	0	0	0	0

	T_{11}	T_{12}	T_{13}	T_{14}	T_{15}	T_{16}	T_{17}	T_{18}	T_{19}	T_{20}
p_1 (A)	0,125	0,375	0,500	0,250	0,500	0,125	0,500	0,250	0,500	0,125
p_2 (B)	0,375	0,125	0,250	0,375	0,250	0,625	0,250	0,500	0,375	0,375
p_3 (C)	0,500	0,500	0,250	0,375	0,125	0,125	0,250	0,250	0,125	0,500
p_4 (D)	0	0	0	0	0,125	0,125	0	0	0	0
p_5 (E)	0	0	0	0	0	0	0	0	0	0

	T_{21}	T_{22}	T_{23}	T_{24}	T_{25}	T_{26}	T_{27}	T_{28}	T_{29}	T_{30}	T_{31}
p_1 (A)	0	0,500	0,250	0,500	0,375	0	0	0	0	0	0
p_2 (B)	0,250	0,250	0,500	0,125	0,250	0	0	0	0	0	0
p_3 (C)	0,750	0,250	0,250	0,375	0,375	0	0	0	0	0	0
p_4 (D)	0	0	0	0	0	1,000	0,375	0,875	0,625	0,500	0
p_5 (E)	0	0	0	0	0	0	0,625	0,125	0,375	0,500	0,500

Tabla 3.7: Tabla de vector distribución probabilidad por día.

	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}
$p_1 * log(p_1)$	0	0	0	0	0	-0,5	-0,5	-0,531	-0,531	-0,5
$p_2 * log(p_2)$	0	0	0	0	0	-0,375	-0,5	-0,375	-0,531	-0,5
$p_3 * log(p_3)$	0	0	0	0	0	-0,531	-0,5	-0,5	-0,5	-0,5
$p_4 * log(p_4)$	-0,531	-0,531	-0,5	-0,5	-0,531	0	0	0	0	0
$p_5 * log(p_5)$	-0,424	-0,424	-0,5	-0,312	-0,424	0	0	0	0	0
$Info(T_i)$	0,955	0,955	1	0,812	0,955	1,406	1,5	1,406	1,562	1,5

	T_{11}	T_{12}	T_{13}	T_{14}	T_{15}	T_{16}	T_{17}	T_{18}	T_{19}	T_{20}
$p_1 * log(p_1)$	-0,375	-0,531	-0,5	-0,5	-0,5	-0,375	-0,5	-0,5	-0,5	-0,375
$p_2 * log(p_2)$	-0,531	-0,375	-0,5	-0,531	-0,5	-0,424	-0,5	-0,5	-0,531	-0,531
$p_3 * log(p_3)$	-0,5	-0,5	-0,5	-0,531	-0,375	-0,375	-0,5	-0,5	-0,375	-0,5
$p_4 * log(p_4)$	0	0	0	0	-0,375	-0,375	0	0	0	0
$p_5 * log(p_5)$	0	0	0	0	0	0	0	0	0	0
$Info(T_i)$	1,406	1,406	1,5	1,562	1,75	1,549	1,5	1,5	1,406	1,406

	T_{21}	T_{22}	T_{23}	T_{24}	T_{25}	T_{26}	T_{27}	T_{28}	T_{29}	T_{30}	T_{31}
$p_1 * log(p_1)$	0	-0,5	-0,5	-0,5	-0,531	0	0	0	0	0	0
$p_2 * log(p_2)$	-0,5	-0,5	-0,5	-0,375	-0,5	0	0	0	0	0	0
$p_3 * log(p_3)$	-0,312	-0,5	-0,5	-0,531	-0,531	0	0	0	0	0	0
$p_4 * log(p_4)$	0	0	0	0	0	0	-0,531	-0,169	-0,424	-0,5	0
$p_5 * log(p_5)$	0	0	0	0	0	0	-0,424	-0,375	-0,531	-0,5	-0,5
$Info(T_i)$	0,812	1,5	1,5	1,406	1,562	0	0,955	0,544	0,955	1	0,5

Tabla 3.8: Tabla de $Info(T_i)$ para cada día.

A partir de los valores $Info(T_i)$ precalculados en la Tabla , a continuación se calcula la información ganada por el atributo día $Info(Dia, T)$.

$Info(X, T) = \sum_{i=1}^{n} |T_i|/|T| * Info(T_i)$
$Info(Dia, T) = |T_1|/|T| * Info(T_1) + |T_2|/|T| * Info(T_2) + ... + |T_{31}|/|T| * Info(T_{31})$
$Info(Dia, T) = (8/244)[Info(T_1) + Info(T_2) + ... + Info(T_{30})] + (4/244) * Info(T_{31})$
$Info(Dia, T) = (8/244)[Info(T_1) + Info(T_2) + ... + (1/2) * Info(T_{31})]$

$Info(T_1) = E(PT1)$
$Info(T_2) = E(PT2)$
....
$Info(T_{30}) = E(PT30)$
$Info(T_{31}) = E(PT31)$

$Info(Dia, T) = (8/244)[Info(T_1) + Info(T_2) + ... + (1/2) * Info(T_{31})]$
$Info(Dia, T) = 1{,}2298$
$Ganancia(Dia, T) = Info(T) - Info(Dia, T) = 2{,}313 - 1{,}229 = 1{,}084$

El algoritmo propone elegir en cada paso el atributo que proporcione mayor ganancia. Por lo tanto, evaluando las ganancias de los atributos zona y día, el atributo día resulta ser el mejor separador de los datos y es el elegido para construir el primer nivel del árbol.

Esta técnica de cálculo se aplica sucesivamente hasta que se construye el árbol de decisión completo.

Resultados obtenidos por ID3 de WEKA Cabezal del archivo de entrada ARFF:

```
@relation eventos
@attribute dia 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,
        18,19,20,21,22,23,24,25,26,27,28,29,30,31
@attribute zona 1,2
@attribute eventos A,B,C,D,E
```

Salida de ejecución de ID3 de WEKA:
=== Run information ===

Scheme: weka.classifiers.trees.Id3
Relation: eventos

Instances: 244
Attributes: 3
dia
zona
eventos
Test mode: evaluate on training data

=== Classifier model (full training set) ===
Id3

dia = 1 — zona = 1: E — zona = 2: D
dia = 2 — zona = 1: D — zona = 2: E
dia = 3 — zona = 1: E — zona = 2: D
dia = 4: E
dia = 5 — zona = 1: D — zona = 2: E
dia = 6 : A
dia = 7 — zona = 1: B — zona = 2: C
dia = 8 — zona = 1: C — zona = 2: A
dia = 9 — zona = 1: A — zona = 2: C
dia = 10— zona = 1: B — zona = 2: A
dia = 11— zona = 1: B — zona = 2: C
dia = 12— zona = 1: C — zona = 2: A
dia = 13— zona = 1: A — zona = 2: C
dia = 14— zona = 1: B — zona = 2: C
dia = 15 : A
dia = 16 : B
dia = 17— zona = 1: A — zona = 2: B
dia = 18— zona = 1: A — zona = 2: B
dia = 19: A
dia = 20— zona = 1: B — zona = 2: C
dia = 21— zona = 1: C — zona = 2: B
dia = 22: A
dia = 23: B
dia = 24— zona = 1: C — zona = 2: A
dia = 25— zona = 1: C — zona = 2: A
dia = 26: D
dia = 27— zona = 1: D — zona = 2: E
dia = 28: D
dia = 29: D
dia = 30: D
dia = 31: E

Interpretación: El modelo de clasificación es un árbol de decisión construído con el clasificador ID3. Las líneas superiores muestran como el clasificador utiliza los atributos para tomar decisiones. Las hojas indican que clase se le asigna a una instancia que llegue hasta ellas.

Time taken to build model: 0.03 seconds

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

Correctly Classified Instances 149 (61.0656 %)
Incorrectly Classified Instances 95 (38.9344 %)
Kappa statistic 0.5121
Mean absolute error 0.1844
Root mean squared error 0.3037
Relative absolute error 57.815 %
Root relative squared error 76.0386 %
Total Number of Instances 244

Interpretación: El sumario da los niveles de error obtenidos al aplicar el clasificador a los datos de entrenamiento con los que fue construído. Para nuestro propósito los datos más interesantes son el número de instancias correcta e incorrectamente clasificadas. La definición de todos los indicadores se encuentra en el Anexo E.

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.698	0.162	0.544	0.698	0.612	A
0.521	0.117	0.521	0.521	0.521	B
0.491	0.086	0.636	0.491	0.554	C
0.791	0.09	0.654	0.791	0.716	D
0.581	0.035	0.781	0.581	0.667	E

=== Confusion Matrix ===

A	B	C	D	E	classified as
37	7	9	0	0	A
16	25	7	0	0	B
14	15	28	0	0	C
1	1	0	34	7	D
0	0	0	18	25	E

Interpretación: La matriz de confusión muestra para cada clase, como se clasificaron las instancias. Por ejemplo, para la clase “B”, 25 instancias se clasificaron correctamente pero 7 se clasificaron como clase “A”, 15 como clase “C” y 1 como clase “D”.

En las Figuras 3.1 y 3.2 se grafican los resultados obtenidos con ID3 de WEKA. La clase A corresponde a la clase 1 en la gráfica, la clase B a la clase 2 y así sucesivamente hasta la clase E que corresponde al 5. La Figura 3.1 muestra como afecta el atributo día el valor de resultado y la Figura 3.2 muestra como afecta la combinación de atributo día y zona los resultados.

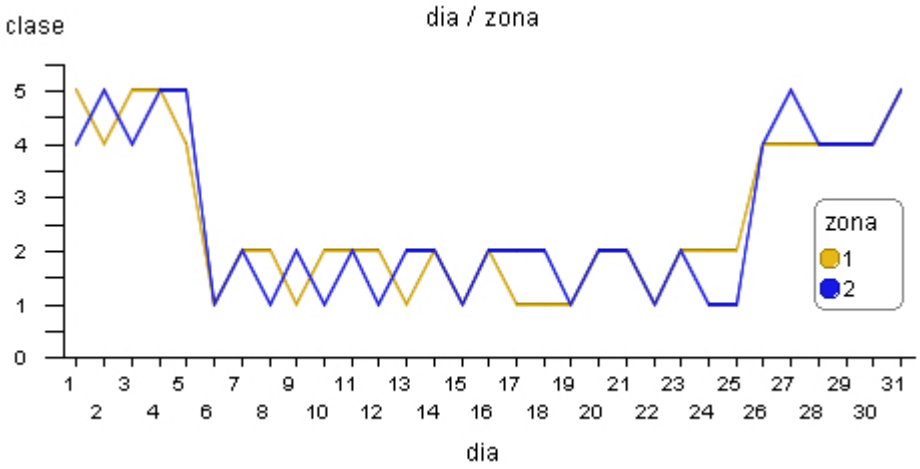

Figura 3.1: Resultados ID3 por día y zona.

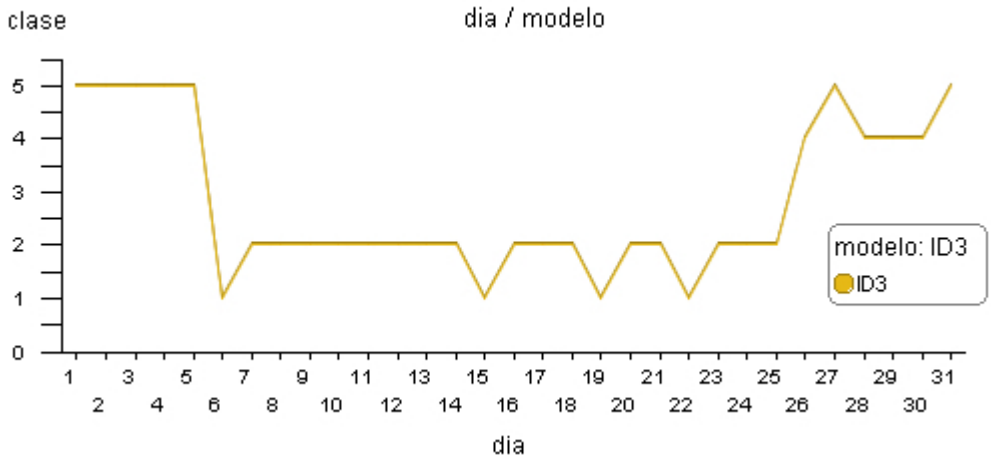

Figura 3.2: Resultados ID3 por día.

Es evidente de la comparación de ambas gráficas que el atributo zona no influye en el comportamiento global de la función. Esto comprueba lo que se determinó mediante la aplicación del algoritmo ID3, el atributo día es el que proporciona una mejor división de los datos.

Modelo 2: Clasificación por árboles de decisión C4.5 El algoritmo C4.5 es una mejora del algoritmo ID3[2] que permite trabajar con datos continuos entre otras mejoras. Por lo tanto para aplicar C4.5 al caso de estudio no es necesario procesar los datos para que la variable a predecir sean categóricas.

Dada la gran cantidad de valores posibles del atributo día se puede elegir entre representarlo en forma categórica con sus 31 valores o como un número real. A continuación figuran los resultados de ejecutar el algoritmo J48 de WEKA (versión de WEKA del algoritmo C4.5). La Prueba 1 se realiza con el atributo día definido como atributo categórico y la Prueba 2 con el atributo día definido como atributo real.

Resultados obtenidos por J48 de WEKA **Prueba 1:** Atributo día categórico. Cabezal del archivo de entrada ARFF:

```
@relation eventos

@attribute dia 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,
        21,22,23,24,25,26,27,28,29,30,31
@attribute zona 1,2
@attribute eventos A,B,C,D
```

Salida de ejecución de J48 de WEKA: el árbol decisión obtenido es idéntico al obtenido con el Modelo 1 - ID3.

Prueba 2: Atributo día real. Cabezal del archivo de entrada ARFF:

```
@relation eventos

@attribute dia real
@attribute zona 1,2
@attribute eventos A,B,C,D
```

Salida de ejecución de J48 de WEKA:
=== Run information ===

Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2
Relation: eventos
Instances: 244
Attributes: 3
dia
zona
eventos
Test mode: evaluate on training data

=== Classifier model (full training set) ===

J48 pruned tree
dia ≤ 25
— dia ≤ 5: E (40.0/15.0)
— dia > 5
— — zona = 1
— — — dia ≤ 19
— — — — dia ≤ 12
— — — — — dia ≤ 11: B (24.0/14.0)
— — — — — dia > 11: C (4.0/1.0)
— — — — dia > 12: A (28.0/14.0)
— — — dia > 19: C (24.0/10.0)
— — zona = 2
— — — dia ≤ 14: C (36.0/17.0)
— — — dia > 14
— — — — dia ≤ 21: B (28.0/17.0)
— — — — dia > 21: A (16.0/8.0)
dia > 25
— dia ≤ 30: D (40.0/14.0)
— dia > 30: E (4.0)

Number of Leaves : 10
Size of the tree : 19
Time taken to build model: 0.05 seconds

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

Correctly Classified Instances 134 (54.918 %)
Incorrectly Classified Instances 110 (45.082 %)
Kappa statistic 0.4343
Mean absolute error 0.2204
Root mean squared error 0.332
Relative absolute error 69.0912 %
Root relative squared error 83.1238 %

Total Number of Instances 244

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.415	0.115	0.5	0.415	0.454	A
0.438	0.158	0.404	0.438	0.42	B
0.632	0.15	0.563	0.632	0.595	C
0.605	0.07	0.65	0.605	0.627	D
0.674	0.075	0.659	0.674	0.667	E

=== Confusion Matrix ===

A	B	C	D	E	classified as
22	13	18	0	0	A
17	21	10	0	0	B
5	16	36	0	0	C
0	2	0	26	15	D
0	0	0	14	29	E

En las Figuras 3.3 y 3.4 se grafican los resultados obtenidos con J48 de WEKA. La clase A corresponde a la clase 1 en la gráfica, la clase B a la clase 2 y así sucesivamente hasta la clase E que corresponde al 5. La Figura 3.1 muestra como afecta el atributo día el valor de resultado y la Figura 3.2 muestra como afecta la combinación de atributo día y zona los resultados.

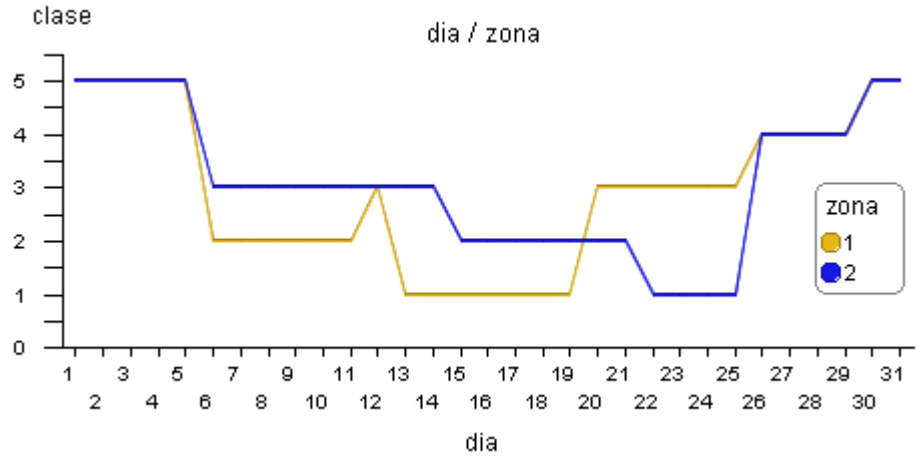

Figura 3.3: Resultados J48 por día y zona.

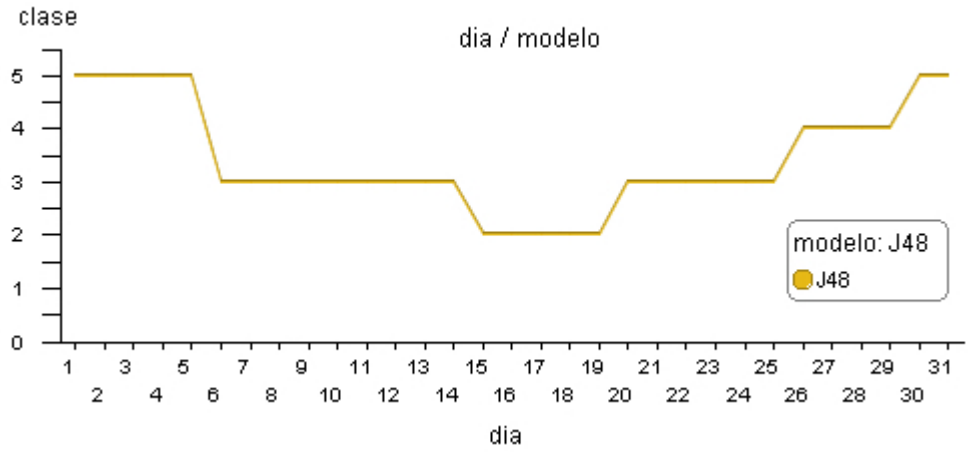

Figura 3.4: Resultados J48 por día.

El atributo zona no afecta mayormente el comportamiento global de la función. Los resultados obtenidos son similares a los obtenidos con el Modelo 1 usando ID3.

Modelos de redes neuronales

Para la construcción de redes neuronales para el caso de estudio se utiliza como base el artículo publicado por Kasstra y Boyd en 1995. En este artículo se describe un procedimiento a seguir que consta de 8 pasos para la construcción de una red neuronal para modelos de predicción [22].

El procedimiento tiene puntos en común con el método aplicado en los modelos anteriores, pero da un enfoque especial del mismo a las redes neuronales.

Procedimiento:

- Selección de variables: determinar cuales variables de las disponibles en los datos son importantes y cuales no.
- Recolección de datos: debe considerarse el costo y disponibilidad de los datos seleccionados que deberán ser recolectados, así como tener en cuenta el tiempo requerido para esta tarea.
- Preprocesamiento de datos: la representación de los datos es una etapa crítica en el diseño de la red pues la red neuronal realiza iden-

tificación de patrones. El procesamiento incluye transformar los datos de forma de minimizar el ruido, resaltar relaciones importantes, detectar tendencias y aplanar la distribución de la variables lo cual ayuda a que la red aprenda los patrones importantes.

- Entrenar, testear y validar los datos: como es común se dividen los datos disponibles en tres conjuntos llamados de entrenamiento, de testeo y de validación. Los datos de entrenamiento se usan para entrenar la red, los de testeo para evaluar el comportamiento de la red y los de validación para verificar el rendimiento de la red. Se deben cuidar las proporciones entre los conjuntos de datos e integración de los conjuntos. Generalmente los datos de testeo corresponden a 10 % a 30 % de los datos de entrenamiento y se elegirán las redes que se comporten mejor para los datos de testeo. Por su parte el tamaño de los datos de validación deben balancearse entre tener los suficientes para realizar la validación de las redes y dejar suficientes datos para las otras dos tareas. Los datos de validación deberían consistir de las observaciones continuas más recientes.

- Definir la red neuronal: número de capas ocultas, número de neuronas en las capas ocultas, número de neuronas de salida, funciones de transferencia.

  Definir el número de neuronas de entrada es generalmente el paso más sencillo, luego de definidas las variables independientes en los datos, cada una de ellas se convertirá en una neurona de entrada.

  En teoría una capa oculta con una cantidad suficiente de neuronas puede aproximar cualquier función continua. Incrementar el número de capas ocultas aumenta el tiempo de procesamiento y puede conducir a sobreentrenamiento A.6.7. Se recomienda por lo tanto comenzar con una o como máximo dos capas ocultas, si los resultados no son buenos se puede experimentar con más capas ocultas. El número de neuronas de las capas ocultas es importante, pero no existe una fórmula mágica para determinar el valor óptimo. Seleccionar el mejor número de neuronas requiere experimientación, para ello se pueden tener en cuenta ciertas reglas que pueden guiar esta tarea.

  - Regla de Masters: para una red de 3 capas (1 capa oculta), con n neuronas de entrada y m neuronas de salida, el número de neuronas de la capa oculta debería ser $\sqrt{n * m}$.

- Regla de Baily y Thompson: para una red de 3 capas (1 capa oculta), el número de neuronas de la capa oculta debe ser el 75 % del número de neuronas de entrada.
- Regla de Katz: el número óptimo de neuronas de la capa oculta está entre la mitad y tres veces el número de neuronas de entrada.
- Regla de Ersoy: duplicar el número de neuronas de entrada hasta que el rendimiento de la red en el entrenamiento se deteriore.

Decidir el número de neuronas de salida es más sencillo, en general se recomienda el uso de una única neurona de salida. Las redes con múltiples neuronas de salida producen resultados inferiores.

- Definir los criterios de evaluación. La función de error más comunmente usada es la suma de los errores al cuadrado y otras funciones para el cálculo del error, aunque las funciones de error pueden no ser el criterio definitivo de evaluación.

- Entrenar la red neuronal: determinar el número de iteraciones, tasa de entrenamiento y momentum. El objetivo de entrenar la red neuronal es encontrar el conjunto de pesos que generar el valor mínimo para la función de error. Hay dos corrientes en cuanto a la determinación del número de iteraciones. El primer método es parar las iteraciones cuando las mismas no producen mejoras en el error, el objetivo es encontrar un mínimo local para el error y el punto de parada se llama convergencia. El segundo método es parar las iteraciones luego de un número predeterminado de las mismas. La tasa de entrenamiento es una constante de proporcionalidad que determina el tamaño de los cambios de peso. El cambio en el peso es proporcional al impacto del peso de la neurona en el error. Una tasa usual varía entre 0.1 y 0.9, y a su vez la misma decrece durante el entrenamiento a medida que el resultado se acerca a la convergencia. El momentum determina como los cambios de pesos pasados afectan a los pesos actuales, previene de oscilaciones dramáticas filtrando variaciones bruscas. La tasa de entrenamiento y el momentum se eligen para mejorar acelerar el entrenamiento sin provocar oscilaciones.

- Implementación. Una gran ventaja de las redes neuronales es su gran adaptabilidad a cambios en los datos mediante reentrenamiento de la red. Sin el reentrenamiento el rendimiento de la red se degradará con el tiempo, con reentrenamiento no hay garantías de que se mantenga

el rendimiento pues las variables que componen la red pueden perder importancia en la realidad.

Modelo 3: Perceptrón multicapa o multiperceptrón Para el caso de estudio podrían construirse innumerables redes producto de organizar los datos de distintas formas en base a los pasos de *Selección de variables importantes* y *Preprocesamiento de datos* del proceso de creación de la red.

Ejemplos:

- Si se toma la planilla original de la Tabla 3.1 es posible dividirla en 2 planillas, una correspondiente a la zona 1 con las columnas *día* y *eventos zona 1* y otra correspondiente a la zona 2 con las columnas *día* y *eventos zona 2*. A partir de estos conjuntos de datos se puede construir una red en que la variable de entrada es el día y la de salida es la cantidad de eventos y analizarla en forma independiente para las zonas 1 y 2.

- A partir de la planilla de la Tabla 3.2 o de la Tabla 3.3 podemos construir redes en que la variables de entrada son el día y la zona, y la variable de salida es la cantidad de eventos, esta cantidad será un entero para el primer caso y un identificador de rango para el segundo.

- Se puede incluso observar los resultados obtenidos en el modelo de árboles de decisión para identificar rangos de días que se comporten de forma similar y transformar la variable de entrada día de un rango (1,..,31) a un rango de identificadores (D_1,...,D_n).

Luego de creada y entrenada la red neuronal se evaluan los casos del conjunto de datos de testeo de la siguiente forma. Para cada neurona de la capa de entrada se calcula el valor de las neuronas de la capa oculta y para cada neurona de la capa oculta se calcula el valor de las neuronas de la capa de salida. Sea la tupla $x = (x_1, x_2, ..., x_n)$ un registro de entrada y sea $a = (a_1, a_2, ..., a_n)$ los pesos de las aristas de la capa entrada a la capa oculta. Entonces se calcula el valor en las neuronas ocultas con la siguiente secuencia de cálculos, donde la función f3(x) es una función sigmoidal [1].

- $f1(x) = x * a = (x_1, x_2, ..., x_n) * (a_1, a_2, ..., a_n)$

[1] Ver definición de función sigmoidal en el Anexo F

- $f2(x) = f1(x) - desvioE = (f1_1 - desvioE, f1_2 - desvioE, ..., f1_n - desvioE)$
- $f3(x) = \frac{1}{1+\exp(f2(x))}$

Luego sea la tupla $y = (y_1, y_2, ..., y_n)$ el registro de entrada de valores en la capa oculta calculado en $f3(x)$ y sea $w = (w_1, w_2, ..., w_n)$ el vector de pesos de las aristas de la capa oculta. Entonces se calcula el valor en las neuronas de salida con la siguiente secuencia de cálculos, donde la función g3(x) es una función sigmoidal.

- $g1(x) = y * w = (y_1, y_2, ..., y_n) * (w_1, w_2, ..., w_n)$
- $g2(x) = g1(x) - desvioO = (g1_1 - desvioO, g1_2 - desvioO, ..., g1_n - desvioO)$
- $g3(x) = \frac{1}{1+\exp(g2(x))}$

De los resultados obtenidos para la función $g3(x)$ para las neuronas de la capa de salida se deduce que el mayor es valor corresponde a la salida más probable y por lo tanto el generado por la evaluación de los datos en la red.

Para probar el trabajo con redes neuronales se contruyen 3 redes neuronales multiperceptrón diferentes y analizaré como la variación en los datos de entrada afecta los resultados. Sean estas redes la red A, red B y red C. Se usa la red C para ejemplificar como se realiza la clasificación de nuevos casos luego de construida la red. Luego se determina cual de las 3 redes produce mejores resultados, en base al porcentaje de error obtenidos al predecir los valores de salida para los datos de testeo.

Red neuronal A:

- Selección de variables importantes: día, zona, eventos.
- Recolección de datos: los datos están disponibles.
- Preprocesamiento de datos: ninguno.
- Entrenar, testear y validar los datos: El conjunto de datos disponibles es de 244 registros, se divide el mismo en 50 % de datos de entrenamiento, 25 % de datos de testeo y 25 % de datos de validación.
- Definir la red neuronal:

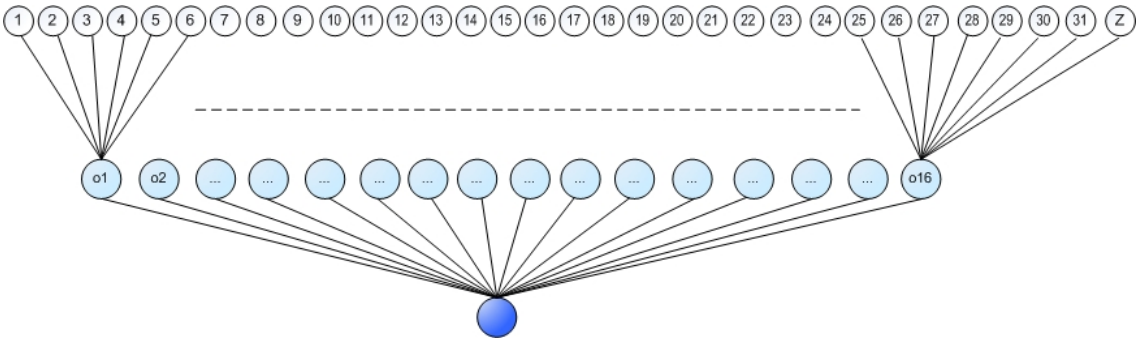

Figura 3.5: Red neuronal A.

- 32 neuronas de entrada: zona, día=1, día=2, ..., día=31.
- 1 capa oculta, 17 neuronas en la capa oculta.
- 1 neurona de salida, correspondiente a la cantidad de eventos.

- Definir criterio de evaluación: porcentaje de casos correctamente clasificados del conjunto de datos de testeo.

MultiLayerPerceptron de WEKA MultiLayerPerceptron es una red neuronal que se entrena usando propagación hacia atrás y minimiza el error al cuadrado del valor de salida por el método del gradiente descendente [33]

Resultados obtenidos por MultiLayerPerceptron de WEKA Cabezal del archivo de entrada ARFF:

@relation eventos

@attribute dia 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,
18,19,20,21,22,23,24,25,26,27,28,29,30,31
@attribute zona 1,2
@attribute eventos integer

Salida de ejecución de MultiLayerPerpectron de WEKA:
=== Run information ===

Scheme: weka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
Relation: eventos

```
Instances 122
Attributes: 3
        dia
        zona
        eventos
Test mode: user supplied test set: 122 instances

    === Classifier model (full training set) ===

     Linear Node 0
    Inputs Weights
    Threshold 0.4595318273604999
    Node 1 0.6955776619730751
    Node 2 -0.4773140255059659
    Node 3 -0.6144184251073048
    ....    Node 16 -0.5874742191078342
Sigmoid Node 1
    Inputs Weights
    Threshold -0.11490227380291533
    Attrib dia=1 0.9503722243849017
    ....
    Attrib dia=31 1.0453733219081003
    Attrib zona 0.5429111718957668
Sigmoid Node 2
    Inputs Weights
    Threshold -0.21037796381710533
    Attrib dia=1 -0.4023714607782501
    ....
    Attrib dia=31 -0.3588484206344755
    Attrib zona -0.25289202825361146
....
....
Sigmoid Node 16
    Inputs Weights
    Threshold -0.19577734579736822
    Attrib dia=1 -0.4210748817561034
    ....
    Attrib dia=31 -0.3764687212281562
    Attrib zona -1.3099269546645833
Class
    Input
    Node 0
```

Interpretación: El modelo de clasificación es una red neuronal. Las líneas superiores describen los nodos que componen la red y las aristas que los conectan. El nodo etiquetado con el número 0 de tipo “Linear” (Lineal) es el nodo de salida. Los nodos etiquetados entre el 1 y el 16 como “Sigmoid Node” (Nodo Sigmoidal) son nodos de la capa oculta y el nodos etiquetados

como “Attrib” son los nodos de entrada. Los pesos de las aristas entre el nodo de entrada y los 16 nodos de la capa oculta son especificados a continuación de la definición de cada nodo de la capa oculta. Los pesos de las aristas entre los nodos de la capa oculta y el nodo de salida se especifican a continuación del nodo de salida. Con estos datos podemos construir graficamente la red como se muestra en la Figura 3.5.

Time taken to build model: 4.28 seconds

=== Predictions on test set ===

inst#, actual, predicted, error
1 24388 30177.909 5789.909
2 27495 25419.341 -2075.659
3 30391 29903.447 -487.553
4 35909 35287.249 -621.751
5 22876 27620.891 4744.891
6 12255 12641.078 386.078
7 18142 3019.099 -15122.901
8 1060 9829.362 8769.362
9 1066 5200.419 4134.419
....
122 23235 28401.402 5166.402

=== Evaluation on test set ===
=== Summary ===

Correlation coefficient 0.9124
Mean absolute error 3797.1256
Root mean squared error 4751.3199
Relative absolute error 38.7881 %
Root relative squared error 41.4009 %
Total Number of Instances 122

Red neuronal B:

- Selección de variables importantes: día, zona, eventos.
- Recolección de datos: los datos están disponibles.
- Preprocesamiento de datos: ninguno.
- Entrenar, testear y validar los datos: El conjunto de datos disponibles es de 244 registros, se divide el mismo en 50 % de datos de entrenamiento, 25 % de datos de testeo y 25 % de datos de validación.
- Definir la red neuronal:

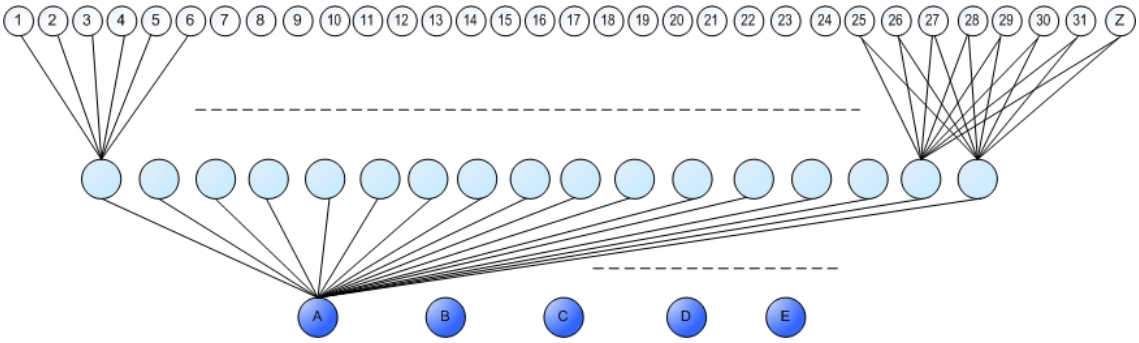

Figura 3.6: Red neuronal B.

- 32 neuronas de entrada: zona, día=1, día=2, ..., día=31.
- 1 capa oculta, 11 neuronas en la capa oculta, determinadas por la regla de Masters.
- 5 neuronas de salida, correspondientes a las 4 posibles valores para la cantidad de eventos.

- Definir criterio de evaluación: porcentaje de casos correctamente clasificados del conjunto de datos de testeo.

Para esta red las entradas día=1 a día=31 sólo se activará una a la vez, lo que es lo mismo que decir que una única arista toma el valor 1 y las demás son cero. En cuanto a la entrada zona se activará o desactivará en forma independiente a las demás entradas.

Red neuronal C:

- Selección de variables importantes: día, zona, eventos.

- Recolección de datos: los datos están disponibles.

- Preprocesamiento de datos: definir 6 rangos de días según sus valores

  - de 1 a 5 valor d1.
  - de 6 a 10 valor d2.
  - de 11 a 15 valor d3.
  - de 16 a 20 valor d4.
  - de 21 a 25 valor d5.

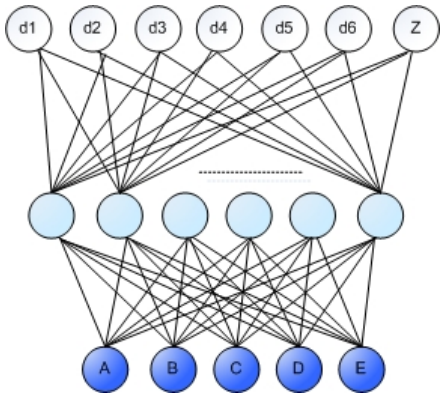

Figura 3.7: Red neuronal C.

- de 26 a 30 valor d6.

- Entrenar, testear y validar los datos: El conjunto de datos disponibles es de 244 registros, se divide mismo en 50 % de datos de entrenamiento, 25 % de datos de testeo y 25 % de datos de validación.

- Definir la red neuronal:

  - 7 neuronas de entrada: zona, d1, d2, ..., d6.
  - 1 capa oculta, 6 neuronas en la capa oculta.
  - 5 neuronas de salida, correspondientes a las 4 posibles valores para la cantidad de eventos.

- Definir criterio de evaluación: porcentaje de casos correctamente clasificados del conjunto de datos de testeo.

Para esta red las entradas día=1 a día=31 sólo se activará una a la vez, lo que es lo mismo que decir que una única arista toma el valor 1 y las demás son cero. En cuanto a la entrada zona se activará o desactivará en forma independiente a las demás entradas.

Resultados obtenidos por MultiLayerPerceptron de WEKA
Cabezal del archivo de entrada ARFF:

@relation eventos

@attribute dia d1,d2,d3,d4,d5,d6
@attribute zona 1,2
@attribute eventos A,B,C,D,E

Salida de ejecución de MultiLayerPerpectron de WEKA:
=== Run information ===

Scheme: weka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
Relation: eventos
Instances: 122
Attributes: 3
dia
zona
eventos
Test mode: user supplied test set: 122 instances

=== Classifier model (full training set) ===

Sigmoid Node 0
Inputs Weights
Threshold -1.7675499906943424
Node 5 -4.7167132738733555
Node 6 -1.3834443804739596
Node 7 1.9800943163654237
Node 8 0.4612246284637466
Node 9 -1.237186922105903
Node 10 0.8527035047350702
Sigmoid Node 1
Inputs Weights
Threshold -0.1583251445408486
Node 5 -1.8162971094171523
Node 6 -2.7905537897141803
Node 7 -0.011779389438079425
Node 8 1.7124881871931392
Node 9 0.23982011480393473
Node 10 -1.8636645133947467
Sigmoid Node 2
Inputs Weights
Threshold -0.7707751023804151
Node 5 -3.735006367602504
Node 6 -2.4374974088714514
Node 7 0.2525529857255525
Node 8 -1.628705021969945
Node 9 0.6762635960105778
Node 10 1.2073663050783165
Sigmoid Node 3
Inputs Weights

Threshold -2.2023325654166275
Node 5 2.9170414308057957
Node 6 1.5339921423917628
Node 7 -2.8862236407079314
Node 8 -0.800407281484188
Node 9 -3.129838337287769
Node 10 -0.7159342140610128
Sigmoid Node 4
Inputs Weights
Threshold -3.0057907236654446
Node 5 3.145540131074929
Node 6 1.637617804861985
Node 7 -4.714693109212551
Node 8 -1.47173002715266
Node 9 0.141615092106646
Node 10 -0.7246056465703541
Sigmoid Node 5
Inputs Weights
Threshold -0.7637915834694045
Attrib dia=d1 3.093949377178309
Attrib dia=d2 -1.4961983549303068
Attrib dia=d3 -0.09152694302103709
Attrib dia=d4 -0.1430190273563388
Attrib dia=d5 -0.6329221203467724
Attrib dia=d6 2.2167244080480613
Attrib zona 0.7122821859007462
Sigmoid Node 6
Inputs Weights
Threshold -0.5589650269005058
Attrib dia=d1 2.449779065256489
Attrib dia=d2 -0.8766990978387309
Attrib dia=d3 -0.3780846722261038
Attrib dia=d4 -0.2199471984661637
Attrib dia=d5 -0.4892346972621378
Attrib dia=d6 1.6697549146067914
Attrib zona -0.5195405819423442
Sigmoid Node 7
Inputs Weights
Threshold -1.0355775543566756
Attrib dia=d1 -0.8679643052071625
Attrib dia=d2 1.2486730344900128
Attrib dia=d3 1.7139932657300423
Attrib dia=d4 2.105926640049083
Attrib dia=d5 0.7344637745488116
Attrib dia=d6 -0.7277657108985164
Attrib zona -1.2750989448864034
Sigmoid Node 8
Inputs Weights
Threshold -1.3832644314076785

Attrib dia=d1 0.38187066271020037
Attrib dia=d2 -0.4039817154685291
Attrib dia=d3 1.2748349766905855
Attrib dia=d4 0.8919842623487451
Attrib dia=d5 3.2948644449941162
Attrib dia=d6 0.2012375449598364
Attrib zona 2.855571217201355
Sigmoid Node 9
Inputs Weights
Threshold -0.5974724953369454
Attrib dia=d1 3.783271319612113
Attrib dia=d2 -0.4401525253780918
Attrib dia=d3 0.16561271199670344
Attrib dia=d4 -1.4112400134700853
Attrib dia=d5 1.5111235864547456
Attrib dia=d6 -1.1468543360828416
Attrib zona -1.7120301382899121
Sigmoid Node 10
Inputs Weights
Threshold -0.3145632818696628
Attrib dia=d1 -1.0669598896054182
Attrib dia=d2 0.7735676415695568
Attrib dia=d3 0.5016497397042868
Attrib dia=d4 -2.9514668280544525
Attrib dia=d5 2.8436218498940082
Attrib dia=d6 1.059931371501168
Attrib zona 2.611998842104452
Class A
Input
Node 0
Class B
Input
Node 1
Class C
Input
Node 2
Class D
Input
Node 3
Class E
Input
Node 4

Time taken to build model: 1.14 seconds

=== Evaluation on test set ===
=== Summary ===
Correctly Classified Instances 56 45.9016 %
Incorrectly Classified Instances 66 54.0984 %

Kappa statistic 0.3406
Mean absolute error 0.2442
Root mean squared error 0.36
Relative absolute error 75.572 %
Root relative squared error 88.4268 %
Total Number of Instances 122

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0	0	0	0	0	A
0.864	0.31	0.38	0.864	0.528	B
0.636	0.15	0.483	0.636	0.549	C
0.56	0.093	0.609	0.56	0.583	D
0.5	0.106	0.45	0.5	0.474	E

=== Confusion Matrix ===

A	B	C	D	E	classified as
0	23	12	0	0	A
0	29	3	0	0	B
0	8	14	0	0	C
0	0	0	14	11	D
0	0	0	9	9	E

EJEMPLO de cálculo de predicciones para nuevos casos a partir de la red.

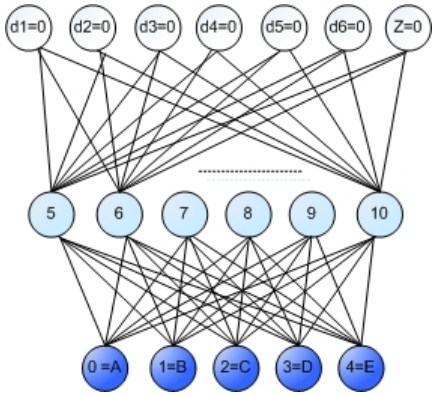

Figura 3.8: Ejemplo clasificación de un caso en la red neuronal C.

Datos de entrada: día=1, zona=1
Vector de entrada de la red es entonces $x = (1, 0, 0, 0, 0, 0, 0)$

Paso 1: Se calculan los valores de las neuronas de la capa oculta a partir de las neuronas de entrada $f1(x) = x * a = (1, 0, 0, 0, 0, 0, 0) * a = a1$

	Nodo 5	Nodo 6	Nodo 7	Nodo 8	Nodo 9	Nodo 10
desvío	-0,763792	2,449779	-0,867964	-1,383264	3,783271	-0,314563
$f1 = x * a = a1$	3,093949	-0,558965	-1,035578	0,381871	-0,597472	-1,066960
$f2 = f1 - desvio$	3,857741	-3,008744	-0,167613	1,765135	-4,380744	-0,752397
$f3 = \frac{1}{1+\exp(f2)}$	0,979321	0,047032	0,458195	0,853852	0,012361	0,320299

Paso 2: Se extraen de la red los pesos de las aristas que conectan la capa oculta con la capa de salida y se contruye con estos pesos la matriz M.

	Nodo 0	Nodo 1	Nodo 2	Nodo 3	Nodo 4
Nodo 5	-4,716713	-1,816297	-3,735006	2,917041	3,145540
Nodo 6	-1,383444	-2,790554	-2,437497	1,533992	1,637610
Nodo 7	1,980094	-0,011779	0,252553	-2,886224	-4,714693
Nodo 8	0,461225	1,712488	-1,628705	-0,800407	-1,471730
Nodo 9	-1,237187	0,239820	0,676264	-3,129838	0,141615
Nodo 10	0,852704	-1,863665	1,207366	-0,715934	-0,724606

Se extraen de la red los desvíos para cada uno de los nodos de la capa oculta y se construye con ellos el vector d.

	Nodo 0	Nodo 1	Nodo 2	Nodo 3	Nodo 4
desvio	-1,767550	-0,158325	-0,770775	-2,023326	-3,005790

$d = (-1{,}767550, -0{,}158325, -0{,}770775, -2{,}023326, -3{,}005790)$

Paso 3: Se calculan los valores de las neuronas de salida.
x = (0.979321 , 0.047032, 0.458195, 0.853852, 0.012361, 0.320299)
a = M(i), tal que i es la columa de M

	Nodo 0=A	Nodo 1=B	Nodo 2=C	Nodo 3=D	Nodo 4=E
$g1 = x * a$	-3,125330	-1,047137	-4,652287	0,654984	-0,489712
$g2 = f1 - desvio$	-1,357780	-0,888812	-3,881512	2,678310	2,516078
$g3 = \frac{1}{1+\exp(f2)}$	0,204601295	0,29135513	0,020203038	0,935734575	0,925261318

Por lo tanto, como el mayor valor se da para Nodo 3 el valor de salida que se predecirá para este registro es D, entre 20.001 y 30.000 eventos.

3.3.4. Validación de los modelos

Los modelos se validan clasificando un conjunto de datos de prueba y construyendo la matriz de confusión [2] para verificar su exactitud.

El error en los resultados obtenidos es muy alto, una de las razones de esto puede ser que el método no es el adecuado o que la calidad o cantidad de los datos no es suficiente. Esta primera evaluación del caso de estudio con varios métodos puede desembocar entonces en descartar alguno de ellos o mejorar los datos sobre los cuales se han aplicado los mismos.

En cuanto a la calidad de los datos, la fuente de datos disponible sólo contiene información de día y zona. El día del mes (1..31) es el atributo tipo fecha y es más importante que la zona como se comprueba en el estudio de los árboles de decisión. Sería deseable contar con más información de la fecha del evento como ser día de la semana y mes. Como todas las actividades el comportamiento de los fines de semana puede variar respecto a los del resto de la semana. El mes por sí mismo puede influir en los resultados y además se puede deducir de él otro dato como es la estación del año que podría ser de utilidad.

Otro aspecto es la cantidad de los datos, 244 registros quizás es suficiente para la construcción del árbol de decisión pero no parece ser suficiente para el entrenamiento de la red. Otra opción para mejorar el error es recolectar más datos de la misma fuente y probar los modelos construídos con mayores volumenes.

3.3.5. Puesta en producción los modelos

Tanto los modelos de árboles de decisión como las redes se ponen a disposición de los usuarios para que los mismos puedan usarlos para predecir

[2] Definición de Matriz de Confusión en el Anexo F

casos a futuro. En el caso de los árboles de decisión su utilización es trivial y los resultados se obtienen en forma visual. En el caso de las redes neuronales es necesario efectuar cálculos para cada registro a evaluar. Por lo tanto, es de mayor utilidad que el usuario disponga de un programa que ejecute los cálculos implícitos en la red que disponer la red misma.

3.3.6. Administración de los meta datos

Se guardan todos los archivos utilizados: fuentes de datos en planillas excell, archivos ARFF (archivos de texto) construidos para la ejecución en Weka, planillas excell con resultados y archivos de texto con ejecuciones de cada uno de los modelos. Se administran todos los archivos desde su versión inicial a la definitiva y todos los modelos con sus respectivos parámetros de entrada y salida.

3.4. Evaluación de los resultados obtenidos

Lo primero que hay que destacar de estos resultados es que no podemos confiar completamente en su fiabilidad, ya que el conjunto de datos que se ha empleado tanto para su entrenamiento como para su posterior evaluación es muy pequeño.

A partir de la matriz de confusión de la salida ejecución de cada algoritmo podemos obtener una medida de performance de cada algoritmo que permita compararlos entre sí. El porcentaje de éxito de cada algoritmo lo mediremos como las clasificaciones correctas (suma de los valores de la diagonal de la matriz de confusión) entre la cantidad total de instancias clasificadas.

El algoritmo ID3 clasifica correctamente $35 + 25 + 28 + 34 + 25 = 147$ instancias sobre un total de 244, lo que corresponde a un porcentaje de éxito de 60.2 %

El algoritmo C4.5 obtiene los mismo resultados al ser ejecutado con el atributo día en formato discreto. Cuando el mismo se ejecuta con el atributo día con datos continuos clasifica correctamente $22 + 21 + 36 + 26 + 29 = 134$ instancias sobre un total de 244, lo que corresponde a un porcentaje de éxito de 54.9 %

Los algoritmos de árboles de decisión tienen prestaciones aceptables, destacando el algoritmo ID3. No obstante, no es lógico que de unos resultados mejores que el C4.5, ya que este último no es más que una extensión del

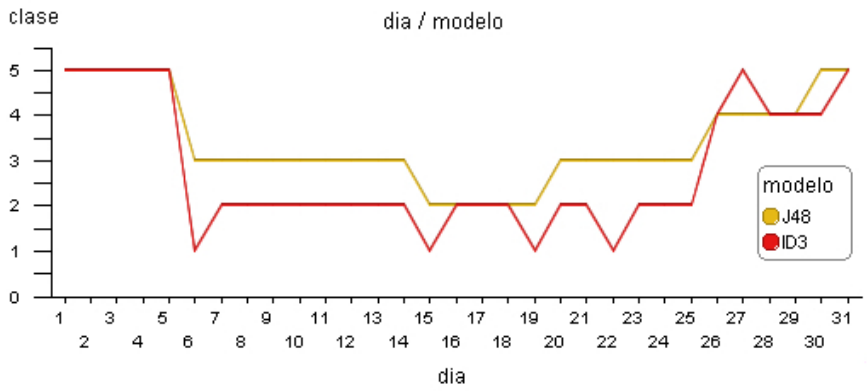

Figura 3.9: Comparación de resultados de ID3 y J48 por día.

primero para poder trabajar con datos continuos, incompletos o ruidosos. Esta diferencia se debe a la no discretización del atributo día, por lo tanto al no plantear restricciones en los datos, es preferible este modelo antes que el de ID3.

El algoritmo perceptrón multicapa para la red C clasifica correctamente $0 + 29 + 14 + 14 + 9 = 66$ sobre un total de 244, lo que corresponde un porcentaje de éxito de 27.0 %

El porcentaje de éxito de el perceptrón multicapa es muy inferior al de los árboles de decisión, además de que estos son preferibles por su simplicidad y facilidad de interpretación. Por lo tanto para el caso de estudio preferimos cualquiera de los modelos de árboles de decisión construídos, en especial el Modelo 2 de C4.5 que genera un árbol más compacto.

3.5. Análisis del caso de estudio desde el punto de vista de Investigación Operativa

3.5.1. Escenario 1

Descripción del problema a resolver A partir de la fuente de datos proporcionada se desea pronosticar la cantidad de eventos que ocurrirán en el futuro.

A que tipo de problemas de IO representa? Es lo que se conoce

como un problema de pronóstico. Existen varias áreas donde los pronósticos juegan un papel importante, como la comercialización, la planificación financiera y la planeación de producción. Las decisiones gerenciales rara vez se toman sin contar con alguna forma de pronóstico [19].

Cuales son las técnicas de IO aplicables? Para obtener pronósticos

se pueden emplear técnicas cualitavivas o cuantitativas, que en general se usan en forma combinada. Para las técnicas cualitativas el pronóstico es casi siempre el resultado de la expresión de opiniones personales de los expertos, se conoce también como *técnica subjetiva*. Dentro de las técnicas cuantitativas, existen dos opciones, ambas basadas en estadísticas: el análisis de series de tiempo y el análisis de regresión. Una serie de tiempo estadística es una serie de valores numéricos que toma una variable aleatoria a lo largo de un período de tiempo. Se analizan las series de datos históricos para pronosticar los valores de la variable de interés en el futuro. En el modelo de regresión, la variable que se va a pronosticar (variable dependiente) se expresa como una función matemática de otras variables (variables independientes). [19]

En el caso de estudio que se está tratando, en series de tiempo la variable de interés es la cantidad de eventos y la serie de tiempo está definida por la secuencia de días. En análisis de regresión la variable dependiente es la cantidad de eventos y las variables independientes son el día y la zona.

A que escenarios de la vida real pueden corresponder los datos del caso de estudio?

Si los datos pertenecen a una cadena de venta de electrodomésticos y la cantidad de eventos es la cantidad de ventas realizadas en ese día en los locales comerciales de una zona. Al gerente de ventas le interesará preveer que días se realizaran más ventas para calcular de forma acorde la cantidad de vendedores necesarios para cada día del mes en cada zona. Al gerente de marketing le interesa saber cuales son los días de menor venta para colocar en esos días promociones interesantes para los clientes. Al gerente financiero les interesa realizar una prospección de las ventas en el próximo mes para evaluar el monto de inversión en mercadería que la empresa debe realizar y el retorno de la misma. Al gerente general le interesa realizar una prospección de las ventas a futuro por zona y determinar la tendencia de las compras a los largo de un período para evaluar el cierre o apertura de locales comerciales en cada zona.

3.5.2. Escenario 2

Descripción del problema a resolver A partir de la fuente de datos proporcionada se desea simular el comportamiento futuro de las medidas eventos de zona 1 y zona 2.

A que tipo de problemas de IO representa? Es un problema de simulación a eventos discretos.

Cuales son las técnicas de IO aplicables? El problema se puede modelar con un sistema de colas [19].

A que escenarios de la vida real pueden corresponder los datos del caso de estudio? Si los datos pertenecen a una estación de bomberos de una ciudad que atiende 2 zonas y los eventos representan llamados de asistencia por efectuados en un día y zona de la ciudad. Se prevee en los próximos 5 años un crecimiento poblacional en ambas zonas que elevará en un 10 % la cantidad de llamadas de asistencias. La simulación del comportamiento de llamados de asistencia - atención de llamado pueden ser útiles para considerar si es necesaria la instalación de una estación de bomberos en cada zona o la estación actual podrá todavía atender todos los llamados.

3.5.3. Escenario 3

Descripción del problema a resolver A partir de la fuente de datos proporcionada se desea encontrar interrelaciones entre los atributos.

A que tipo de problemas de IO representa? Representa un problema de análisis de decisiones que muestre las relaciones entre los atributos de los datos que permitan al dueño de los mismos tomar decisiones a partir de ellos.

Cuales son las técnicas de IO aplicables? Métodos de toma de decisiones con y sin experimentación [19].

A que escenarios de la vida real pueden corresponder los datos del caso de estudio? Si los datos pertenecen a una institución financiera y los eventos representan retiros de dinero realizados en cajeros automáticos de la institución en un día y zona de la ciudad. La descripción de interrelaciones en los datos se puede utilizar para planificar las necesidades de reposición de dinero en un plan mensual.

Capítulo 4

Desarrollo de un algoritmo

4.1. Motivación

El objetivo de esta etapa del proyecto es desarrollar un algoritmo de Minería de Datos y observar los resultados de su aplicación. Este algoritmo se aplicará al caso de estudio del capítulo anterior y se busca mejorar los resultados obtenidos por la aplicación de algoritmos existentes *ID3*, *C4.2* y *MultiLayerPerceptron*.

4.2. Decisiones de desarrollo

Se explica a continuación como se llegó a tomar la decisión de extender el paquete WEKA y cuales otras opciones se consideraron para ello.

Una vez tomada esta decisión y dados los pobres resultados obtenidos con el algoritmo MultiLayer Perceptron de redes neuronales de WEKA se decidió implementar una nueva versión del mismo, incorporando mejoras en el cálculo de los pesos de las aristas de la red.

4.2.1. Porqué extender un paquete existente en vez de desarrollar código propio?

Existen innumerables desarrollos de paquetes de software de Minería de Datos en el ámbito de software de código abierto. En todos ellos se implementan tareas básicas como la lectura de diferentes formatos de archivos

de datos, la generación de archivos de salida con los resultados y en general se cuenta con una batería típica de algoritmos de Minería de Datos que incluye los algoritmos más conocidos o los más simples. Algunos de estos paquetes cuentan incluso con interfase gráfica, por lo general modesta. Uno de los objetivos de la mayoría de los paquetes de software de código abierto es que los interesados extiendan estos paquetes con sus propios algoritmos y aprovechen el desarrollo existente al mismo tiempo que le agregan nuevas funcionalidades.

La opción de extender un paquete existente evita repetir tareas que son comunes a todos ellos; como la lectura de los datos de entrada desde archivos, la presentación de resultados, la interfase gráfica, etc. Esta elección permite concentrarse en el desarrollo del algoritmo de Minería de Datos, que es el objetivo fundamental del desarrollo.

4.2.2. Paquetes de código abierto considerados

Al estudiar los paquetes de código abierto disponibles se evaluaron los siguientes: Gnome Data Mine, Orange, R, TANAGRA, WEKA, YALE.

- **Gnome Data Mine** Gnome Data Mine es una colección de herramientas, empaquetadas para proporcionar una única colección de herramientas gratuitas de Minería de Datos [2]. Está disponible en un paquete *gnome-datamine-tools.tar.gz*. Para la instalación del paquete requiere que Python y Gnome esten instalados en el sistema y se recomienda el uso de la distribución Linux Debian. Cuenta con las siguientes aplicaciones:

  - gdmapriori, realiza Minería de Datos por reglas de asociación Apriori.
  - gdmbayes, realiza Minería de Datos por para clasificación de Bayes.
  - gdmtree, realiza Minería de Datos mediante árboles de decisión.
  - barchart, genera gráficas a partir de conjuntos de datos en formatos de salida a elección tipo PDF, PNG, FIG o EPS.
  - bin chart, genera gráficas de la distribución de un conjunto de datos en formatos de salida a elección tipo PDF, PNG, FIG o EPS.
  - gdmplot, genera planos en varios formatos incluyendo LaTeX, PostScript y PDF.

- **Orange** es un paquete de software de Minería de Datos basado en componentes que se distribuye bajo licencia GNU GPL. Incluye técnicas de pre-procesamiento, modelado y exploración de datos. Se basa en componentes C++, que son accedidos directamente, a través de scripts Python (se recomienda este último método), o a través de los objetos gráficos llamados Orange Widgets. Los componentes existentes pueden ser usados para crear componentes propios [6].

- **R** es un lenguaje y un ambiente para la computación estadística. R proporciona herramientas para la manipulación, cálculo y visualización de datos. Proporciona una gran cantidad de técnicas estadísticas y gráficas. Los códigos fuente de R están disponible bajo los términos de GNU GPL. Compila y corre en una gran variadad de plataformas UNIX, Windows and MacOS. R se puede extender fácilmente mediante el agregado de paquetes. Para tareas computacionales intensivas se puede llamar a código C, C++ y Fortran. Para usuarios más avanzados provee la posibilidad de manipular los objetos R directamente mediante código C. R cuenta tanto con manuales como con documentación en línea [7].

- **TANAGRA** es un software libre de Minería de Datos con propósitos académicos y de investigación. Propone variados métodos de Minería de Datos: exploración de datos, aprendizaje estadístico y aprendizaje de máquina. TANAGRA es un proyecto de código abierto que permite al usuario agregar sus propios algoritmos. Su primer propósito es proporcionar a estudiantes e investigadores un software de Minería de Datos fácil de utilizar. El segundo propósito es proponer una arquitectura sencilla de extender con métodos propios y comparar su desempeño. El tercer propósito es difundir de forma pedagógica una metodología para construir este tipo de software, aprovechandose del estudio del código proporcionado.

- **WEKA** [1] es un software de código abierto desarrollado bajo licencia GPL[2] por la Universidad de Waikato [46] (Nueva Zelanda). Es una colección de algoritmos de aprendizaje de máquina para Minería de Datos. Los algoritmos pueden ser aplicados directamente desde las interfases gráficas de WEKA o pueden ser llamados desde código Java propio. Entre ellos se encuentran herramientas para pre-procesamiento de datos, clasificación, regresión, clustering, reglas de asociación y visualización. Además de los algoritmos que implementa; permite desarrollar nuevos esquemas de algoritmos

[1] WEKA:The Waikato Environment for Knowledge Analysis

[2] GPL: General Public License. Significa que el software es de libre distribución y difusión.

e incorporarlos al paquete fácilmente [45]. El mayor defecto de WEKA es la escasa documentación orientada al usuario con la que cuenta. La ayuda en línea es escueta e incompleta y conduce a una usabilidad bastante pobre, lo que la hace una herramienta difícil de comprender y manejar sin información adicional [8].

- **YALE** [3] es un ambiente interactivo para la realización de experimentos de aprendizaje de máquina y Minería de Datos. Desde 2001 YALE ha sido desarrollado por la Unidad de Inteligencia Artificial de la Universidad de Dortmund. Está enteramente escrito en Java, por lo tanto corre en cualquier plataforma o sistema operativo para el cual exista una máquina virtual Java [4]. YALE puede ser usado por interface gráfica o por línea de comandos. La API Java permite usar operadores o experimentos desde aplicaciones Java propias. YALE provee además una forma de extensión simple mediante el uso del mecanismo de plugins [5] que permite integrar operadores nuevos y adaptar YALE a necesidades personales de cada usuario. El concepto de operador permite diseñas operadores complejos anidando y concatenando operadores más simples. Los experimentos pueden construirse como un conjunto arbitrario anidado de operadores y su configuración se describe mediante archivos XML que pueden ser creados fácilmente por la interface gráfica. YALE cuenta con más de 100 esquemas de aprendizaje que incluyen regresión, clasificación y clustering; así como también con métodos de pre-procesamiento de datos. Además de sus propios métodos permite el uso de todos los clasificadores existentes en WEKA. Cuenta con documentación de la API Java, tutorial y ayuda en línea básica [9].

4.2.3. Porqué extender WEKA y no otro software de código libre?

De todos estos paquetes se prefirió WEKA por las razones que se señalan a continuación:

- Al estar programado en Java, WEKA es independiente de la arquitectura, funciona en cualquier plataforma sobre la que este disponible una máquina virtual Java.

[3] YALE: Yet Another Learning Environment

[4] JVM™ (Java Vitual Machine), marca registrada de Sun Microsystems http: www.sun.com

[5] Un plugin (o plug-in) es un programa de ordenador que interactúa con otro programa para aportarle una funciónn o utilidad específica. Este programa adicional es ejecutado por el programa principal.

- Cuenta con gran variedad de algoritmos.

- Una de las propiedades más interesantes de WEKA, es su facilidad para añadir extensiones al mismo y también la posibilidad de modificar métodos existentes [8]. Su capacidad de extensión lo ha convertido en uno de los paquetes de código abierto más utilizados en el área en los últimos años. Su popularidad ha fomentado la depuración de errores y consiguiente evolución que se demuestra por la generación periódica de nuevas versiones.

- Existe abundante documentación de WEKA en la Web, en sitios oficiales, sitios no oficiales y foros de discusión de desarrolladores.

- Existe un libro llamado *Data mining: Practical Machine Learning Tools and Techniques* escrito por dos desarolladores experimentados de WEKA: Ian Witten y Eibe Frank [**?**]. Este libro consta de dos partes, la primera proporciona abundante información teórica de Minería de Datos, sus técnicas y algoritmos. La segunda parte describe el paquete WEKA, sus interfases gráficas, archivos de entrada, algoritmos de los cuales dispone y presenta una guía para el desarrollo de nuevos algoritmos y su incorporación a WEKA.

- Su interfase gráfica es modesta pero atractiva.

- La conversión de archivos de texto y formato excell al formato arff de WEKA es un procedimiento simple.

TANAGRA se descartó porque de la lectura de los códigos fuentes y documentación no se pudo extraer una idea clara del procedimiento de extensión y porque su interfase con el usuario es la más pobre de todos los paquetes evaluados. Gnome Data Mine, Orange y R se descartaron porque ante la elección de un lenguaje de programación se prefirió un paquete diseñado en Java, que se pudiera compilar y ejecutar en cualquier plataforma. YALE se descartó por la existencia de menos documentación disponible que para WEKA, sin ser por esto, posee todas las propiedades por las cuales se seleccionó WEKA y perfectamente pudo haberse extendido este paquete.

4.3. Procedimiento para la extensión de WEKA

Paso 1

El primer paso es bajar de la administración central de WEKA la versión que actualmente se encuentra en desarrollo. Esto se realiza vía CVS [6].
Configurar la variable de ambiente que indica el servidor de CVS con el

[6] CVS: Concurrent Versions System. http://www.nongnu.org/cvs/

siguiente valor
: *pserver* : *cvs_anon@cvs.scms.waikato.ac.nz* : */usr/local/globalcvs/ml_cvs*

Luego bajar la última versión en desarrollo siguiendo estos pasos :
cvs login
cvs co weka
cvs logout

Cuando solicita la contraseña al ejecutar *cvs login*, digitar ENTER.

Paso 2

Crear un proyecto Java con los fuentes bajados de la administración de WEKA. Para el desarrollo se eligió usar la Plataforma Eclipse [1] [7]. Instalar la máquina virtual sugerida para compilar el proyecto, en este caso es la JVM versión $1{,}2{,}0_0 6$.

Paso 3

Estudiando el código fuente y tomando como guía el libro *Data mining: Practical Machine Learning Tools and Techniques* [**?**], determinar cuales clases deben modificarse para integrar un nuevo algoritmo a WEKA y cual es el procedimiento a seguir para agregar nuevas clases.

Paso 4

Decidir que algoritmo agregar y proceder con las modificaciones de código. Se toma como punto de partida el algoritmo de propagación hacia atrás para el perceptrón multicapa desarrollado en WEKA (*MultiLayerPerceptron*) al cual se realizarán mejoras. Las nuevas versiones del algoritmo se probarán con los mismos datos utilizados en el caso de estudio y se evaluarán los resultados obtenidos.

La fase de entrenamiento de una red neuronal con el algoritmos de propagación hacia atrás es un problema de optimización sin restricciones. La meta del entrenamiento es obtener el conjunto de pesos que minimice los errores en los valores de salida de la red. El algoritmo de propagación hacia atrás de WEKA utiliza el método del gradiente descendente para calcular los errores, este el método de cálculo de errores más popularmente usado en este algoritmo. La primera modificación a realizar es cambiar el método del

[7] La Fundación Eclipse es una coorporación sin fines de lucro formada para fomentar el uso y evolución de la Plataforma Eclipse y cultivar los fundamentos de la comunidad de código abierto. Su sitio oficial http://www.eclipse.org.

gradiente descendente por el método de gradientes conjugados. La segunda modificación que realizar es cambiar la inicialización randómica de los pesos por inicialización de pesos por regresión.

4.4. Fundamentos teóricos y su aplicación al desarrollo

La fase de entrenamiento de una red neuronal es un problema de optimización no lineal. La meta del entrenamiento es buscar un peso óptimo para cada arista de la red de forma de minimizar los errores en la salida [14]. El algoritmo de gradiente conjugado y otros algoritmos relacionados son considerados generalmente los algoritmos de minimización multiptopósito más potentes. La derivada de segundo orden se utiliza durante búsquedas lineales para que no sea necesario el uso de la matriz Hessiana; sólo se necesita el producto de la matriz Hessiana y un vector [20].

El algoritmo *MultiLayer Perceptron* de WEKA calcula el error mediante el algoritmo de gradiente descendiente. Existen otros métodos de optimización más sofisticados que, en general, proporcionan mejores resultados que el descenso por el gradiente. Muchos algoritmos de gradiente conjugado han sido propuestos como algoritmos de aprendizaje para redes neuronales. Algunos de ellos son el algoritmo de Johansoon et al. de 1990, el de Battiti de 1992 y el algoritmo de gradiente conjugado escalado de Moller de 1993 [20], o el algoritmo de los gradientes conjugados de Shepherd de 1997. Los algoritmos de aprendizaje de gradiente conjugado son un tipo especial de algoritmos de propagación hacia atrás donde se utiliza información de las derivadas parciales de segundo orden.

A continuación se fundamenta el cambio a realizar en clasificador *MultiLayer Perceptron* del método de gradiente descendiente por el método de gradientes conjugados. Se planea desarrollar una nueva versión del clasificador MultiLayer Perceptron llamado *New MultiLayer Perceptron*.

4.4.1. Aplicación del método de gradiente conjugado

El algoritmo de propagación hacia atrás realiza los cambios en la dirección del negativo del gradiente, dirección en la que se presenta el mayor cambio de pendiente y donde la función decrece más rápido. Pero que decrezca más rápido no garantiza que por ello llegue antes al punto mínimo.

Esto se debe a que la superficie de error puede tener pendientes con grandes descensos que luego se estanquen en un valor cercano a un mínimo local. [32]

El gradiente conjugado propone que la dirección de cambio esté definida por el resultado de una búsqueda en distintas direcciones conjugadas del gradiente que indiquen al algoritmo por cual dirección la convergencia será más rápida. Esta búsqueda hace que se deje de lado el coeficiente *velocidad de aprendizaje* ya que el ajuste se modifica en cada iteración. El método del gradiente conjugado no es más que una variante o extensión del método del gradiente descendente utilizando direcciones conjugadas. Por lo tanto es necesario tener las mismas precauciones de cálculo que para el método de gradiente descendente para que no resulte divergente. Es un método directo que se usa como si se tratase de un método iterativo [32].

Los métodos de gradiente conjugado son populares por las siguientes razones [32] :

- Buscan direcciones descendientes que minimicen la perturbación de los resultados obtenidos en iteraciones previas.
- No usan la matriz Hessiana [8] directamente, por lo tanto no requiere su almacenamiento, utiliza sólo algunos vectores.
- Tiene orden O(N)

Este algoritmo no involucra el cálculo de las derivadas segundas e las variables y converge al mínimo de la función cuadrática en un número finito de iteraciones. El algoritmo del gradiente conjugado, sin aplicarlo aún al algoritmo de propagación inversa consiste en:

(1) Seleccionar la dirección de p0 , la condición inicial, en el sentido negativo del gradiente:
$p_0 = -g_0$
Donde
$g_k = \nabla e(x)|x = x_k$

(2) Seleccionar el porcentaje de aprendizaje para minimizar la función a lo largo de la dirección
$x_{k+1} = x_k + \alpha_k * p_k$

(3) Seleccionar la dirección siguiente de acuerdo a la ecuación

[8] Definición de Matriz Hessiana en Anexo F

$p_k = -g_k + \beta_k * p_{k-1}$
con
$\beta_k = \frac{\Delta g_{k-1}^T * g_k}{\Delta g_{k-1}^T * p_{k-1}}$
o
$\beta_k = \frac{\Delta g_k^T * g_k}{\Delta g_{k-1}^T * g_{k-1}}$

(4) Si el algoritmo en este punto aún no ha convergido, se regresa al punto (2)

Con el método de gradiente conjugado se puede modificar métodos de gradiente descendiente, para tomar ventaja de direcciones conjuntas de descenso [32].

4.5. Aplicación del algoritmo al caso de estudio

En el caso de estudio descripto en el capítulo anterior, se utilizan 3 redes neuronales que se llamaron red A, red B y red C. Se utilizaran la red A y C para comparar los algoritmos *MultiLayerPerceptron* y *NewMultiLayerPerceptron*. Cada una de estas redes se corresponde a un archivo de entrada ARFF con diferente formato, el cual ha sido específicado oportunamente. A continuación se procede a probar el nuevo algoritmo con cada uno de estos archivos de entrada y se analizan los resultados obtenidos.

Red A

Algoritmo	MultiLayerPerceptron	NewMultiLayerPerceptron
Relative absolute error	38.7881 %	41.1331 %
Root relative squared error	41.4009 %	43.3078 %

Red C

Algoritmo	MultiLayerPerceptron	NewMultiLayerPerceptron
Relative absolute error	57.815 %	67.5237 %
Root relative squared error	76.0386 %	79.3725 %

A continuación se muestra la aplicación del algoritmo *New MultiLayer Perceptron* desde la interfase de WEKA 4.1 y la salida de ejecución del algoritmo cuando el mismo es aplicado al archivo de datos correspondiente a la red C.

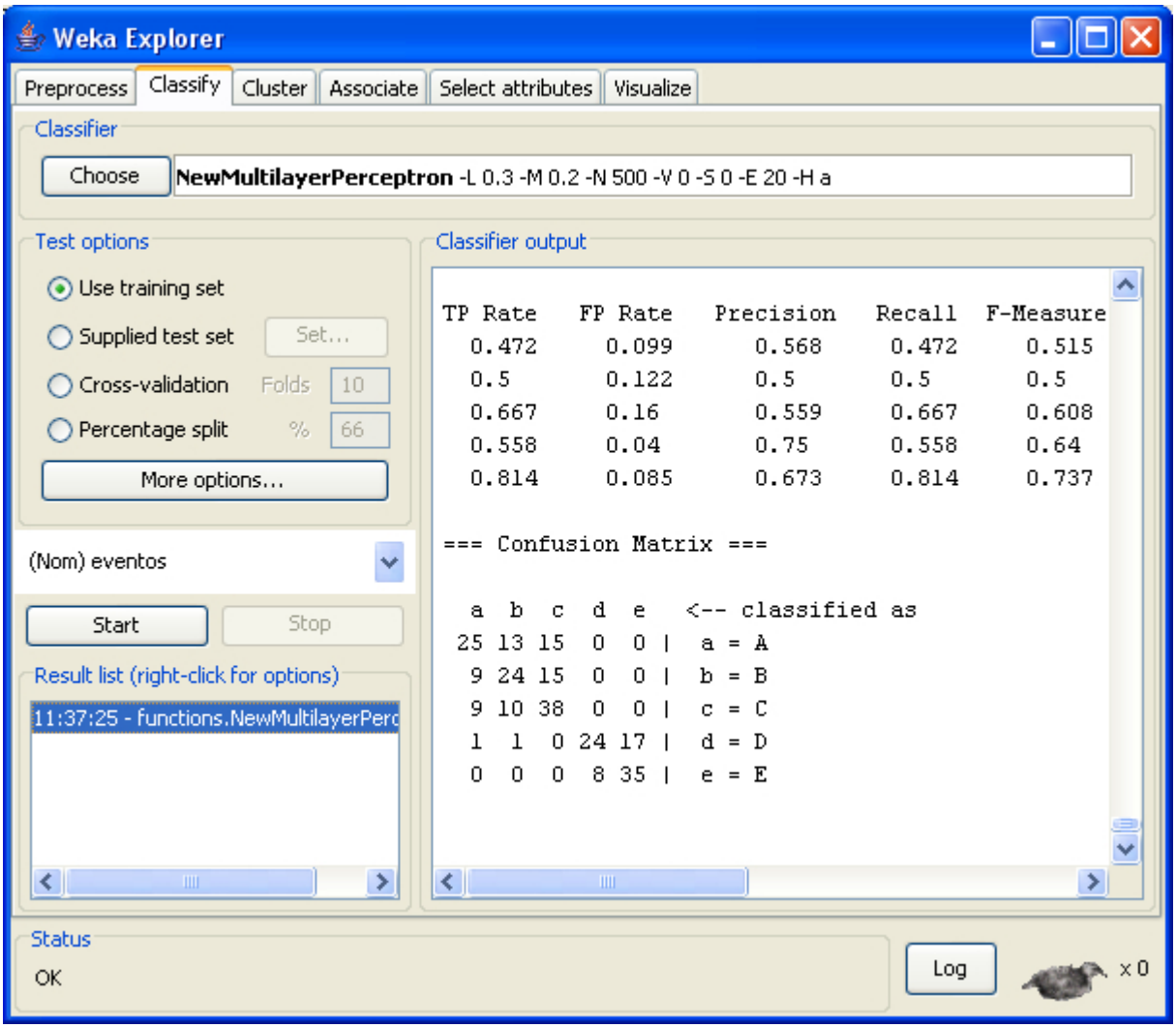

Figura 4.1: Aplicación de NewMultiLayerPerceptron.

Resultados de este caso obtenidos por NewMultiLayer Perceptron
Cabezal del archivo de entrada ARFF:

@relation eventos

@attribute dia 1,2,3,4,2,6,7,5,9,10,11,12,13,14,12,16,17,
15,19,20,21,22,23,24,22,26,27,25,29,30,31
@attribute zona 1,2
@attribute eventos integer

Salida de ejecución de NewMultiLayerPerpectron:
=== Run information ===

Scheme: weka.classifiers.functions.NewMultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
Relation: eventos
Instances: 244
Attributes: 3
dia
zona
eventos
Test mode: evaluate on training data

=== Classifier model (full training set) ===

Node 0
Inputs Weights
Threshold -1.043251453455095
Node 5 -2.596731970505046
Node 6 0.41395638567679305
Node 7 -2.9536125390135566
Node 8 1.2924599957653098
Node 9 -0.7123648019720636
Node 10 -0.14216347403317203
Node 11 -2.2584478852367695
Node 12 -0.7049462810401663
Node 13 0.2644600263255001
Node 14 -0.23362157669567046
Node 15 -0.47595315375671066
Node 16 -0.04536691316521883
Node 17 -0.479593819114802
Node 18 -0.32725592471092435
Node 19 -0.4816594346197144
Node 20 0.9171621649802432
Node 21 0.46744691422608486
Node 22 -0.16524948739098835
Node 1
Inputs Weights

Threshold -0.9784872588343849
Node 5 -0.031308700582059416
.....
Node 22 -0.971709738410239
Node 5
Inputs Weights
Threshold -0.11406092185083039
Attrib dia=1 -0.4065072028556096
Attrib dia=2 0.6164053534609366
Attrib dia=3 0.009022085583198818
Attrib dia=4 -0.3487471672027342
.....
Attrib dia=31 -0.323570279564459
Attrib zona -1.8452751510380057
Node 22
Inputs Weights
Threshold -0.10107113614053842
Attrib dia=1 0.43694485875045136
Attrib dia=2 0.44226667220318405
Attrib dia=3 -1.771711535207777
Attrib dia=4 0.40463897210668365
...
Attrib dia=31 1.2122316305598995
Attrib zona 2.056441190707233
Class A
Input
Node 0
Class B
Input
Node 1
Class C
Input
Node 2
Class D
Input
Node 3
Class E
Input
Node 4

Time taken to build model: 11.94 seconds

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

Correctly Classified Instances 146 59.8361 %
Incorrectly Classified Instances 98 40.1639 %
Kappa statistic 0.4957
Mean absolute error 0.2154

Root mean squared error 0.317
Relative absolute error 67.5237 %
Root relative squared error 79.3725 %
Total Number of Instances 244

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.472	0.099	0.568	0.472	0.515	0.863 A
0.5	0.122	0.5	0.5	0.5	0.855 B
0.667	0.16	0.559	0.667	0.608	0.861 C
0.558	0.04	0.75	0.558	0.64	0.932 D
0.814	0.085	0.673	0.814	0.737	0.955 E

=== Confusion Matrix ===

A	B	C	D	E	classified as
25	13	15	0	0	A
9	24	15	0	0	B
9	10	38	0	0	C
1	1	0	24	17	D
0	0	0	8	35	E

Como se ve los porcentajes de error *Relative absolute error* y *Root mean squared error* son mayores para *NewMultiLayer Perceptron* que los obtenidos con *MultiLayer Perceptron* aplicado a los mismos datos de entrada. Este no es el resultado que se esperaba.

4.6. Conclusiones

No se pudo verificar mediante la ejecución del nuevo algoritmo al caso de estudio que el mismo mejore los resultados respecto al algoritmo original *MultiLayer Perceptron*. Esto puede deberse al tamaño reducido de la muestra de datos o a características particulares de la función que se aproxima. No estamos en condiciones de llegar a conclusiones firmes únicamente luego de esta prueba, por lo tanto, queda pendiente para futuros trabajos la evaluación del algoritmo para otros casos y su corrección si fuese necesario.

Capítulo 5

Conclusiones y Trabajos futuros

5.1. Conclusiones

La búsqueda de información que impulsa la Minería de Datos nos afecta o afectará a todos en un futuro cercano. Datos que hemos generado serán usados en la búsqueda de información, buscaremos información en datos de nuestro interés o trabajaremos en proyectos de Minería de Datos. La Minería de Datos es uno de los campos más prometedores y de más rápido crecimiento en la actualidad en la industria de la computación. Este fenómeno no se da porque la Minería de Datos proponga tecnologías novedosas o revolucionarias, sino porque reune en forma inteligente técnicas, métodos, tecnología y algoritmos ya probados en otras áreas y permite usarlos para generar información a partir de datos. Las técnicas de Minería de Datos se han desarrollado especialmente para manejar grandes volumenes de datos, pero la noción de "grande" cambia cada vez que un avance tecnológico permite almacenar más datos. Por lo tanto el desarrollo de Minería de Datos como concepto promueve el desarrollo de la Minería de Datos como área de investigación y fomenta la creación y reinvención de algoritmos más eficientes y que soporten mayor cantidad de datos.

La Minería de Datos se encuentra en un proceso de consolidación y apertura de mercados; uno de estos mercados puede ser el hoy satisfecho con técnicas de Investigación Operativa. La Investigación Operativa usa técnicas de gran variedad de disciplinas científicas como matemática y estadística. Las técnicas tradicionales de análisis de información propuestas por estas

áreas científicas no han tenido un desarrollo equivalente al aumento de los volumenes de datos sobre los cuales se requiere su aplicación hoy en día. En estos casos en que el volumen de datos es muy grande y la Investigación Operativa puede incorporar las ventajas que la Minería de Datos propone para el manejo de grandes volumenes de datos. Si así lo hace se verá beneficiada por el inminente desarrollo de nuevos algoritmos o por el mejoramiento de algoritmos existentes, tarea en la cual se está trabajando intensamente en varias áreas. Esta relación entre Minería de Datos e Investigación Operativa no tiene porque ser de una vía, el área de Investigación Operativa puede aportar su experiencia e intervenir de forma activa en estos desarrollos. Experiencias de este tipo ya se han llevado adelante y es de esperar que se sigan produciendo. En el objetivo de alcanzar la madurez de la Minería de Datos y todos pueden participar.

El estudio de Minería de Datos realizado durante este proyecto de grado muestra la gran gama de técnicas, metodologías y algoritmos utilizados en esta área. Investigación Operativa por su parte es un área igualmente amplia, en la cual se trabaja con gran gama de métodos. Estos métodos no se estudiaron detalladamente durante el proyecto, ya que no forma parte del objetivo del mismo. La amplitud de Investigación Operativa y la falta de documentación detallada de sus métodos significó un contratiempo tanto para la tarea de compararla con Minería de Datos, como para la tarea de estudiar la aplicación de un área en la otra. Igualmente se identificaron múltiples posibilidades de aplicación de técnicas de Minería de Datos en proyectos de Investigación Operativa. Minería de Datos puede ser usada en el preprocesamiento de los datos disponibles; para simplificar el problema reduciendo los datos o aportar información útil para definir los modelos. También es de utilidad en la determinación de los parámetros que componen las funciones objetivo y las restricciones. El estudio realizado durante este proyecto de grado es una primera aproximación entre ambas áreas que señala algunas formas de aplicación de Minería de Datos en IO. Es necesario estudiar en detalle cada una de ellas para poder determinar cuales técnicas de Minería de Datos son de mayor utilidad en cada caso. Es por esto que en la siguiente sección se proponen varias formas de extender este trabajo, para lo cual se sugiere contar primero con un Estado de Arte de Investigación Operativa.

El Estado del Arte de Minería de Datos realizado muestra las técnicas, metodologías y algoritmos utilizados en Minería de Datos; así como también se evalúan algunas de las ofertas de software del mercado. La documentación entregada a tales efectos cumple el objetivo de documentación del Estado

del Arte planteado al comienzo del proyecto. Se ha ejemplificado con un caso de estudio sencillo la aplicación de técnicas y algoritmos de Minería de Datos y sus posibles usos en problemas de Investigación Operativa. Los resultados del experimento desarrollado a partir del caso de estudio confirman que es posible modelar un problema de Investigación Operativa como un problema de Minería de Datos y viceversa. La información obtenida con el modelo de Minería de Datos puede ser usada en cualquiera de las etapas de un proyecto de Investigación Operativa, inyectando nueva información que permita al investigador reforzar los modelos de IO. Se han alcanzado por lo tanto el objetivos planteados de analizar la aplicación de Minería de Datos a Investigación Operativa. Por último se ha desarrollado un nuevo algoritmo de redes neuronales, en particular para un perceptrón multicapa, agregando a las implementaciones estándar de perceptrón multicapa un mejor método para el cálculo del error. Este método se usa en cada paso de la generación de la red y se esperaba con esto obtener redes que produjeran mejores porcentajes de éxito en la clasificación de datos. Estos resultados esperados no pudieron comprobarse con el caso de estudio propuesto, por lo tanto queda pendiente para futuros trabajos su validación.

5.2. Trabajos futuros

Luego de presentadas las innumerables interrelaciones de las áreas de Minería de Datos e Investigación Operativa, este trabajo deja abierta la puerta a múltiples enfoques que permitan ahondar el estudio de las mismas. Desde trabajos puramente teóricos hasta los puramente prácticos, a continuación se mencionan posibles trabajos que se pueden realizar tomando como punto de partida este proyecto de grado. Pero, como se señaló en capítulos anteriores sería de mucha utilidad para proceder con esta investigación contar con un Estado del Arte de Investigación Operativa.

Un tema interesante y que plantea grandes retos es la exploración de interacción de Minería de Datos e Investigación Operativa en un mismo proyecto. No sólo porque puede ser beneficioso que Minería de Datos sea usado en Investigación Operativa o que Investigación Operativa aporte sus conocimientos en el desarollo de algoritmos de Minería de Datos. Sino también por la posibilidad de generar nuevos procesos y metodologías que fusionen técnicas de las dos áreas.

Por otra parte, existe una extensa variedad de algoritmos planteados teóricamente en artículos disponibles al público en general, de los cuales

no existen implementaciones. Tanto esta opción como la de desarrollar un nuevo algoritmo pueden partir de este trabajo. En ambos casos será necesario profundizar el Estado del Arte de Minería de Datos para detallar al mayor nivel posible los fundamentos teóricos de los algortimos desarrollados.

Otra posibilidad es estudiar teóricamente la aplicación de técnicas de Minería de Datos a los problemas típicos de Investigación Operativa, llevando a cabo una evaluación de cuales de estas técnicas son mejores en cada uno de los problemas. En este caso es necesario partir de el Estado del Arte de Minería de Datos desarrollado en este proyecto y de un Estado del Arte de Investigación Operativa.

Finalmente, otra propuesta interesante es dedicarse enteramente a trabajar con un problema de Investigación Operativa del mundo real y aplicar a él todas las técnicas de Minería de Datos que la estructura de sus datos permita. La información obtenida por la aplicación de modelos Minería de Datos se vierte luego a los modelos de Investigación Operativa. Para completar el estudio sería interesante investigar y utilizar un indicador que permita evaluar "cuanto" han mejorado los modelos de Investigación Operativa y los resultados con ellos obtenidos luego de incorporar la información generada por Minería de Datos.

Bibliografía Parte I

[1] Eclipse. *http://www.eclipse.org/*. Ultima visita 31/10/2005.

[2] Gnome data mine. *http://www.togaware.com/datamining/gdatamine/*. Ultima visita 16/05/2006.

[3] The institute for operations research and the management sciences. *http://www.informs.org/*. Ultima visita 16/01/2006.

[4] International federation of operational research societies. *http://www.ifors.org*. Ultima visita 16/01/2006.

[5] International federation of operational research societies. *http://www.orsoc.org.uk o http://www.theorsociety.org/*. Ultima visita 16/01/2006.

[6] Orange data mining friutful & fun. *http://www.ailab.si/orange/*. Ultima visita 16/05/2006.

[7] Proyecto r. *http://www.r-project.org/*. Ultima visita 16/05/2006.

[8] Wiki de documentación de weka. *http://weka.sourceforge.net/wekadoc*. Ultima visita 12/02/2006.

[9] Yale. *http://www-ai.cs.uni-dortmund.de/SOFTWARE/YALE/index.html*. Ultima visita 16/05/2006.

[10] Arnoff Ackoff, Churchman. *Introduction to Operations Research*, 1957.

[11] Chid Apte. *Data Mining Analytics for Business Intelligence and Decision Support. ORMS Today*, page *Sin información de páginas*, 2003.

[12] A.Tuzhilin B.Padmanabhan. On the use of optimization for data mining: Theoretical interactions and ecrm opportunities. In R.Krishnan; A.M.Geoffrion, editor, *Management Science*, volume 49, pages 1327–1343. INFORMS, 2003.

[13] Çaglar Guven. *Operational Research from a Critical Viewpoint*, 1999.

[14] Lam Chan, Wong. Financial time series forecasting by neural networks using gradient learning algorithm and multiple regression weight initialization. *Publicación desconocida*, 1994.

[15] Padhraic Smyth David Hand, Heikki Mannila. *Principles of DataMining, ISBN: 0-262-08290-X*. The MIT Press, 2001.

[16] Universidad de Waikato. *http://www.cs.waikato.ac.nz*. Ultima visita 31/10/2005.

[17] Universidad de Waikato. Weka : Waikato environment for knowledge analysis. *http://www.cs.waikato.ac.nz/ml/weka/index.html*. Ultima visita 31/10/2005.

[18] Campbell et. al. Optimizing customer mail streams at fingerhut. *Interfaces*, 2001.

[19] G.Lieberman F.Hillier. *Introduction to Operations Researchs*. McGraw-Hill. ISBN: 0-072321-69-5, 7ma edition, 2001.

[20] Jin Gupta and Homma. *Static and dynamic neural networks. From fundamentals to Advanced Theory*. John Wiley & Sons, Inc. ISBN: 0-471-21948-7, 2003.

[21] R.Sarker H.Abbassan, C.Newton. *Data mining: A Heuristic Approach*. University of New South Wales, Australia. ISBN: 1-930708-25-4, 2002.

[22] M.Boyd I.Kaastra. *Designing a neural network for forecasting financial and economic time series. Neurocomputing*, page *Sin información de páginas*, 1995.

[23] Sigmound Olaffson. *What is Decision Support?* Sin información de la publicación, page *Sin información de páginas*, 2001.

[24] S. Khan R.Ganguly, A.Gupta. Data mining and decision support for business and science. In J.Wang, editor, *Encyclopedia of Data Warehousing and Mining*, volume 1, pages 233–268. Idea Group Inc. ISBN 1-59140-559-9, 2006.

[25] Y.Guo R.Grossman. Data mining and knowledge discovery. In High performance data mining - Scaling Algorithms, Applications and Systems, volume 3, pages 235–236. Kluwer Academic Publishers, 1999.

[26] R. Robinson. *More profit, productivity, and cost reduction from Operations Research. INFORMS*, Año de publicación desconocido.

[27] R. Sharda S. Menon. *Data mining update: new modes to pursue old objectives. ORMS Today*, 1999.

[28] N. Indurkhya S. Weiss. *Predictive Data Mining: A Practical Guide*. Morgan Kaufmann. ISBN: 1-558-60403-0, 1998.

[29] Shobrys Sery, Presti. Optimization models for restructuring basf north americas distribution system. In *Decision Analysis Applications*, pages 55–65. INFORMS, 2001.

[30] The OR Society. *A common destiny: measurement and synthesis*, 2001.

[31] M.Houle V.Estivill-Castro. Approximating proxomity for fast and robust distance-based clustering. In R.Sarker H.Abbassan, C.Newton, editor, *Data mining: A Heuristic Approach*, pages 22–46. University of New South Wales, Australia. ISBN: 1-930708-25-4, 2002.

[32] V.Kecman. *Learning and Soft Computing - Support Vector Machines, Neural Networks, and Fuzzy Logic Models*. The MIT Press, Inc. ISBN: 0-262-11255-8, 2001.

[33] Ian Witten and Eibe Frank. *Data mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, Inc. ISBN: 0-12-088407-0, 2da edition, 2005.

[34] Peter Cabena. Pablo Hadjinian. Rolf Stadler. JaapVerhees. Alessandro Zanasi. *Discovering Data Mining: From Concept to Implementation*. Prentice Hall. ISBN: 0-137-43980-6, 1998.

Parte II

Apéndices

Apéndice A

Minería de Datos

A.1. Introducción

El conocimiento es un recurso estratégico para el desarrollo económico y social. La información es el principal elemento en el proceso de generación, adquisición, gestión y transmisión del conocimiento. La mayoría de decisiones de empresas, organizaciones e instituciones se basan en información de experiencias pasadas extraídas de fuentes muy diversas. Es por eso que organizaciones de diversa envergadura dedican tiempo y recursos a la tarea de recolectar y almacenar datos, esperando poder convertirlos algún día en información de valor. A menudo, términos tales como datos, información, inteligencia y conocimiento se utilizan indistintamente. No obstante, se trata de cosas muy diferentes. Los datos son información en bruto, sin procesar. Se trata de una relación de hechos, cifras y registros que, por sí mismos, es posible que no tengan ningún valor. La información son datos procesados. Antes de que puedan ser considerados información, los datos deben ser clasificados, organizados y presentados en un formato que el usuario entienda. El conocimiento es información procesada, aplicada a fines específicos. [3].

La tecnología informática constituye en la actualidad una infraestructura fundamental dentro de las organizaciones, existen enormes volúmenes de datos almacenados en computadoras en todo el mundo. La variedad de los datos disponibles comprende información financiera, científica, gubernamental, educativa y otros. A pesar del esfuerzo realizado en recolectar los mismos y los grandes volúmenes alcanzados, la realidad indica que sólo una pequeña porción de ellos podrá ser usada. Eso se debe a que en muchos casos, los volúmenes son inmanejables o la estructura de los datos es muy

compleja. La principal razón tras este problema es que el esfuerzo original de crear estos repositorios de datos se enfocó al almacenamiento eficiente de los mismos y no al diseño de un plan de almacenamiento que facilitara su análisis. Se ha alcanzado un punto en el cual el enorme volumen de datos almacenados y su rápido crecimiento han excedido la capacidad humana para la comprensión de los mismos. Se hace entonces imprescindible establecer procesos informáticos que conviertan los grandes volúmenes de datos existentes en experiencia y conocimiento. Las tecnologías, métodos y herramientas asociadas con estos procesos se han desarrollado notablemente en los últimos años. Su fin es explorar y analizar las bases de datos para extraer información existente, ayudar en la toma de decisiones, así como a crear sistemas inteligentes capaces de entenderlos.

El término data mining, - en español Minería de Datos - fue introducido en los años 90; refiere a la extracción de conocimiento de grandes cantidades de datos.

"Data mining es el análisis de, a menudo grandes, conjuntos de datos de observación para encontrar relaciones insospechadas y para resumir los datos de nuevas maneras que son comprensibles y útiles al dueño los datos." [4]

El término no es realmente el más adecuado para la actividad que representa. La definición de minería describe el proceso de encontrar una pequeña cantidad de material valioso en una gran cantidad de materia prima. En el caso de la minería de datos, los datos son la materia prima y no el objetivo que es el conocimiento, la información. Sería más apropiado llamarle minería de Conocimiento, pero Minería de Datosse ha convertido en el término popular. Es posible que este término intente enfatizar la importancia que los grandes volúmenes de datos tienen en la Minería de Datos, si es así, es apropiado, pues este punto amerita especial consideración.

Es posible identificar muchos términos que tienen un significado similar a minería de datos, tal como minería de conocimiento en bases de datos, extracción de conocimiento, análisis de datos, análisis de patrones de datos, arqueología de datos. Resulta también común encontrar el término Minería de Datos usado como sinónimo de otro término de uso popular Knowledge Database Discovery (KDD) [1]. Las opiniones de si la minería de datos es lo mismo que KDD están dividas; algunos describen a la Minería de Datos simplemente como un paso dentro del proceso de descubrimiento de conocimiento en bases de datos.

[1] KDD, en español descubrimiento de conocimiento en bases de datos

"KDD es el proceso de encontrar información útil y patrones en los datos." "Minería de Datos es el paso en el proceso KDD que efectivamente accede a los datos." [5]

"KDD es el proceso de identificar en los datos estructuras válidas, novedosas, potencialmente útiles y entendibles." "Minería de Datos es el paso en el proceso KDD que se ocupa de los medios algorítmicos por los cuales los patrones o modelos se obtienen de los datos bajo limitaciones aceptables de eficacia computacional." [6]

De esta definición se desprende una diferencia considerable, la utilidad de los patrones encontrados, la Minería de Datos produce patrones que no necesariamente serán útiles. El proceso KDD comprende un conjunto de pasos además de la Minería de Datos, como la limpieza de datos y validación de resultados. Según esta visión, la Minería de Datos es sólo un paso en el proceso de descubrimiento de conocimiento, no obstante es un paso esencial que descubre los patrones ocultos en los datos para su evaluación. La Minería de Datos se diferencia de otros tipos de análisis de datos en dos aspectos importantes. En primer lugar, no comienza con una hipótesis clara, sino que genera hipótesis a partir de los datos. En segundo lugar, es un análisis secundario de grandes volúmenes de datos; los datos están disponibles en algún repositorio generado con un fin primario específico y la Minería de Datos saca provecho de ellos.

Lamentablemente no hay un acuerdo general en que es y que no es Minería de Datos y tampoco ayuda a alcanzar un acuerdo el hecho de que la Minería de Datos se encuentre en un área compartida por varias disciplinas dentro y fuera del campo informático. Fuera del campo informático la estadística y dentro del campo informático las base de datos, el aprendizaje de máquina, el reconocimiento de patrones, las redes neuronales, la visualización de los datos, la recuperación de datos, alto rendimiento computacional, procesamiento de imágenes y de señal y análisis de datos espacial. Las opiniones son muy variadas, desde las que señalan a la Minería de Datos simplemente como un conjunción de metodología y tecnología preexistente como los que lo catalogan como un campo nuevo y revolucionario.

La Minería de Datos es un proceso iterativo cuyo progreso se define por el descubrimiento de información por métodos automáticos o manuales. No existe un único enfoque para enfrentar los problemas de Minería de Datos, en cambio, están disponibles una amplia gama de metodologías y técnicas que pueden ser aplicadas a los mismos. Hay dos claves para el éxito:

encontrar una formulación exacta del problema que se intenta solucionar o un objetivo y utilizar un conjunto de datos adecuado. La mejor estrategia es que el usuario experimente interactivamente con los datos y de esta forma encuentre el enfoque que mejor se ajuste al conjunto de datos y sus objetivos. Es por eso que las herramientas de minería de datos deben ser guiadas por usuarios expertos, que entiendan los datos y la naturaleza general de los métodos analíticos involucrados. Los mejores resultados se obtienen cuando se balancea adecuadamente el conocimiento humano en la definición de los problemas y las metas, con la capacidad de búsqueda de las computadoras.

"Data mining es la búsqueda de información de valor en grandes volúmenes de datos. Es un esfuerzo cooperativo entre humanos y computadoras." [7]

Un sistema de Minería de Datos tiene potencial para generar millares o aún millones de patrones. Sólo una pequeña fracción de ellos resultan ser realmente interesantes para los usuarios, por lo tanto sólo estos serán útiles. Los patrones interesantes son los que representan conocimiento. Un patrón es interesante cuando: es fácilmente entendido por el usuario, es posible validarlo en datos nuevos o de prueba, es potencialmente útil, es nuevo o valida una hipótesis que el usuario intentó confirmar. Ya que la búsqueda se enfoca a los patrones interesantes es común plantearse si el proceso de Minería de Datos los genera todos. Es poco realista e ineficaz que los sistemas de Minería de Datos generen todos los patrones posibles. En vez de esto se debe controlar la generación de patrones enfocando la búsqueda con el uso medidas de interés. Esto es a menudo suficiente asegurar la completitud del algoritmo. Además, sería deseable que los sistemas de Minería de Datos generen sólo patrones interesantes. Esto sería más eficiente tanto para el sistema de Minería de Datos como para el usuario y se puede lograr mediante la optimización del problema. Esta idea aunque es intuitivamente lo que el usuario desea, es nueva en cuanto al nivel de investigación a la que ha llegado y aún no ha sido aplicada en los productos existentes.

La elección de los datos es fundamental y entre las consideraciones a tener en cuenta para realizar la elección, la consideración de volumen es muy importante. A pesar de existir quienes opinan que en Minería de Datos más es mejor; como en todas las situaciones que involucran datos, la calidad es fundamental. Lo que si es posible asegurar es que más es preferible. La Minería de Datos es más efectiva cuando el volumen de datos es grande y oculta patrones que podían ser detectados en menores cantidades de datos. Determinar el volumen adecuado dependerá de un conjunto variado de factores de distinta naturaleza e inherentes al problema a tratar.

A.2. Reseña histórica y aportes de otras áreas

Como la mayoría de los procesos intelectuales, el desarrollo de la Minería de Datos se ha realizado incrementalmente, construyéndose en las últimas cuatro décadas sobre bases de conocimiento preexistentes [8]. La Minería de Datos implica una integración de disciplinas y múltiples técnicas tales como tecnología de base de datos, estadística, aprendizaje de máquina, reconocimiento de patrones, redes neuronales, visualización de los datos, recuperación de datos, alto rendimiento computacional, procesamiento de imagen y de señal y análisis de datos espacial. Este es uno de esos casos, en que el trabajo realizado en diversas áreas se pone en uso en forma conjunta para la resolución de problemas.

"Data mining es un campo interdisciplinario que une técnicas de aprendizaje de máquina, reconocimiento de patrones, estadística, bases de datos, y visualización para tratar la aplicación de extracción de información de bases de datos grandes."[8]

A continuación analizaremos brevemente los aportes más importante de cada una de estas áreas al surgimiento y posterior evolución de la Minería de Datos.

A.2.1. El aporte de la estadística

Por ser la estadística la rama más antigua de las que influencian a la minería, es de esperar que haya sido en ésta que por primera vez surgió la práctica de esta actividad, aún cuando se conociera por otro nombre análisis de datos. Dentro del campo estadístico existen dos referentes muy importantes de la Minería de Datos, ellos son John Tukey y Leo Breiman.

John Tukey, considerado uno de los contribuyentes más importantes de la estadística moderna, en su artículo El futuro del análisis de datos, publicado en los Anales de la estadística en 1962 sostiene que la estadística matemática ignora el análisis de datos del Mundo real. John Tukey impulsó una vuelta a los orígenes de la estadística científica; usando los métodos modernos, en los cuales la descripción estadística de los datos es esencial. Su trabajo pionero en extraer patrones y correlaciones ocultos en grandes cantidades de datos lo distinguen dentro del campo estadístico como el precursor de la Minería de Datos. Junto a él otros representantes de la estadística han aportado trabajo de valor a esta área.

Leo Breiman, organizó en 1977 la Conferencia para el Análisis de conjuntos de datos grandes y complejos. El planteó la existencia de dos culturas en el uso del modelado estadístico para alcanzar conclusiones de datos. Uno asume que los datos son generados por un modelo estocástico dado de los datos. El otro utiliza modelos algorítmicos y trata el mecanismo de los datos como desconocido. La comunidad estadística ha estado limitada al uso casi exclusivo de los modelos de los datos. Esta restricción ha conducido a la teoría irrelevante, conclusiones cuestionables, y ha mantenido a los estadísticos alejados de trabajar en una gama grande e interesante problemas actuales. El modelado algorítmico, en teoría y práctica, se ha convertido rápidamente en campos fuera de la estadística. Puede ser utilizado en conjuntos de datos grandes y complejos o como alternativa más exacta e informativa a conjuntos de datos más pequeños. Leo Breiman presentó en artículos y conferencias durante muchos años su idea de las dos culturas del modelado estadístico y afirmó que si la meta de la estadística como campo es utilizar datos para solucionar problemas, entonces debe liberarse de la dependencia exclusiva de los modelos de los datos y adoptar un sistema más diverso de herramientas, entre los cuales figura la Minería de Datos. Leo Breiman fue galardonado en 2005 con el premio a la innovación en Data Mining y descubrimiento de conocimiento en reconocimiento a sus aportes a ambas áreas.

El desarrollo de la mayoría de las técnicas estadísticas eran, hasta hace poco, basados en teoría y métodos analíticos, los mismos funcionaban bien con modestos volúmenes de datos. El aumento del poder de las computadoras y la baja de costos, unido a la necesidad de analizar enorme cantidad de datos, han permitido el desarrollo de nuevas técnicas basadas en la exploración por fuerza bruta de soluciones. Nuevas técnicas incluyen algoritmos recientes como redes neuronales y árboles de decisión, y nuevas formas de viejos algoritmos como el análisis de discriminantes. El punto clave es que la Minería de Datos es la aplicación de estos y otras técnicas a problemas comunes. La estadística tiene sus raíces en matemáticas, y por tanto, hace énfasis en el rigor matemático de establecer que algo es detectado en el ámbito teórico antes de ser probado en la práctica. La resistencia de los estadísticos clásicos, basaba en la falta de respaldo teórico en la búsqueda de patrones por fuerza bruta, ha mantenido abierta la discusión referente a si la estadística como campo debe considerar la minería de datos como una subdisciplina o dejarla a los informáticos. Se puede resumir la inquietud existente entre los interesados en incorporar la Minería de Datos como una herramienta estadística en las palabras de Jerome H.Friedman.

"Ya no somos el único jugador en la ciudad. Hasta hace poco tiempo, si uno estaba interesado en análisis de datos, la estadística era uno de los pocos (incluso remotamente) campos apropiados en los cuales trabajar. Este no es más el caso. Ahora hay muchas otras ciencias emocionantes orientadas a los datos que están compitiendo con nosotros por los clientes, los trabajos, y nuestros propios estadísticos. [9]

El uso de la estadística es fundamental para que la Minería de Datos crezca; ésta es una oportunidad única para que estadistas e informáticos trabajen en conjunto. El orgullo y ansias de independencia de las dos áreas ha demorado la evolución de la Minería de Datos y puede seguirlo haciendo. Se requiere de la madurez de ambas ramas para que la colaboración sincera entre ellas permita alcanzar el éxito.

A.2.2. El aporte de la inteligencia artificial

La inteligencia artificial o inteligencia de máquina, cubre una vasta gama de teorías y prácticas. Su finalidad consiste en crear teorías y modelos que muestren la organización y funcionamiento de la inteligencia. Los esfuerzos por reproducir algunas habilidades mentales humanas en máquinas se remontan muy atrás en la historia. La inteligencia artificial surgió en 1943; cuando Warren McCulloch y Walter Pitts propusieron un modelo de neurona del cerebro humano y animal, proporcionando una representación simbólica de la actividad cerebral. El postulado era "El cerebro es un solucionador inteligente de problemas, de modo que imitemos al cerebro". A pesar de que la investigación sobre el diseño y las capacidades de las computadoras habían comenzado algún tiempo antes, fue recién en 1950 que la idea de una máquina inteligente comenzó a cautivar la atención de los científicos. El detonante fue el artículo "Maquinaria Computacional e Inteligencia" de Alan Turing. El trabajo de Turing fue continuado por John Von Neumann, cuya contribución central fue la idea de que las computadoras deberían diseñarse tomando como modelo al cerebro humano. Sin embargo, esta línea de investigación pronto encontró serias limitaciones. McCulloch, a mediados de los cincuentas, formuló una posición radicalmente distinta al sostener que las leyes que gobiernan al pensamiento deben buscarse entre las reglas que gobiernan a la información y no entre las que gobiernan a la materia. Fue entonces que se llegó a la idea de que la analogía sería más eficiente si se estudiaran entonces las funciones del cerebro, es decir, sus capacidades como procesador de información, en vez de sus funciones físicas.

Esta disciplina, que se construye sobre la heurística en comparación con

estadística, procura aplicar el proceso de pensamiento humano a los problemas estadísticos. Este acercamiento requiere mucha potencia computacional, es por eso que no era práctico hasta los años 80, cuando las computadoras comenzaron a ofrecer esta potencialidad a precios razonables. Actualmente, el mayor esfuerzo en la búsqueda de la inteligencia artificial se centra en el desarrollo de sistemas de procesamientos de datos que sean capaces de imitar a la inteligencia humana, realizando tareas que requieran aprendizaje, solución de problemas y decisiones.

A.2.3. El aporte del aprendizaje de máquina

El aprendizaje de máquina es un área importante y de gran envergadura en informática, es la invención y uso de técnicas que permitan a la máquina aprender. El aprendizaje de máquina mezcla la heurística de la inteligencia artificial con análisis estadístico avanzado. Es un método para crear programas de computadora mediante el análisis de conjuntos de datos. Los métodos de aprendizaje de máquina permiten a un programa de computadora analizar automáticamente un conjunto grande de datos y decidir qué información es la más relevante. La información obtenida se puede utilizar para hacer predicciones o ayudar a tomar decisiones más rápidas y más acertadas.

En contraste con la estadística, la comunidad de aprendizaje de máquina tiene sus orígenes en la práctica computacional. Esto ha conducido a una orientación práctica de los problemas, al deseo de demostrar algo para ver como se comporta, sin esperar una prueba formal de efectividad. En lugar del énfasis que la estadística realiza en los modelos, el aprendizaje de máquina tiende para acentuar los algoritmos. La palabra aprendercontiene la noción de un proceso, de un algoritmo implícito.

A.2.4. El aporte del reconocimiento de patrones

Los patrones ocurren en todos los aspectos del mundo que percibimos: actividades sociales, comportamiento animal, física y química, vida emocional. El reconocimiento de patrones puede ser visto como la asignación de una etiqueta a una observación. Simbolizar estos patrones con palabras y números nos permite describir los patrones y su comportamiento como objetos simbólicos. Asociar estos objetos simbólicos entre ellos es una forma de representar nuestro entendimiento del mundo. La estructura de interrelación de los símbolos es lo que llamamos modelo. Por lo tanto extraer símbolos

apropiados y entender las relaciones de los mismos y el mundo, está íntimamente ligado con nuestra habilidad de tomar decisiones. Mucho del esfuerzo del hombre a lo largo de la historia ha sido dedicado a descubrir patrones útiles para construir modelos útiles.

El proceso automatizado de reconocimiento de patrones requiere de un programa con instrucciones específicas, como por ejemplo ecuaciones matemáticas, que den la relación entre entradas y salidas. Formular esas ecuaciones matemáticas, o construir el modelo, es el tema central de la automatización del proceso. La Minería de Datos es la última de una larga lista de herramientas para detectar patrones nuevos y significativos para mejorar nuestra comprensión del mundo.

"Data mining es el proceso de descubrir nuevas relaciones, patrones y tendencias significativos examinando grandes cantidades de datos almacenados en repositorios, usando tecnologías de reconocimiento de patrones así como técnicas estadísticas y matemáticas." [10]

A.2.5. El aporte de las bases de datos

Desde la década del sesenta hasta hoy, las tecnologías de bases de datos y de información han evolucionado desde sistemas primitivos de tratamiento de archivos a potentes y sofisticados sistemas de bases de datos. Esta evolución puede describirse a través de los hitos más importantes que la componen.

En la década del setenta se crearon los sistemas de base de datos relacionales, las primeras herramientas de modelado y técnicas de indexación y organización. Métodos eficientes OLPT [2], donde una consulta es tratada como una transacción contribuyeron a la aceptación de la tecnología relacional como una herramienta eficiente para el manejo de grandes cantidades de datos. Fue en esta época en que se comenzó a trabajar en la comunicación con el usuario, mejorando la flexibilidad proporcionada para acceder a los datos a través de lenguajes de consulta, lenguajes de procesamiento e interfaces amigables.

En la década de los ochenta proliferó la definición de nuevos modelos para las bases de datos: relacional extendido, orientado a objetos, relacional orientado a objetos, deductivos y orientados a aplicación como el modelo

[2]OLPT: On line transaction processing. Een español, procesamiento de transacciones en línea

espacial, temporal, multimedia y científico. La distribución y las formas de compartir los datos se estudiaron con profundidad y los sistemas de bases de datos heterogéneos emergieron en esta década así como también sistemas de información basados en Internet. Como en todos los ámbitos, el surgimiento de Internet marco la evolución de la industria de la información.

En la década del noventa el hito más importante fue el surgimiento del data warehouse, un esquema de almacenamiento unificado para un repositorio de fuentes de datos heterogéneas. La tecnología de data warehouse [3] trajo consigo técnicas avanzadas de análisis de datos como el OLAP [4], que incluye técnicas de resúmen, consolidación, agregación y técnicas muy avanzadas de visualización. A finales de esta década surgió otra nueva técnica relacionada a las bases de datos, la Minería de Datos.

En la actualidad estamos siendo participes del afianzamiento de la Minería de Datos y su incorporación en el área de las bases de datos. En el OLAP, el análisis y exploración de los datos es dirigido enteramente por el usuario. OLAP puede ser visto como una extensión de las consultas SQL a casos en que la realización de las consultas en línea sobre la base de datos es una tarea prohibitiva. Las técnicas de minería de datos permiten la exploración computacional de los datos. Esto abre la posibilidad a una nueva forma de interacción con las bases de datos, especificando consultas a un nivel de abstracción mucho mayor que el SQL e incluso el OLAP. También facilita la exploración para problemas que, debido a sus grandes dimensiones, serían difíciles de analizar y resolver para un usuario. Es por eso que la Minería de Datos se considera una de las fronteras más importantes de los sistemas de base de datos y uno de los usos más prometedores de las base de datos en la industria de la información.

A.2.6. El aporte tecnológico

El progreso constante de la tecnología de hardware ha sido el catalizador en el progreso reciente de la Minería de Datos, así como también la baja de los costos. Dentro del avance tecnológico se han dado dos situaciones, la mejora de la tecnología existente y la generación de nuevas tecnologías. La construcción de computadoras potentes tanto en poder de procesamiento como en disponibilidad de almacenamiento y además a precios cada día más accesibles han dado un gran empuje a la industria de las bases de datos

[3] Ver definición en Glosario G

[4] OLAP, On line analytical processing. En español, procesamiento de análisis en línea

y la información. Estas mejoras han hecho posible la proliferación de un gran número de bases de datos y repositorios de información para el análisis de datos. El desarrollo de nuevas tecnologías, como por ejemplo los scanners y sensores digitales, han disparado el volumen de datos recolectados. Los científicos están en el extremo más alto en la recolección de datos en la actualidad. Los instrumentos científicos, como los microscopios, pueden generar fácilmente terabytes de datos en un período corto de tiempo y almacenarlos automáticamente en la computadora. La tecnología existente se difundirá en los próximos años a otros ámbitos en los que posibilitará mayor, mejor y más rápida recolección de datos; y los datos son la materia prima de la Minería de Datos.

A.2.7. El aporte de Internet

Nos encontramos en la era de la información; la expansión de Internet, ha causado un crecimiento exponencial en las fuentes de información y también en unidades de almacenaje de información. Se ha producido en los últimos 3 o 4 años un aumento dramático de los servidores de Internet, en números este aumento es directamente proporcional al aumento de datos almacenados en Internet. Este incremento no ha parado y es de esperar que la brecha entre la cantidad de datos recolectados y la capacidad de analizar los datos siga creciendo rápidamente en los años por venir.

La Web es un inmenso almacén de conocimiento, construido en forma descentralizada y colaborativa. Sin embargo, la información que abunda en la red no se almacena de forma estructurada. Esta compuesto de billones de documentos de hipertexto, estos contienen texto e hiperlinks a otros documentos distribuidos a través de la Web. Esta situación plantea un gran desafío a aquellos que intentan buscar con eficacia información de alta calidad y descubrir conocimiento oculto en sus páginas. Web Mining es un tipo de Minería de Datos que permite descubrir información de valor en la Web.

A.3. Comparación de Minería de Datos con otras soluciones

A.3.1. Minería de Datos versus estadística

El hecho de que la Minería de Datos esta basada en algoritmos y métodos estadísticos genera dudas en cuanto a la diferencia entre la aplicación de Minería de Datos y la aplicación de estadística pura a un problema. Tanto la Minería de Datos como las estadísticas refieren a aprender de los datos o convertir datos en información. Las técnicas estadísticas se centran generalmente en técnicas confirmatorias, mientras que las técnicas de Minería de Datos son generalmente exploratorias y son menos restrictivas que las estadísticas. Así, cuando el problema trata de refutar o confirmar una hipótesis, ambas técnicas pueden ser utilizadas, donde la más robusta es la estadística. Sin embargo, existen numerosos problemas en los cuales una de ellas se adecua mejor que la otra.

Minería de Datos es preferible a la estadística:

- Cuando el objetivo es meramente exploratorio. En estos tipos de problemas surge la necesidad de delegar parte del conocimiento analítico en técnicas de aprendizaje de máquina. Minería de Datos es mejor cuando no existen hipótesis de partida y se pretende buscar conocimientos nuevos.

- Cuanto mayor es la dimensionalidad del problema. Cuantas más variables entran en el problema, más difícil resulta encontrar hipótesis de partida interesantes; a veces aún cuando fuese posible, el tiempo necesario no justificaría la inversión. En ese caso, las técnicas de Minería de Datos permiten encontrar nuevas relaciones para luego concretar la investigación sobre las variables más interesantes.

- Cuando los datos no satisfacen los requerimientos del análisis estadístico. Las variables deben ser examinadas para determinar que tratamiento permite adecuarlas al análisis estadístico, lo cual no es posible o conveniente en todos los casos. La Minería de Datos es menos restrictivo que la estadística es posible utilizarla con los mínimos supuestos posibles, permite escuchar a los datos.

- Cuando el conjunto de datos es muy dinámico. Un conjunto de datos poco dinámico permite que la inversión en un análisis estadístico sea justificada. En un marco de metodología rígida y respuestas a preguntas muy concretas, las conclusiones estadísticas van a tener un ciclo de vida largo. Sin embargo, en un almacén de datos demasiado dinámico las técnicas de Minería de Datos permiten explorar cambios e inciden positivamente en la actualización del conocimiento a mediano y largo plazo.

El análisis estadístico es más adecuado que la Minería de Datos:

- Cuando el objetivo de la investigación es encontrar causalidad. Si se pretende determinar cuales son las causas de ciertos efectos, deberemos utilizar técnicas de estadística. Las relaciones complejas que subyacen a técnicas de Minería de Datos impiden una interpretación certera de diagramas causa-efecto.

- Cuando las conclusiones han de ser extensibles a otros elementos de poblaciones similares. Este tipo de problemas se relaciona con situaciones en las que se dispone exclusivamente de muestras. En Minería de Datos, se generarán modelos y luego habrán de validarse con otros casos conocidos de la población, utilizando como significación el ajuste de la predicción sobre una población conocida.

A.3.2. Minería de Datos versus OLAP

Las técnicas OLAP [5] se están usando cada día más en sistemas de apoyo en la toma de decisiones para proporcionar análisis de los datos. OLAP provee al usuario de técnias avanzadas que le permiten analizar los datos. Las herramientas OLAP presentan a los usuarios una visión multidimensional de los datos que se conoce con el nombre de cubo. El usuario puede navegar [6] el cubo mediante métodos asistidos y en general visuales. En el OLAP el análisis es dirigido por el usuario. En cambio la Minería de Datos es dirigida por las herramientas y algoritmos y guiada por el usuario. El usuario puede intervenir a lo largo de todo el proceso, proporcionando datos de entrada, observaciones y suposiciones. En OLAP, el usuario recibe el conjunto el cubo construido con los datos para los que fue diseñado, y aunque es posible, es también poco probable que los usuarios participen en el diseño del cubo.

[5] OLAP: On line Analytical Processing

[6] navegar: analizar los datos desde distintas perspectivas

Las soluciones OLAP no pueden predecir que pasará en datos futuros, que es uno de los objetivos de la Minería de Datos. El OLAP en cambio tiene por objetivo dar al usuario una visión global de los datos, en cambio minería de datos trabaja en detalle, descubriendo patrones en los datos para poder generalizar a partir de ellos, pero la tarea de generalización se deja al usuario que interpreta los resultados. En el OLAP el usuario investiga los datos o dada una hipótesis analiza los datos con el objetivo de verificarla o refutarla; en Minería de Datos no hay hipótesis de partida, la misma surge de los datos.

A.4. Requerimientos de la Minería de Datos

M.Chen, J.Han, P.Yu [11] plantean un conjunto de requerimientos deseables en el desarrollo de técnicas de Minería de Datos. Los requerimientos: manejo de tipos de datos heterogéneos, eficiencia y escalabilidad de los algoritmos, utilidad y correctitud de los resultados, uso de diferentes tipos de fuentes de datos, obtención de diferentes tipos de resultados, Minería de Datos interactiva, protección y privacidad de los datos manejados (tanto de los datos de entrada como de los resultados). A continuación se detalla en que consiste cada uno de estos requerimientos.

A.4.1. Tipos de datos heterogéneos

Un sistema potente debe ser capaz de efectuar Minería de Datos en diferentes tipos de datos y de bases de datos. Los datos de entrada para un proceso de Minería de Datos se pueden clasificar en datos estructurados, datos semi-estructurados, y datos no estructurados. Los datos estructurados se conocen también como datos tradicionales (datos numéricos, alfanuméricos), mientras que los datos semi-estructurados y no estructurados son datos no tradicionales (también llamados los datos multimedia). La mayoría de los métodos de Minería de Datos actuales y de las herramientas comerciales se aplican a datos tradicionales. Sin embargo, el desarrollo de herramientas de Minería de Datos para datos no tradicionales, así como los interfaces para su transformación en formatos estructurados, está progresando rápidamente. En lo que respecta a bases de datos el tipo más común es el de las bases relacionales, es por eso imprescindible que la Minería de Datos funcione efectivamente en datos relacionales. Además, muchas bases de datos contienen

datos más complejos, como datos estructurados y objetos complejos, hipertexto, datos espaciales, datos temporales y más. La diversidad de tipos hace poco realista que un mismo sistema funcione correctamente con todos los tipos de datos existentes y genera diversificación en cuanto a las herramientas existentes donde surgen especializaciones según el tipo de datos.

Especializaciones de Minería de Datos

Minería de texto El texto es uno de los formatos de datos más usados, es además un formato natural de intercambio de información entre las personas, lo cual lo hace muy abundante. [12] Existe enorme cantidad de información almacenada en documentos y variadas formas de intercambiar esta información como correos electrónicos, publicación de páginas en la web, publicaciones en formato electrónico y bibliotecas digitales.

Los algoritmos de la Minería de Datos han sido desarrollados en su mayoría para extraer información de conjuntos de datos estructurados, pues los tipos estructurados y semi estructurados son los que más abundan en las bases de datos. Los datos de tipo texto no son en general estructurados y si tienen cierta estructura esta varía según el idioma en que están escritos. Dentro de la minería de datos han surgido técnicas exclusivamente desarrolladas para descubrir información en datos de texto que se conoce como Text Mining [7].

Minería de imagen y video (datos multimedia) De la misma forma que la minería de texto, las imágenes son datos estructurados. El contenido de una imagen es de naturaleza visual y su interpretación es subjetiva. Como en otros casos, se desearía que la máquina interpretara la imagen de la forma en que un humano lo haría; pero el proceso de interpretación humano es mayormente desconocido, por lo tanto no reproducible. La minería de imágenes es relativamente nueva, como lo es la recolección de imágenes respecto a la recolección de otros tipos de datos. El avance del hardware en el área en los últimos años permite recolectar imágenes de calidad con mayor facilidad y rapidez. La minería de imágenes y video tiene como objetivo extraer patrones de grandes conjuntos de imágenes. Pero los algoritmos para minería de imágenes y video deben ser diferentes a los algoritmos comunes de Minería de Datos ya que los mismos tratan con datos espaciales y con múltiples interpretaciones de patrones. Debido a la complejidad de análisis

[7] Text Mining; en español minería de texto

requerido, la minería de imágenes y video es una tarea interdisciplinaria que requiere experiencia en visión de computadora, reconocimiento de patrones, procesamiento de imágenes, recuperación de imágenes, Minería de Datos, aprendizaje de máquina, base de datos, inteligencia artificial, y posiblemente compresión. La compresión de los datos se hace muy necesaria, ya que los tamaños de las imágenes y video son privativos aún con las facilidades actuales de almacenamiento y la baja de costos de hardware. Se está estudiando en la actualidad la posibilidad de efectuar minería de imágenes en imágenes comprimidas[12].

Web Mining La web es una la mayor colección de datos existente. La extracción de información de la web es un área de gran interés dentro de la Minería de Datos. Los datos contenidos en la web abarcan datos semiestructurados (HTML[8], XML[9]), hipervínculos, información dinámico. La web ha creado un medio de publicación de información de acceso libre y enormes proporciones. Estos factores han permitido que las personas utilicen la web y las bibliotecas digitales de forma interactiva. Sin embargo, los motores de búsqueda actuales no son eficientes en la búsqueda de información en la web. Los problemas más comunes son la abundancia de información de poco o nulo valor para el usuario, ocultamiento de datos tras motores especializados de búsqueda, interfaces de búsqueda limitadas basadas en claves.

Web Mining es el uso de técnicas de Minería de Datos para buscar, extraer y evaluar información de la web. Si consideramos la naturaleza de los datos en la web es obvia la influencia de otra especialización de la Minería de Datos como la minería de texto y la Minería de Datos multimedia.

A.4.2. Eficiencia y escalabilidad de los algoritmos

Los algoritmos de Minería de Datos deben ser eficientes y escalables a grandes bases de datos. El tiempo de ejecución de los algoritmos debe ser predecible y por supuesto razonable. Los algoritmos de complejidad exponencial o de grados polinómicos altos no serán útiles para bases de datos de gran tamaño.

[8]HyperText Markup Language
[9]Extensible Markup Language

A.4.3. Utilidad y correctitud de los resultados

Existe una motivación latente en Minería de Datos de estudiar y medir la calidad del conocimiento obtenido. Ante esto es necesario saber que se entiende por calidad del conocimiento, ya que parece ser una apreciación totalmente subjetiva. Para evaluar la calidad de los resultados se pueden considerar muchas variables; como el interés y la confianza en los mismos. Es posible construir modelos analíticos, estadísticos, o de simulación que permitan medir.

A.4.4. Diferentes tipos de resultados

Diferentes tipos de conocimientos pueden ser descubiertos durante el proceso de Minería de Datos. Además el usuario deseará analizar los resultados desde diferentes puntos de vistas y presentarlos de diferentes formas. Esto requiere que los requisitos y el conocimiento descubierto se expresen en lenguajes de alto nivel o en forma gráfica. Esto permite que los objetivos de la Minería de Datos puedan ser especificados por cualquier tipo de usuario, sin necesidad de que sea un experto en Minería de Datos. De la misma forma los resultados podrán ser entendidos fácilmente y usados directamente por los usuarios.

A.4.5. Diferentes fuentes de datos

Los tipos de fuentes de datos que es posible utilizar en Minería de Datos son diversos y están relacionados generalmente con los tipos de datos a minar. Las bases de datos son, sin duda, el tipo preferido de fuente de datos. La disponibilidad de bases de datos es amplia, heterogénea y distribuida. Las bases de datos se pueden clasificar de acuerdo a dos criterios: los modelos de bases de datos o el tipo de datos que contienen. Según el modelo de base de datos tenemos bases de datos relacionales, transaccionales, orientadas a objetos, orientadas a objetos relacionales o data warehouse [10]. Según el tipo de datos tenemos bases de datos espaciales, temporales, multimedia, de texto, de series de tiempo, la Web. Es un desafío para la Minería de Datos trabajar sobre fuentes de datos tan diversas, por lo que se han originado técnicas propias de Minería de Datos para cada tipo de base de datos.

[10] Ver definición en Glosario G

A.4.6. Protección y Privacidad de los datos

Es importante poder determinar cuando el descubrimiento de información está en el límite de la invasión de privacidad y que medidas de seguridad pueden tomarse para evitar divulgar información confidencial o sensible. Los usuarios deben considerar el impacto de la Minería de Datos en términos legales, como la propiedad intelectual de los datos. La preocupación de los dueños de los datos por los datos en sí, se extenderá también a los resultados de la Minería de Datos. En esta área existe otra cara a tener en cuenta, y es que algunos de los requerimientos pueden generar objetivos conflictivos. El objetivo de preservar la privacidad de los datos va en contra de la Minería de Datos interactivo y en ocasiones también contra la utilización de diferentes fuentes de datos. En países donde el uso de Minería de Datos es fuerte se ha comenzado a estudiar el impacto legal de estos procesos.

A.5. Tipos de problemas

Los problemas de Minería de Datos pueden separarse en dos grandes grupos: predicción y descripción. Los problemas de predicción producen como resultado un modelo a partir de un sistema descrito por un conjunto de datos. El propósito de los problemas de descripción es generalmente obtener información de los datos analizados. Una buena descripción de los datos es fundamental para obtener una explicación de los mismos, así como también para plantear un problema de predicción.

Los problemas de predicción se describen con metas específicas, relacionando la respuesta que se desea obtener al análisis de muestras de datos previas. Los problemas descriptivos se encuentran en una etapa previa a la predicción, donde la información que se tiene de los datos es insuficiente para plantear una meta y por lo tanto es inviable realizar predicciones[7].

A.5.1. Predicción

En predicción un conjunto de datos históricos es examinado y el mismo se generaliza en un modelo que se aplicará a datos futuros. Los tipos más importantes de problemas de predicción son *clasificación* y *regresión* y existe además un caso especial conocido como *series de tiempo*[7].

Clasificación Descubrimiento de una función que permita clasificar datos en un grupo de clases predefinidas. En el caso de la clasificación, aplicar el modelo a los nuevos datos generará como respuesta "si" o "no"[7], (si, si pertenece a la clase y no en caso contrario).

Regresión Descubrimiento de una función que mapee cada caso de los datos a un número, este número es el valor de la variable a predecir. En el caso de la regresión, la respuesta obtenida al aplicar el modelo a cada caso de los nuevos datos es un número[7].

Series de tiempo Es una especialización de regresión o clasificación donde las medidas se toman a lo largo de un período de tiempo y generalmente en escalas de tiempo uniformes [7].

A.5.2. Descripción

Los problemas descriptivos pueden ser de muchos tipos: detección de desviaciones, segmentación, clustering, reglas de asociación, resumen y visualización.

Detección de desviaciones Detectar cambios significativos en los valores de los datos. Desde la prespectiva de la predicción las desviaciones están relacionadas a eventos de bajo predominio.

Segmentación Para simplificar los problemas es común que los datos se segmenten usando alguna de las dimensiones (columnas) de la base de datos.

Clustering Es una función descriptiva común en la cual se buscan identificar un conjunto de subgrupos o clusters de los datos analizados.

Reglas de asociación Descubrir reglas de asociación que describan dependencias interesantes entre las variables o los valores en un conjunto de datos.

Resumen Una de las mayores virtudes de la Minería de Datos es encontrar descripciones compactas de los datos. Esta técnica puede aplicarse tanto al conjunto completo de datos como a un subconjunto de los mismos.

Visualización Uno de las formas más interesantes para describir la información es la visualización, esta forma tiene gran potencial para el descubrimiento de información cuando la misma no se ha organizado de forma que pueda expresarse en forma de características.

A.6. El proceso de Minería de Datos

El proceso de Minería de Datos es un proceso iterativo, se trata de un ciclo de pasos que se repiten tantas veces como sea necesario para el investigador. En la práctica, las dos metas primarias de la Minería de Datos son la predicción o la descripción. La predicción usa variables o campos de los datos para predecir valores desconocidos del conjunto de datos o valores futuros de variables de interés. La descripción busca patrones que describan los datos para que los mismos puedan ser interpretados por los usuarios.

Dentro del proceso de Minería de Datos se pueden distinguir las siguientes tareas:

- Definición del problema
- Preparación de los datos
- Construcción de modelos
- Validación de los modelos
- Puesta en producción los modelos
- Administración de los meta datos

Las primeras 5 tareas se realizan iterativamente y su inicio depende de la ejecución de las tareas anteriores, en cambio la administración de metadatos se lleva a cabo durante todo el proceso. La administración de meta datos tiene como fin administrar la información relevante de cada una de las tareas como ser las transformación, reducción y limpieza de los datos, la construcción y validación de los modelos, la documentación de puesta en producción y los resultados obtenidos.

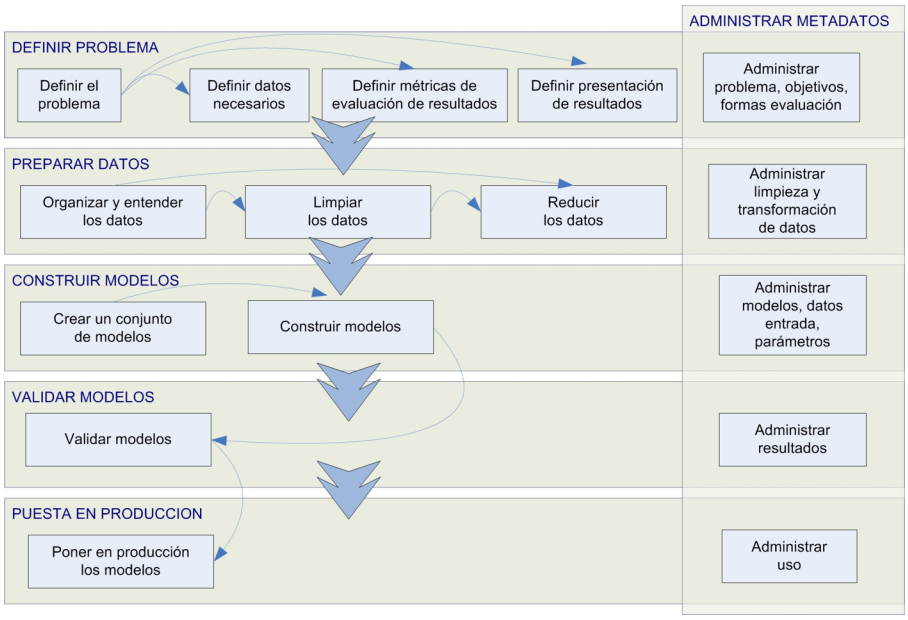

Figura A.1: Proceso de Minería de Datos.

El concepto más importante de la Minería de Datos es entender los datos. Si no se comprenden la estructura de los conjuntos de datos, no se sabrá qué información es posible obtener de ellos por lo que no será posible definir un problema y tampoco se sabrá que datos incluir en el modelo. Si se construye el modelo sin conocer los datos, se harán las preguntas erróneas y con esto se reduce o anula la efectividad del proceso de Minería de Datos. No es raro entonces que la preparación de los datos sea una de las etapas del proceso con mayor peso, tanto en cuanto a su importancia para obtener buenos resultados, como en el tiempo que suele consumir esta tarea.

A.6.1. Definición del problema

Antes de construir un modelo, es imprescindible entender los datos con los cuales se trabajará y que definen claramente el problema que se está intentando solucionar. Esto incluye analizar los requisitos, definir el alcance del problema, definir la o las métricas por la cual el modelo será evaluado, y definir el objetivo final para el proyecto de mineríade datos. Es posible que sea necesario llevar a cabo un estudio de la disponibilidad de datos e investigar las necesidades de los usuarios respecto a los datos disponibles. Si los datos no apoyan lo que los usuarios necesitan descubrir o verificar,

será necesario redefinir el proyecto.

Dentro de la tarea de definición del problema se identifican entonces las siguientes subtareas:

- Definir formalmente el problema, implica entender el problema planteado por el usuario y traducirlo al lenguaje al lenguaje de Minería de Datos.
- Definir los datos a usar, determinar que información se necesita para aplicar minería de datos el problema planteado.
- Definir las métricas con que se evaluarán los modelos y sus respectivos resultados, esto implica entender cual es el criterio con el que evaluará el usuario los resultados obtenidos.
- Definir como se presentaran los resultados, esto se debe definir dirigido a que es el usuario el que utilizará esta información.

A.6.2. Preparación de los datos

La preparación de los datos es, sin lugar a dudas, la fase crítica del proceso de Minería de Datos. En esta etapa se realizan varias tareas que tienen como finalidad lograr el mejor conjunto de datos posible para el problema. La mayoría del tiempo empleado en un proyecto de Minería de Datos es dedicado a preparar los datos, y tal preparación es el lugar en donde un usuario comienza el trabajo. La preparación de datos para Minería de Datos es un tema complejo y es de esperar invertir en esta tarea del 60 % al 90 % del tiempo total del proyecto.

Una tendencia popular en la industria de información es realizar la preparación de datos como paso de pre-procesamiento donde los datos resultantes se almacenan en un data warehouse [11]. La construcción se un data warehouse es una muy buena opción para organizar eficientemente los datos, pero no es imprescindible.

Organizar los datos

Reunir los datos puede ser una tarea compleja, es común que los datos se encuentren dispersos, almacenados en diferentes formatos y hasta en diferentes manejadores de bases de datos. En esta fase se hacen evidentes dos requerimientos que debe tener en Minería de Datos: manejar diferentes tipos de datos y diferentes fuentes de datos. Una vez que se han identificado las fuentes de datos a utilizar deben organizarse los datos en una forma estándar

[11] Ver definición en Glosario G

en la cual los programas de procesamiento puedan usarlos. Para organizar los datos basta con obtener un formato de planilla estándar compuesto de filas (casos) y columnas (dimensiones). La organización de los datos implica un entendimiento de los mismos, no basta con juntar todas las fuentes de datos disponibles, es necesario encontrarle un sentido a las mismas dentro del ámbito del proceso de Minería de Datos.

Reducir los datos

La reducción del volumen de datos se aplica generalmente a volúmenes grandes de datos Las ventajas técnicas de los volúmenes grandes de datos para el entrenamiento y prueba son obvias, pero en la práctica pueden ser demasiado grandes. Las dimensiones pueden exceder la capacidad del sistema o puede tomar mucho tiempo realizar el procesamiento y obtener la solución. La reducción de datos tiene como objetivo generar un conjunto de datos manejable por los algoritmos de Minería de Datos y mejorar los tiempos de procesamiento de los mismos. Una vez que los datos se encuentran en una forma estándar, existen técnicas efectivas para reducir sus dimensiones. El tema central de la reducción de datos es descartar parte de los datos sin sacrificar la calidad de los resultados. Se debe tener presente en todo momento que uno de los objetivos de la reducción de datos es la mejora de los tiempos de procesamiento, por lo tanto si el proceso de reducción de datos lleva más tiempo que ejecutar con el conjunto completo de datos, la reducción no vale la pena.

Las técnicas de reducción se pueden dividir en dos clases: las que preservar las características originales de los datos y las que no. Dentro de la primera clase se encuentran las operaciones de eliminación de casos, eliminación de dimensiones, reducción del número de valores de cada dimensión (suavizado de los datos). De los tres tipos la eliminación de dimensiones es la que lograr una reducción más rápida del volumen de datos y tiene la virtud de conservar los valores originales de las dimensiones restantes. Para efectuar reducción de dimensiones sin impactar negativamente en los resultados, se deben examinar cuidadosamente las dimensiones considerando su potencial. Algunas serán descartadas por ser redundantes y otras por aportar poca información. Las técnicas del segundo grupo sustituyen las dimensiones originales por nuevas dimensiones creadas por composición de las dimensiones originales. Estas técnicas generan datos que son irreconocibles comparados con los originales y es por lo tanto menos usado.

De las técnicas de reducción utilizadas, los procedimientos de discretización son los más comunes. Cuando los datos son continuos, en ocasiones es

necesario individualizarlos para que sea posible usar algunos algoritmos de minería de datos que no funcionan con datos continuos. Los procedimientos de discretización, es decir el proceso de repartir valores continuos de una variable en un sistema de intervalos, pueden tener una influencia importante en la calidad de las reglas que se generan.

Limpiar los datos

Los datos incompletos, con ruido e inconsistentes son características comunes de las bases de datos y de almacenes de datos. Esto se acentúa cuando las mismas son grandes, que como ya hemos visto es una característica común en las fuentes de datos de Minería de Datos. Los datos incompletos pueden ocurrir numerosas razones: pueden no estar disponibles, pueden no haber sido considerados importantes a la hora de ingresar los casos, puede deberse a un malentendido o a funcionamiento incorrecto del equipo o de la base de datos. Los datos inconsistentes en general son el resultado de eliminación de datos registrados o errores en la generación de la historia en la que se paso por alto algún caso. En ocasiones es necesario deducir los datos faltantes, en especial cuando se trata de registros con valores faltantes. Esta realidad hace necesaria otra tarea previa a a Minería de Datos, la limpieza de datos. La tarea de limpiar los datos incluye completar los datos faltantes, suavizar el ruido y resolver inconsistencias.

Para realizar la limpieza de los datos es un requisito fundamental conocerlos. Las decisiones tomadas en esta etapa tendrán consecuencias en la validez y precisión del modelo. Es necesario saber cual es el objetivo de la Minería de Datos, como están dispuestas las dimensiones y cual es su formato. Es con este conocimiento que se debe enfrentar los datos para eliminar inconsistencias antes de construir el modelo. La forma de la cual resolvemos estos problemas depende de la situación, de los requisitos del modelo, y de la manera elegida para encarar el problema.

Algunos problemas comunes en la limpieza de datos son: columnas que contienen gran cantidad de valores nulos, columnas con enorme variedad de valores (por ejemplo, número telefónico), valores incoherentes con el rango de valores válidos de la columna (por ejemplo, valor X para el sexo de una persona que puede ser Masculino o Femenino), valores que no cumplen con el formato estipulado para la columna (por ejemplo, un valor de fecha 01/15/2005 para un formato de fecha dd/mm/aaaa), grupos de columnas dentro de un registro que no tienen sentido cuando se comparan entre sí (por ejemplo, fecha de nacimiento 01/08/1990 y de casamiento de una persona 31/07/2000).

A.6.3. Construcción del modelo

Un modelo es una representación abstracta, en la mayoría de los casos aproximada, de un proceso de la vida real. La clasificación más simple de modelos es la que los divide en fijos, paramétricos y no paramétricos [8].

Modelos fijos, es el tipo más simple, formula una ecuación que define como las salidas se derivan de las entradas. Estas ecuaciones son luego traducidas a un programa. El reconocimiento de patrones no es en general tan sencillo, el desconocimiento del procedimiento impide la formulación de modelos de este tipo, por lo tanto los modelos fijos son los menos usados y están restringidos a problemas sencillos.

Modelos paramétricos, se formulan ecuaciones matemáticas que caracterizan las relaciones entre las entradas y las salidas. La diferencia con los modelos fijos es que la información en los modelos paramétricos es parcial, es decir que las relaciones para algunas parámetros no se conocen. La especificación de estos parámetros puedes ser mejorada analizado los datos empíricamente.

Modelos no paramétricos, están fuertemente relacionados al uso de los datos. La hipótesis sobre la cual trabajan es que las relaciones que ocurren en el conjunto de datos observado, ocurrirán en el futuro. Los modelos fijos y paramétricos están limitados por el entendimiento humano, por lo tanto se restringuen a realidades más simples. Los modelos no paramétricos producen modelos de alta complejidad a partir de los datos y no requieren un entendimiento completo del problema.

La consideración más importante al construir un modelo de Minería de Datos es recordar que el proceso de Minería de Datos es un proceso iterativo. Será necesario probar varios modelos para encontrar el más útil para el problema planteado. En el proceso de búsqueda de ese modelo puede ser necesario retroceder a pasos anteriores, ya sea para modificar los datos o redefinir el problema. El modelo óptimo es aquel que produce los resultados más acertados. Cuando hay incertidumbre, la existencia de errores no significa que exista un modelo mejor. En casos de existencia de incertidumbre, hasta el mejor modelo producirá errores y el mejor es aquel que los minimiza.

La construcción de modelos para Minería de Datos es guiada por los datos y el tipo de modelo generado dependerá del tipo de minería de datos prevista. La Minería de Datos descriptiva produce un modelo del sistema

descrito por el conjunto de datos dados. La Minería de Datos de predicción produce un modelo de información nueva y no trivial basándose en las variables disponibles en los datos. El proceso de construir modelos de predicción requiere separar en forma aleatoria el conjunto de datos original en por lo menos dos grupos: datos de entrenamiento y datos de prueba. Se utilizará el conjunto de datos de entrenamiento para construir el modelo. Después de que se genere el modelo usando la base de datos del entrenamiento, se utiliza para predecir la base de datos la prueba, y la exactitud que resulta es una buena estimación de cómo el modelo se comportará con datos futuros. No garantiza que el modelo está correcto, valida simplemente que si la misma técnica fuera utilizada en una sucesión de bases de datos con datos similares a los datos del entrenamiento y de prueba, la exactitud media estaría cerca de la obtuvo esta manera. Si no se utiliza esta división de datos de entrenamiento y de prueba, la exactitud del modelo será sobrestimada. La validación del modelo con los datos de prueba para asegurar las predicciones más exactas y más robustas. Si el conjunto de datos de entrenamiento no es representativo del conjunto de datos total entonces la utilidad del modelo se verá comprometida. Este método de construcción se conoce como aprendizaje supervisado, consiste en entrenar el modelo en una porción de datos y después validarlo en el resto de los datos. A veces un tercer conjunto de datos, llamado datos de validación, este tercer conjunto es necesario porque los datos de prueba pueden influenciar las características del modelo, y la validación actúa como medida independiente de la exactitud del modelo.

A.6.4. Validación del modelo

Validar el modelo es verificar que el mismo cumple los objetivos de precisión definidos por los usuarios. Luego de construido el modelo, se prueba su exactitud creando escenarios de predicción contra el conjunto de datos de prueba. El usuario conoce el resultado de las predicciones, porque los datos provienen del mismo conjunto usado para entrenar al modelo, entonces puede calcular la exactitud de funcionamiento del modelo. La validación no garantiza que el modelo sea correcto, sino que da una estimación del desempeño que el mismo tendrá en conjuntos de datos futuros.

"La validación del modelo es una condición necesaria pero insuficiente para la credibilidad y aceptación de los resultados de la Minería de Datos." [13]

Es común que se construyan varios modelos y se comparen los resultados que cada uno produce con el conjunto de datos de prueba para elegir el que

produzca los mejores resultados.

Tipos de validación:

Validación simple. Se divide el conjunto de datos al azar en dos partes; una de ellas es usada en la construcción del modelo y la otra se reserva para validarlo. Usualmente la porción reservada es menor, la proporción varía según el tamaño de los datos origen y puede llegar hasta un tercio del mismo. La división de los datos debe realizarse en forma aleatoria para garantizar que ambos grupos de datos contengan las características de los datos modelados.

Validación cruzada. Si se dispone de un conjunto modesto de datos, no es posible prescindir de una parte de ellos para validar el modelo. La validación cruzada divide el conjunto de datos en N subconjuntos disjuntos de tamaño similar y se construyen N+1 modelos, uno con cada subconjunto y otro con el conjunto completo de datos. Se reserva uno de los subconjuntos y se construye el modelo con los otros N-1 subconjuntos. El modelo resultante se usa para predecir el comportamiento del subconjunto reservado y calcular el error. Luego de aplicado el mismo procedimiento para los N subconjuntos, la media de los errores obtenidos da una estimación del error del modelo que se construyó con el total de los datos.

Bootstrapping. Bootstrapping es un método estadístico para estimar la distribución del muestreo, fue creado por Efron (1979-1981) y luego desarrollado por Efron y Tibshirani (1993). Esta técnica se usa para estimar el error del modelo y es más comúnmente usada en conjuntos de datos pequeños. El modelo se construye usando todos los datos disponibles. Luego se construyen subconjuntos de datos - llamados bootstrap -, mediante muestreo de los datos originales. En los métodos anteriores las instancias de datos no se repetían, una vez seleccionada, no podía seleccionarse nuevamente. La mayoría de las técnicas de aprendizaje pueden utilizar la misma instancia más de una vez. La idea del bootstrap es tomar muestras del conjunto de datos con repeticiones para formar un conjunto de datos de entrenamiento. Para ello, un conjunto de N inst ancias se muestrea N veces, con reemplazo, y se obtiene otro conjunto de datos de N instancias. Como algunas instancias del segundo conjunto estarán repetidas, deben existir algunas instancias del conjunto original que no fueron seleccionadas y se utilizaran estas instancias para construir el conjunto de datos de prueba.

A.6.5. Puesta en producción de los modelos

La puesta en producción de los modelos no es el paso final del proceso, el proceso de Minería de Datos es un proceso iterativo que aprende de los resultados. La constante actualización de los modelos es parte de la estrategia de puesta en producción. A medida que la cantidad de datos recolectados aumenta es necesario reconstruir los modelos en base a ellos y mejorar su efectividad.

A.6.6. Administración de los meta datos

La información relacionada con la exploración de los datos, la limpieza de los datos, la construcción de los modelos es importante y debe ser guardada. La gestión de estos datos puede convertirse en un proyecto por si mismo y es una etapa realmente importante del proceso. La forma usual y más segura de almacenamiento es en bases de datos. Se debe incluir en ella los datos eliminados por el proceso de limpieza, todos los modelos evaluados y los resultados obtenidos con los mismos. El meta dato más importantes es el modelo, se debe registrar una historia detallada de cómo se llegó a su definición y la evaluación de su funcionamiento. Registrar la historia de los modelos previos al modelo definitivo hace posible reconstruirlo en caso de que sea necesario y al mismo tiempo evitar repetir errores ya resueltos. También es importante esta historia para representar el valor del modelo obtenido. Los datos básicos que se deben documentar son: nombre del modelo, versión del modelo, fecha de creación, conjuntos de datos de entrenamiento de prueba, algoritmos, parámetros de los algoritmos y resultados.

"En el mejor de todos los mundos, el modelo de Minería de Datos datos final se debe documentar con una historia detallada. Un minero de datos profesional deseará saber exactamente como un modelo fue creado, para explicar su valor, evitar repetir errores y reconstruirlo en caso de necesidad." [14]

Según cada autor, varían la definición los pasos del proceso de Minería de Datos, el orden de los pasos y las tareas que se llevan a cabo en cada paso. Uno de los esquemas es el propuesto por Berry y Linoff [15].

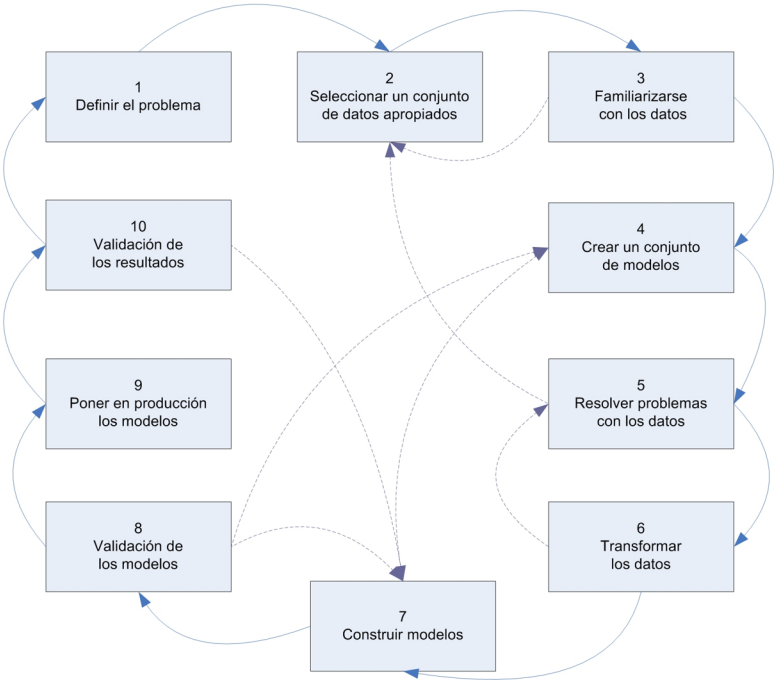

Figura A.2: Proceso de Minería de Datos según Berry y Linoff.

A.6.7. Problemas a tener en cuenta

Determinar el volumen adecuado de datos

El procesamiento de cantidades masivas de datos es una característica fundamental de la Minería de Datos. Cantidades grandes de datos, tienen mayor potencial para generar información valiosa. Si la Minería de Datos es una búsqueda a través de un espacio de posibilidades, entonces los grandes cantidades de datos sugieren muchas más posibilidades para enumerar y para evaluar. Este potencial para enumeración y búsqueda tiene limitaciones prácticas. Es fácil demostrar las dificultades del manejo de grandes volúmenes de datos, aún con enorme cantidad de recursos computacionales. Además de la complejidad de cómputo de los algoritmos de Minería de Datos que trabajan con los datos, una búsqueda más exhaustiva puede también aumentar el riesgo de encontrar algunas soluciones de baja probabilidad que evalúen bien para el conjunto de datos dado, pero pueden no resolver las expectativas futuras. Mientras que los grandes volúmenes de datos tienen

mayor potencial para obtener mejores resultados, no hay garantía de que sean mejores que volúmenes pequeños. No es tan importante el volumen de los datos por su tamaño en sí, sino por lo que representa en conocimiento. Un gran volumen de datos generados en forma aleatoria por la computadora es, desde la perspectiva del conocimiento, un conjunto inútil.

Para evaluar la calidad de los datos se hace necesario tener en cuenta su origen. En general las bases de datos y almacenes de datos contienen datos objetivos que representan transacciones reales. Por otra parte existen bases de datos que resultan de encuestas de opinión, estos datos son poco confiables por innumerables razones, desde el planteamiento de las preguntas, el muestreo no uniforme de individuos a la decisión de las personas de participar o no en las mismas. Los datos reunidos sin intervención humana son usualmente más objetivos.

Para tomar una decisión respecto al volumen de datos a utilizar, lo primero que se debe tener en cuenta es la relación entre el tamaño del sistema y su densidad. La densidad refiere al predominio de resultados de interés. Una muestra más pequeña y equilibrada es preferible a una más grande con una proporción muy baja de resultados raros. Lo segundo a tener en cuenta es el tiempo de procesamiento; cuando el sistema es grande, construir modelos buenos y estables requiere de balance. La creación de un modelo con la muestra de datos más grande es ineficaz porque llevará demasiado tiempo de procesamiento. Luego de haber elegido una muestra de datos que se considere razonable para modelar, se puede realizar una prueba simple para saber si la muestra usada es suficientemente grande. La prueba consiste en duplicar su tamaño, medir la mejora en la exactitud del modelo. Si el modelo creado usando la muestra más grande es perceptiblemente mejor que el creado usando la muestra más pequeña entonces la muestra más pequeña no es suficientemente grande. Si no hay mejora, o solamente una mejora leve, entonces la muestra original tiene el tamaño adecuado.

Sobreentrenamiento

Cuando un modelo es generado a partir de datos de una base de datos, es de esperar que el modelo se ajuste a futuros estados de la base de datos. En ciertos casos ocurre overfitting [12], en esta situación el modelo queda demasiado ligado a los datos de entrenamiento y será menos útil para datos futuros de la base de datos. El sobreentrenamiento puede deberse a suposiciones realizadas acerca de los datos o debido a un tamaño de datos de

[12] Overfitting; en español sobreentrenamiento

entrenamiento demasiado pequeño. En este último caso no se logran identificar las tendencias como era el objetivo del entrenamiento del modelo sino que se refleja en el los casos particulares contenidos en el conjunto de datos de entrenamiento

A.7. Metodologías para Minería de Datos

El proceso de Minería de Datos debe ser confiable y repetible, en lo posible para usuarios con poco conocimiento de Minería de Datos. Cuando el uso de técnicas de Minería de Datos se hizo popular, se hizo evidente la necesidad de contar con metodologías que estandarizaran el proceso de Minería de Datos. Esta necesidad tuvo dos orígenes, la del usuario y la de los investigadores. Para el usuario inexperto proporcionan seguridad de que el proceso que está llevando a cabo cumple con estas características deseables de un proceso de Minería de Datos. Para los investigadores contar con modelos de Minería de Datos refleja la madurez del área.

El desarrollo de modelos bien documentados, no propietarios y gratuitos ha sido de gran utilidad para afianzar y extender la aplicación de Minería de Datos. Los modelos sirven de guía a los usuarios que enfrentan un proyecto de Minería de Datos y plantean buenas prácticas a seguir que facilitan su aplicación a usuarios con poca experiencia. La existencia de estos modelos no hace que sea menos común el uso de modelos propios de cada empresa o investigador. En general, los usuarios expertos desarrollan sus propios modelos.

Las metodologías existentes son similares, difieren en la cantidad de pasos que dan al proceso, la repetición o no de algunas tareas e incluso en el orden de las mismas, pero básicamente hacen lo mismo. De los modelos disponibles los más conocidos y usados son CRISP-DM[16] y SEMMA[17]. Debido a que el modelo CRISP-DM es el adoptado por Clementine®, uno de los productos líderes del mercado en Minería de Datos, se ha convertido en el modelo más usado. SEMMA por su parte se ha incorporado a otro producto muy popular SAS®Enterprise Miner™.

A.7.1. Modelo CRISP-DM

El modelo CRISP-DM [16] surgió en 1996 como el resultado del trabajo conjunto de los que en ese momento eran líderes del mercado de Minería de

Datos: Daimler-Benz [13] (ahora DaimlerChrysler), Integral Solutions Ltd. [14] (adquirida por SPSS en 1998), NCR, y OHRA [15]. La idea original era que CRISP-DM se convirtiera en una herramienta comercial, pero en 1997 se creo el grupo SIG (Special Interest Group) con el objetivo de desarrollar un modelo de proceso estándar para la comunidad de Minería de Datos. Este cambio hizo que el modelo comenzó a recibir aportes de otras empresas e interesados. El modelo CRISP-DM se ha desarrollado y evolucionado desde entonces y así también ha sucedido con el grupo SIG que hoy en día cuenta con más de 200 integrantes.

El modelo CRISP-DM para Minería de Datos organiza el proceso de Minería de Datos en un modelo jerárquico de 4 niveles. El primer nivel se denomina fase y en él se identifican 6 fases: entender el negocio, entender los datos, preparar los datos, modelar, evaluar y poner en producción. El segundo nivel se denomina tarea genérica porque intenta ser completa, es decir cubrir todas las situaciones que se pueden dar en un proceso de minería de datos y también estable para soportar nuevas técnicas. El tercer nivel se denomina tarea especializada y describe como se deben llevar a cabo las tareas genéricas del nivel 3 en forma de pasos a seguir en un orden específico. En la práctica es posible realizar las tareas en otro orden e incluso repetir las tareas. El cuarto nivel se denomina instancia y en él se registran las acciones, decisiones y resultados que componen las tareas definidas en los niveles superiores.

[13] Daimler-Benz fue una de las primeras empresas comerciales en aplicar minería de datos, aportó sus proyectos para validar el modelo.

[14] ISL desarrolló y lanzó al mercado el primer paquete comercial de Minería de Datos: Clementine.

[15] OHRA es una importante compañía de seguros alemana, aportó sus proyectos para validar el modelo.

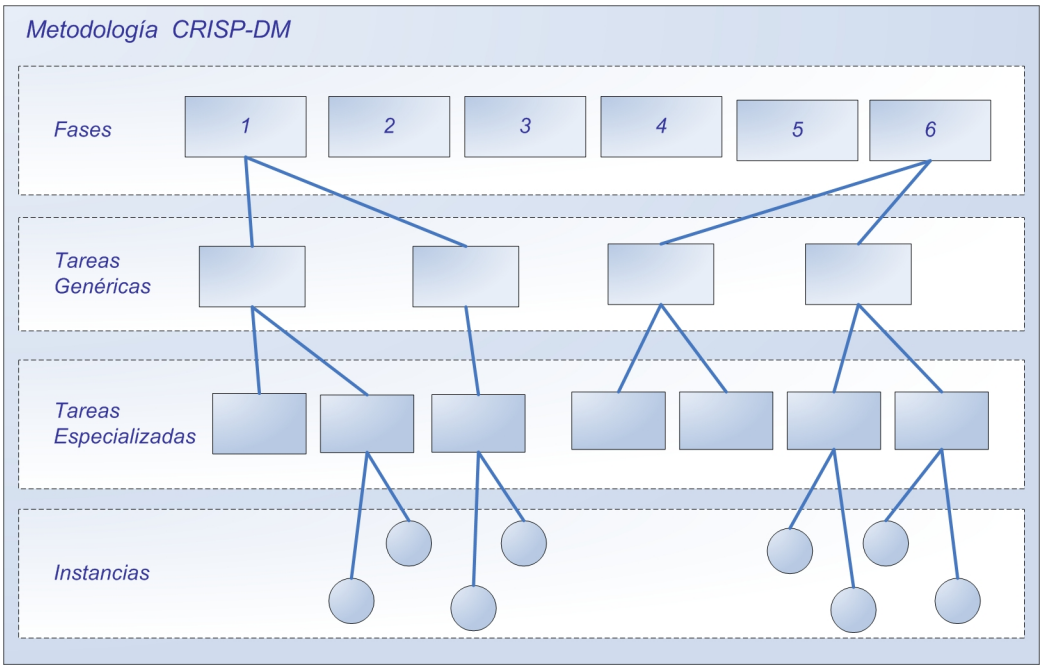

Figura A.3: Metodología CRISP-DM.

Fase 1:

- Entender el negocio
- Determinar los objetivos de negocio
- Determinar los objetivos de la Minería de Datos
- Establecer los criterios de éxito

Fase 2:

- Entender los datos
- Recolectar los datos
- Describir los datos
- Explorar los datos y verificar su calidad

Fase 3:

- Preparar los datos
- Describir los conjuntos de datos
- Seleccionar los datos

- Limpiar los datos
- Transformar los datos (derivación de datos)
- Integrar los datos
- Formatear los datos

Fase 4:

- Modelar
- Seleccionar las técnicas de modelado basándose en los objetivos
- Generar pruebas para los modelos
- Construir los modelos
- Revisar los modelos y determinar los parámetros de cada uno

Fase 5:

- Evaluar
- Evaluar los modelos con las pruebas previstas
- Interpretar los resultados
- Evaluar resultados de la Minería de Datos con los criterios de éxito definidos en la fase 1
- Revisar el proceso

Fase 6:

- Puesta en producción
- Definir el plan de puesta en producción: con que frecuencia se usará, como se usarán los resultados, quienes los usarán.
- Monitorear y mantener el plan
- Generar el reporte final
- Revisión final del proyecto
- Documentar la experiencia

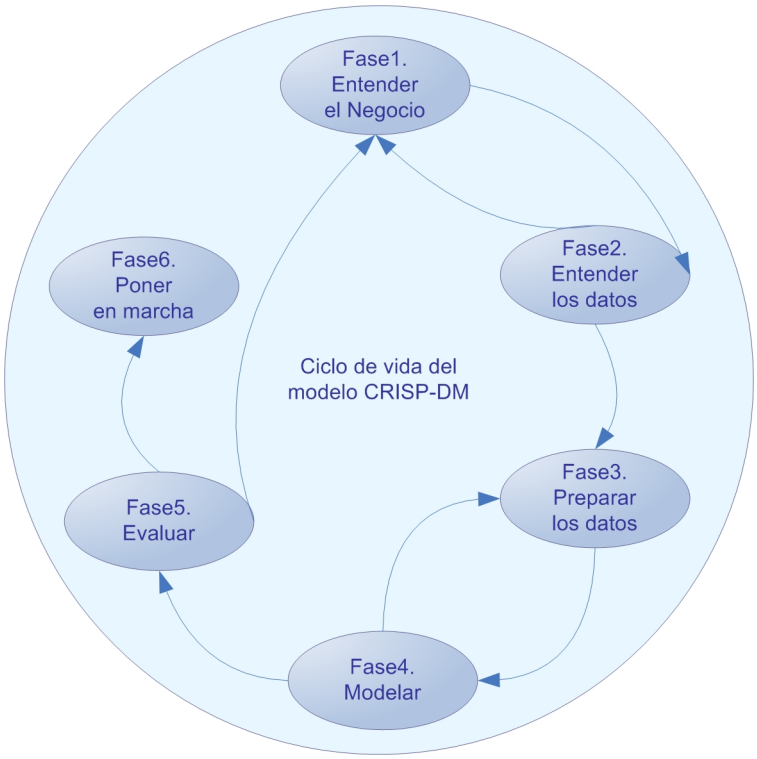

Figura A.4: Ciclo de vida del modelo CRISP-DM.

A.7.2. Modelo SEMMA

SEMMA [17], es un nombre compuesto por las siglas en inglés de las fases del proceso que propone Simple (muestreo), Explore (explorar), Modify (modificar), Model (modelar) y Assess (evaluar).

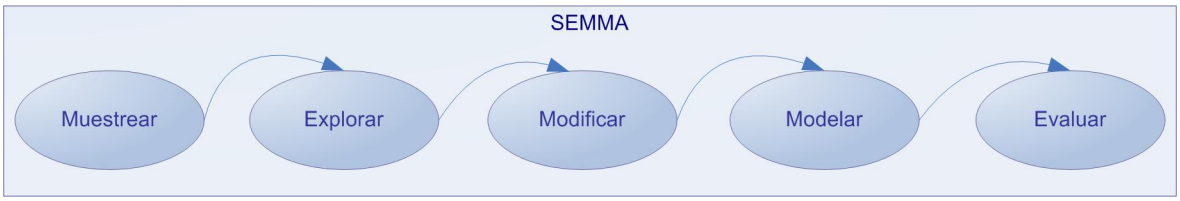

Figura A.5: Ciclo de vida del modelo SEMMA.

Muestrear los datos, es un paso opcional que consiste en extraer una pequeña porción de un conjunto grande de datos que contenga información significativa. La reducción del tamaño de la muestra facilita su manipulación y tiempo de respuesta. Aplicar Minería de Datos a una muestra representativa, en vez del volumen entero, reduce el tiempo de transformación requerido para conseguir la información crucial del negocio. Si los patrones generales aparecen en los datos en su totalidad, éstos serán detectables en una muestra representativa. Si existen datos que no están representados en una muestra pero son importante en el conjunto de datos original, estos podrán ser descubiertos usando métodos sumarios.

Explorar los datos, buscando tendencias desconocidas y anomalías no esperadas de forma de entender mejor los datos. La exploración permite refinar el proceso del descubrimiento. La exploración puede ser visual, usando técnicas estadísticas o clustering.

Modificar los datos seleccionando y transformando los mismos variables para enfocar el proceso de selección modelo. La Minería de Datos es un proceso dinámico e iterativo, por lo tanto esta etapa comprende la actualización de datos y modelos. De acuerdo con la fase de la exploración, puede ser necesario manipular los datos para incluir información faltante, aumentar o reducir el número de variables. Esta etapa cubre también la actualización de datos cuando los mismos cambian en las fuentes de datos de origen.

Modelar los datos permitiendo que el software busque automáticamente un resultado para una combinación de datos. Cada tipo de modelo según el tipo de algoritmo elegido tiene ventajas, desventajas y particulares que lo hacen más o menos apropiado dependiendo de los datos.

Evaluar la utilidad y la confiabilidad de los resultados obtenidos y estimar su precisión. La forma más común de efectuar esta tarea es aplicar el modelo

a una porción de datos separados de la muestra original antes en la etapa de muestreo. Si el modelo es válido, debe funcionar para esta muestra como para la muestra usada para construir el modelo. Es posible también validar el modelo con datos cuyos resultados son conocidos.

A.8. Proyectos de Minería de Datos

En el momento de encarar un proyecto de Minería de Datos hay que tener en cuenta la factibilidad económica, organizativa y técnica del mismo. La factibilidad económica implica que empresa pueda afrontar los costos de del proyecto. Factibilidad organizativa pues el proyecto causará un impacto significativo en la empresa y se deberán prever cuales serán y evitar los negativos. Las situaciones negativas pueden darse por no se conocer los métodos, no contar con personal calificado o por problemas legales como la violación de privacidad de la información. La factibilidad técnica se comprueba si se dispone de suficientes datos, los datos contienen información relevante y representativa de la empresa, existe poco ruido en los mismos y el personal domina la aplicación de los métodos de Minería de Datos a aplicar.

Algunos de los factores que pueden hacer fracasar un proyecto de Minería de Datos son:

- La necesidad de tener experiencia para utilizar herramientas.
- La obtención de resultados que contienen patrones erróneos, triviales o no interesantes.
- Proyectos demasiado largos que no posibilitan al usuario tener resultados en un espacio de tiempo considerable.
- La elección equivocada de la herramienta o los métodos.
- Razones legales u organizativas que impidan la utilización de la información para la aplicación de los métodos.

A.8.1. Costos de un proyecto de Minería de Datos

Un proyecto de Minería de Datos, y más concretamente un proceso de Minería de Datos, es eficiente si logran obtener información cuyo valor supere el costo correspondiente a su ejecución. Llevar a cabo un proyecto de Minería de Datos es realizar una inversión, y como toda inversión, su eficiencia se mide en la relación costo-beneficio. El costo del proyecto está compuesto por el costo del software, el costo del personal asignado al proyecto y el costo de mantener el sistema de Minería de Datos una vez terminado el proyecto.

El costo del software puede ser desequilibrante al momento de tomar esta decisión ya que el personal necesario dependerá de esta elección.

Las tecnologías de Minería de Datos están en apogeo, las posibilidades de aplicaciones son innumerables y crecientes. Existen una serie de elementos hacen a la Minería de Datos aplicable y necesaria, pero por otra parte existe una serie de obstáculos para que sea posible su difusión a las áreas que la necesitan. Uno de los obstáculos es que los productos de software comercializados son costosos, por tanto los consumidores pueden hallar una relación costo - beneficio negativa.

Hay un peligro considerable de subestimar los costos de poner en producción, usar y mantener un proyecto de Minería de Datos en una empresa. Como ocurre a menudo con otras actividades relacionadas a sistemas de información, los costos son subestimados a menudo por 50 % o más (información extraída de Gartner Group). Una fuente de costos frecuentemente pasada por alto es el trabajo de uso y mantenimiento del sistema. Los vendedores del software de minería de datos tienen la tendencia a hacer recomendaciones de contratación de pocos analistas con poca experiencia para la implantación de sistemas. El personal con poca experiencia no es el adecuado para satisfacer las expectativas de la mayoría de las empresas. Con respecto al mantenimiento, las empresas deben hacer frente no sólo a las actividades de mantenimiento asociadas al sistema de Minería de Datos, sino también debe asegurar recursos para mantener los modelos y datos.

Apéndice B

Estudio de Algoritmos de Minería de Datos

B.1. Introducción

"Un algoritmo [1] *de Minería de Datos es un procedimiento bien definido* [2] *que toma los datos como entrada y produce una salida en forma de modelos o patrones." [4].*

El algoritmo es el corazón del proceso de Minería de Datos. Elegir el algoritmo correcto a aplicar en un problema es una tarea compleja. Cada algoritmo produce un resultado diferente, y algunos hasta más de un tipo de resultado; es necesario evaluar cual es el adecuado al problema y principalmente a los datos. Es posible también utilizar diversos algoritmos para solucionar un problema, combinándolos para aprovechar las mejores características de cada uno. Algunos algoritmos se utilizan como medios para explorar los datos, para luego aplicar otros que pueden predecir un resultado específico basado en los datos. Además, es importante tener presente mientras se elige un algoritmo apropiado, que la meta real es crear un modelo robusto y exacto, que pueda ser entendido por los usuarios y puesto en producción con esfuerzo mínimo.

[1] Para ser considerado un algoritmo el procedimiento debe terminar luego de un número finito de paso, un método computacional no asegura esta condición.

[2] Bien definido refiere a que el procedimiento debe estar codificado en forma precisa como un conjunto finito de reglas.

B.2. Componentes de un algoritmo

Hand, Mannila y Smyth [4] identifican los siguientes componentes en un algoritmo de Minería de Datos: tareas, estructura del modelos, funciones de evaluación, métodos de búsqueda u optimización y técnicas de gestión de datos. Esta descomposición de los algoritmos facilita su comparación y también hace más sencilla la tarea de elegir un algoritmo para un caso particular. La idea es identificar los componentes que se adecuan al problema y adoptar los algoritmos que los contienen, en vez de comparar algoritmos. Hoy en día esta posibilidad no está disponible en los productos que se comercializan, donde el usuario debe elegir el algoritmo a utilizar. Viéndolo desde el punto de vista del usuario inexperto es una ventaja, o más bien una simplificación. Para un usuario que cuente con los conocimientos necesarios para realizar este análisis, representa una limitación, que le impide identificar el algoritmo que mejor se ajusta a su problema.

Tarea, el tipo de tarea de Minería de Datos que trata el algoritmo. Los tipos de tareas corresponden a diferentes objetivos que se plantea el usuario que analiza los datos. Una de las posibles clasificaciones de tareas es la siguiente:

- **Exploración,** consiste en explorar los datos sin ninguna idea u objetivo claro de lo que se está buscando. Las técnicas de exploración son en general interactivas y visuales. A medida que el volumen de datos aumente, se hace cada vez más difícil la interpretación de los resultados, aún cuando se cuente con apoyo visual.

- **Descripción,** la meta es lograr la descripción del conjunto de datos o del proceso que los genera. Dentro de esta categoría se encuentra la estimación de densidad, clustering , segmentación y análisis de dependencias.

- **Predicción,** el objetivo es construir un modelo que permita predecir, a partir de valores conocidos, los valores de otras variables. La clave para distinguir predicción de descripción es que la predicción tiene como objetivo una única variable. Dentro de esta categoría se identifican dos clases, la clasificación y la regresión. En clasificación, la variable a predecir es categórica y en regresión, la variable a predecir es cuantitativa.

- **Descubrimiento de patrones y reglas,** detección de patrones en el conjunto de datos.
- **Recuperación a demanda,** el usuario ha identificado un patrón de interés y desea encontrar patrones similares en los datos.

Estructura del modelo, determina la estructura o forma funcional que buscamos en los datos.

Función de evaluación, su cometido es evaluar los modelos en función de su utilidad para el constructor de los modelos. La medición de utilidad en términos prácticos es un proceso complejo y seguramente inexacto, por lo tanto, la función de evaluación debe considerar la mayor cantidad de características importantes de la tarea de Minería de Datos. Es necesario evitar el uso de funciones de evaluación disponibles sólo por su disponibilidad, las diferentes funciones tienen diferentes propiedades y son útiles en determinadas situaciones. Una función de evaluación puede ser definida en orden de minimizar un error o una medida a maximizar. Para los modelos de predicción resulta más intuitiva la definición de funciones de evaluación, en cambio para modelos descriptivos, donde no existe una variable objetivo a predecir, resulta sensiblemente confuso.

Métodos de búsqueda u optimización, búsqueda de diferentes modelos y patrones, y optimización de las funciones de evaluación. La tarea de encontrar los mejores parámetros para un modelo es un problema de optimización o estimación. La tarea de encontrar patrones interesantes en un conjunto extenso de patrones es una tarea de búsqueda. Se utilizan para determinar la estructura y los valores para los parámetros con los cuales se maximiza o minimiza, según el caso, la función de evaluación.

Estrategia de gestión de datos, métodos utilizados para el manejo eficiente de los datos durante el proceso de búsqueda u optimización. La mayoría de los algoritmos trabajan con datos en memoria (RAM), pero existen también algoritmos que permiten trabajar con datos en almacenamiento secundario (disco) o en almacenamiento terciaria (por ejemplo cinta). La enorme mayoría de los algoritmos han sido diseñados sin especificaciones respecto a la estrategia de gestión de datos que utilizan.

Algoritmo	CART	Propagación hacia atrás	A Priori
Tarea	Clasificación y Regresión	Regresión	Descubrimiento de patrones
Estructura	Arboles de decisión	Redes neuronales	Reglas de asociación
Función de evaluación	Validación cruzada	Error al cuadrado	
Método de búsqueda	Greedy	Gradiente de la pendiente para los parámetros	Breath-first con poda
Gestión de datos	No especificado	No especificado	Búsqueda lineal

Tabla B.1: Ejemplo de componentes de algoritmos conocidos.

"La base algorítmica de un algoritmo de Minería de Datos recae en los métodos computacionales usados para instrumentar la búsqueda y la gestión de componentes." [4]

Uno de los aportes más interesantes de esta separación en componentes es que permite observar los diversos énfasis puestos en aspectos algorítmicos de minería de datos en las diferentes comunidades de investigación. En el ámbito estadístico figuran ecuaciones para especificar modelos, funciones de evaluación y métodos de cómputo; pero relativamente pocas o ninguna especificación algorítmica de cómo los modelos se construyen en la práctica. En contraposición, en el ámbito informático se acentúa los métodos y los algoritmos de cómputo, con poco énfasis en la estructura del modelo o la función de evaluación que debe ser utilizada. En el contexto de la Minería de Datos, los diversos énfasis en las dos áreas han conducido al desarrollo de metodologías diferentes y a menudo complementarias. Para hacer un análisis profundo de los algoritmos es necesario recurrir a ambas ramas, si nos limitamos a una de ellas el análisis de los algoritmos será incompleto.

B.3. Clasificación y tipos de algoritmos

Los algoritmos pueden clasificarse en 3 categorías [7]:

- Algoritmos matemáticos
- Algoritmos lógicos

- Algoritmos de distancias

Tanto los algoritmos matemáticos como los lógicos describen soluciones como operación directa sobre nuevos datos. Los algoritmos matemáticos combinan valores con operaciones matemáticas. Los algoritmos lógicos por su parte usan operadores lógicos o de comparación para evalúan expresiones en uno de dos valores: verdadero o falso. Los algoritmos de distancia en cambio, obtienen respuestas para un nuevo caso midiendo similitudes con otros casos almacenados previamente.

Algunos ejemplos de cada tipo

- Algoritmos matemáticos
  - Regresión lineal
  - Redes neuronales
- Algoritmos lógicos
  - Arboles de decisión
  - Reglas de decisión
- Algoritmos de distancias
  - K-medias
  - K vecinos más cercanos

B.4. Métodos supervisados y métodos no supervisados

Los métodos de Minería de Datos pueden dividirse en dos categorías, la de aprendizaje supervisado y aprendizaje no supervisado. Estos conceptos provienen del área de aprendizaje de máquinaA.2.4. Los métodos no supervisados no plantean una variable objetivo, el algoritmo busca libremente patrones y estructuras en los datos. Los métodos supervisados establecen una variable objetivo y se proporcionan al algoritmo ejemplos en los cuales la solución es conocida para que el algoritmo aprenda [18].

La mayoría de los métodos supervisados de Minería de Datos aplican la metodología esquematizada en la figura B.1 para construir el modelo. El

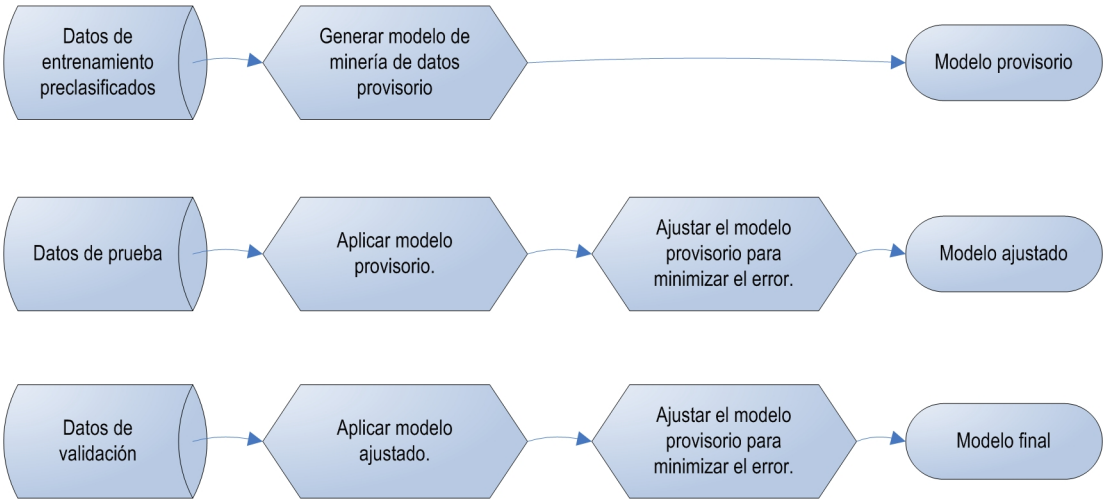

Figura B.1: Metodología de aprendizaje supervisado.

primer paso es separar los datos disponibles en tres grupos: datos de entrenamiento, datos de prueba y datos de validación. Los datos de entrenamiento se preclasifican en función de la variable objetivo y los atributos y se construye así un modelo provisorio. El siguiente paso prueba como funciona el modelo con los datos de prueba, para los cuales se conoce el valor de la variable objetivo, y se ajusta el modelo para que minimizar el error obtenido. El modelo ajustado se aplica luego a los datos de validación, para los cuales también se conoce el valor de la variable objetivo [18]

En Minería de Datos son más los casos de aprendizaje supervisado que los de aprendizaje no supervisado.

Aprendizaje no supervisado:

- Clustering B.7
- Reglas de asociación B.5

Aprendizaje supervisado:

- Clasificación B.8
- Reglas de asociación B.5
- Arboles de decisión B.9
- Redes neuronales B.10

B.5. Reglas de asociación

La búsqueda de reglas constituye una función muy importante dentro de la minería de datos. La relación entre los atributos de un conjunto de datos se expresa en forma de reglas, de forma de simplificar el entendimiento de la información a la vista de los usuarios. Una regla muestra el comportamiento esperado ante una situación específica. Las reglas pueden ser de tres tipos: reglas de asociación, reglas de dependencia y de reglas de clasificación.

Los algoritmos de reglas de asociación generan reglas del estilo A→B, en donde el nuevo conocimiento B puede ser inferido cuando los hechos establecidos en A son verdaderos. Estos algoritmos buscan relaciones casuales en la base de datos y generan un conjunto de reglas.

Donde en otros métodos se valora la exactitud del resultado en las reglas de asociación se valora que la regla sea novedosa. Una regla novedosa es aquella que no se corresponde a comportamientos esperados y que no puede ser derivada fácilmente de reglas más simples. Las reglas de poco interés son aquellas que se hacen evidentes al observar los datos. Existen diferentes medidas para evaluar la calidad de una regla: simplicidad para ser comprendida por los usuarios, confiabilidad, utilidad, novedad e interés.

B.5.1. Algoritmo A priori

El algoritmo Apriori desarrollado por Agrawal et al. [19], es el más importante de todos los algoritmos de reglas de asociación, diseñado para trabajar en bases de datos transaccionales. Para poder aplicarlo a un conjunto de datos, es necesario discretizar los atributos numéricos, pues el algoritmo sólo trabaja con atributos categóricos. Las reglas de asociación se generan a partir de conjuntos de patrones, siendo la generación de los conjuntos de patrones es clave para la eficiencia del algoritmo. Se generar sucesivamente patrones candidatos de largo k a partir de patrones de largo k-1 y se podan los patrones menos frecuentes. Para determinar rápidamente los patrones menos frecuentes el algoritmo utiliza una estructura que permita acceso directo a partir del patrón, por ejemplo un hash. La figura B.2 presenta el pseudocódigo del algoritmo.

```
Algoritmo A priori
  Paso 1
  Generar conjuntos de patrones de largo 1 en C1
  Guardar los patrones más frecuentes en L1

  Paso k
  Generar conjuntos de patrones de largo k en Ck del conjunto de candidatos más
frecuentes Lk-1
      Combinar Lk-1p con Lk-1q
        insertar en Ck
        los elementos p.item1, q.item1, ... , p.itemk-1, q.itemk-1 de Lk-1 p, Lk-1q
         que cumplen (p.item1 = q.item1), ... ,(p.itemk-2 = q.itemk-2), (p.itemk-1 ≤
q.itemk-1)
      Generar todos los (k-1)-conjuntos del conjunto de candidatos en Ck
      Podar todos los conjuntos de candidatos de Ck en los que algun subconjunto (k-1)
del conjunto de candidatos no pertenece a Lk-1
  Recorrer las transacciones DB para determinar el soporte de cada candidate en Ck
  Guardar los candidatos más frecuentes en Lk
```

Figura B.2: Pseudocódigo algoritmo A Priori.

B.6. Razonamiento basado en memoria o basado en casos

La habilidad humana de razonar a partir de la experiencia depende de la habilidad de reconocer ejemplos del pasado, ésta es la esencia del razonamiento basado en memoria. [15] El razonamiento basado en memoria es una opción reciente en la resolución de problemas en Minería de Datos que puede ser aplicada para resolver gran variedad de problemas. Plantea utilizar una base de datos de casos previamente resueltos como repositorio para re-usar soluciones. Cada caso incluye un problema identificado y la solución que lo resuelve. Puede considerarse entonces que el conjunto de datos de entrenamiento son el modelo, es por eso que es muy importante la tarea de seleccionar este conjunto de datos [15] .

Una de las ventajas de estas técnicas que usan los datos en su estado natural. Deben definirse dos funciones: una función de distancia capaz de calcular la distancia entre dos registros y una función combinatoria capaz de combinar resultados para obtener una respuesta [15]. Otra ventaja del razonamiento basado en memoria es la habilidad de adaptarse, incorporando nuevos datos a la base de datos histórica, aprende las categorías y definiciones de los datos anteriores. Además produce buenos resultados y no es

necesario realizar entrenamiento para generar un modelo, se puede empezar a trabajar desde que se disponga de los datos. La mayor desventaja es que es un proceso pesado, ya que se procesan todos los datos históricos para encontrar similitudes con los nuevos datos [15]. Para poder aplicar razonamiento basado en memoria es necesario realizar las siguientes definiciones.

- Función de distancia.
- Función combinatoria.
- Conjunto de datos históricos.
- Elegir una forma eficiente para representar los datos.

B.7. Clustering

Clustering, en español agrupamiento, es la tarea de segmentar una población heterogénea en un conjunto de subgrupos homogéneos, llamados clusters[3]. Las técnicas de clustering pertenecen al aprendizaje no supervisado[20], simplemente reconoce estructuras y similitudes existentes en los datos. El resultado del clustering es un mapeo de los elementos de un conjunto de datos en uno de varios subgrupos que se forman al agrupar los datos basándose en similitudes o en modelos de densidad probabilística. Estas técnicas de agrupamiento son útiles para descubrir conocimiento en conjuntos de datos. El dividir un conjunto de datos en subgrupos tiene como objetivo ayudar a los usuarios a entender la estructura y características más frecuentes del conjunto de datos. Esta técnica plantea dos problemas serios. En ocasiones no es posible agrupar los datos en cluster, esto ocurre generalmente cuando los datos son demasiado complejos. En otros casos ocurre exactamente lo contrario, el número de clusters obtenido es muy grande y si el número demasiado grande generalmente no mejora el entendimiento del conjunto de datos.

Debido a que los datos pueden ser de naturaleza diversa no es posible entregar los datos crudos a los algoritmos de clustering. Es necesario representar estos datos de alguna forma numérica que posibilite la aplicación de los algoritmos. La técnica más usada es convertir todos los campos de los registros en valores numéricos y mapear los casos a puntos en el espacio representados por los registros. De esta forma se deduce la similitud entre los registros al aplicar comparaciones de distancias entre puntos del espacio.

[3]Cluster, término mencionado por primera vez por Tryon en 1939

Esta técnica tiene un par de carencias. La primera es que muchas variables no son adecuadas para ser la componente de un vector posición. La segunda es que el peso de un campo respecto a otro en el registro se pierde al representarlos en forma numérica. Una solución para la segunda es escalar las variables para que sus valores caigan en el mismo rango, normalizando, poniendo en un índice, o estandardizando los valores. Para ello existen varias opciones:

1. Dividir cada variable por un valor que corresponda a la brecha entre el valor más alto y el más bajo de esa variable, esto resulta en un mapeo en el rango de valores 0 a 1.
2. Dividir cada variable entre la media de todos los valores que toma la variable en el conjunto de datos.(se conoce como Indexar)
3. Restar la media a cada variable y dividirla entre la desviación estándar.(se conoce como Estandarización).Este cálculo refleja que tan lejos se encuentra el valor de la media.

Existe gran número de algoritmos de clustering; la elección del algoritmo adecuado para realizar clustering depende de los tipos de datos disponibles y el propósito de su aplicación [20]. Uno de las mayores dificultades cuando se quiere aplicar clustering a un conjunto de datos está en elegir el algoritmo adecuado para el problema y los valores para los parámetros correspondientes al mismo [21].

Los métodos de clustering pueden dividirse, según los criterios aplicados para realizar clustering, en las siguientes categorías [20].

- Métodos de partición
- Métodos jerárquicos
- Métodos basados en densidades
- Métodos basados en grillas
- Métodos basados en modelos

Existen algoritmos híbridos que combinan varios métodos, por lo tanto con esta clasificación no es posible catalogar todos los algoritmos en una única categoría.

En los algoritmos de clustering pueden identificarse dos partes: un mecanismo de búsqueda que genera candidatos y una función de evaluación que mide la calidad de los candidatos.

Un algoritmo de clustering debería tener las siguientes características: [31]

- **Genérico.** La función de evaluación debería poder aplicarse a cualquier dominio de datos.
- **Escalable.** Para poder manejar y almacenar grandes volúmenes de datos. Para que esto sea posible el número de evaluaciones de la función de evaluación debe ser tan pequeño como sea posible. Dado un conjunto de datos de N registros y D atributos, la complejidad del algoritmo debe ser subcuadrática en n y lo menor posible (lineal) en D.
- **Incremental.** Aún cuando el algoritmo sea escalable, es de esperar que para grandes volúmenes de datos los tiempos de ejecución sean altos. Es por eso que es deseable que sea incremental para poder monitorear el progreso a medida que se lleva a cabo la ejecución. El monitoreo puede servir para terminar antes la ejecución si el usuario determina que la calidad obtenida hasta el momento es aceptable o también en caso que sea evidente que no se encontrará una solución.
- **Robusto.** Debe ser robusto respecto al ruido y las desviaciones, estos no deben afectar los resultados.

Encontrar algoritmos de clustering que cumplan estas condiciones es uno de los objetivos de la Minería de Datos. La combinación de robustez y escalabilidad es particularmente difícil de lograr. En general se debe sacrificar uno para obtener el otro, el desafío se encuentra en lograr la mejor compensación para obtener ambos [31].

Un algoritmo de búsqueda consta de tres partes: punto de partida, método para generar soluciones, y criterio de parada. La elección del punto de partida es crítica en la mayoría de los algoritmos de búsqueda porque produce un sesgo al dirigir la búsqueda a una parte del espacio. Para sobreponerse al sesgo es necesario correr el algoritmo varias veces desde diferentes puntos de partida. El segundo paso es definir como se genera una nueva solución, lo cual introduce un nuevo sesgo. Finalmente se elige el criterio de parada que depende del problema a tratar, el volumen de datos, el tiempo de ejecución. A veces por necesidades de tiempo se para el algoritmo antes de que este se estabilice, introduciendo así un nuevo sesgo y en ocasiones que afecta sustancialmente a la solución. La mayoría de los métodos de Investigación Operativa mantienen durante la ejecución del algoritmo una solución a la vez. En cambio, algunas búsquedas heurísticas mantienen una población de soluciones e intentan mejorar la población en vez de una única solución.

Otras soluciones heurísticas mantienen distribuciones probabilísticas de la población. [21]

B.7.1. Métodos de partición

Dado un conjunto de datos que contiene n elementos y un número k tal que $k \leq n$, se construyen k particiones de los datos. Cada partición contiene al menos un elemento y cada elemento de los datos pertence a un único grupo. Estos métodos generan una partición inicial de los datos y luego aplican técnicas para mejorarla moviendo los elementos de una partición a otra. La calidad de las particiones se mide por medio de criterios de evaluación. Dado que obtener el óptimo significa la enumeración de todas las particiones posibles y su evaluación, la mayoría de los algoritmos utilizan heurísticas. Los métodos de partición pueden separarse entonces en dos categorías, los que buscan el óptimo global y los que buscan un óptimo local [13].

Métricas de distancia La mayoría de los algoritmos de partición se basan en la distancia entre los objetos. Es esencial entonces encontrar una buena representación geométrica de los datos. Las medidas de distancia son sensibles a variaciones de representación, y esto puede afectar los resultados. Se puede usar como medida de distancia cualquier función d que tome como entrada dos puntos del espacio y de como resultado un valor numérico. Para que esta función sea una función de distancia en términos matemáticos debe cumplir las siguientes propiedades.

- distancia(X, Y) = 0 si y solo si X = Y
- distancia(X, Y) ≥ 0 $\forall X$, $\forall Y$
- distancia(X, Y) = distancia(Y, X)
- distancia(X, Y) $\geq$ distancia(X, Z) + distancia(Z, Y)

Estas propiedades tienen una interpretación menos estricta en clustering que las expresadas matemáticamente. La segunda y tercera propiedad indican que si dos puntos son muy similares caerán en el mismo cluster. La última condición significa que agregar un nuevo centro al cluster no hará que dos puntos estén más cercanos.

El algoritmo más usados es k-medias propuesto por MacQueen en 1967 y k-medioides o PAM [4], propuesto por Kaufman y Rousseeuw en 1987 [20].

[4] Partition around medioids

En k-medias, el cluster se representa con la media de los elementos que contiene y en k-medioides por el elemento más cercano al centro del cluster. Ambas heurísticas plantean una limitación muy importante, sólo funcionan adecuadamente cuando los clusters tienen formas circulares o esféricas. Para formas más complejas es necesario aplicar otras técnicas, como las que proponen los métodos basados en densidades B.7.3. También presentan limitaciones en cuanto al volumen de datos, por lo cual han surgido otros algoritmos especialmente diseñados para grandes volumenes de datos y siguen surgiendo a medida que la noción de lo que es un volumen grande cambia gracias a avances tecnológicos. CLARA [5] fue propuesto por Kaufman y Rousseeuw en 1990. CLARA toma varias muestras de los datos y calcula los medioides de cada muestra usando PAM. Luego da como resultado el mejor de los resultados obtenidos con estas muestras. Si la muestras elegidas son correctamente contruidas y resultan realmente representativas, los resultados serán similares a los que se pueden obtener con el conjunto completo de datos. CLARANS [6] fue propuesto por Ng y Han en 1994 [22]. El objetivo de su desarrollo fue mejorar la calidad y la escalabilidad de CLARA especialmente para Minería de Datos espacial.

[5] CLARA: Clustering Large Applications

[6] CLARANS: Clustering Large Applications based upon RANdomized Search

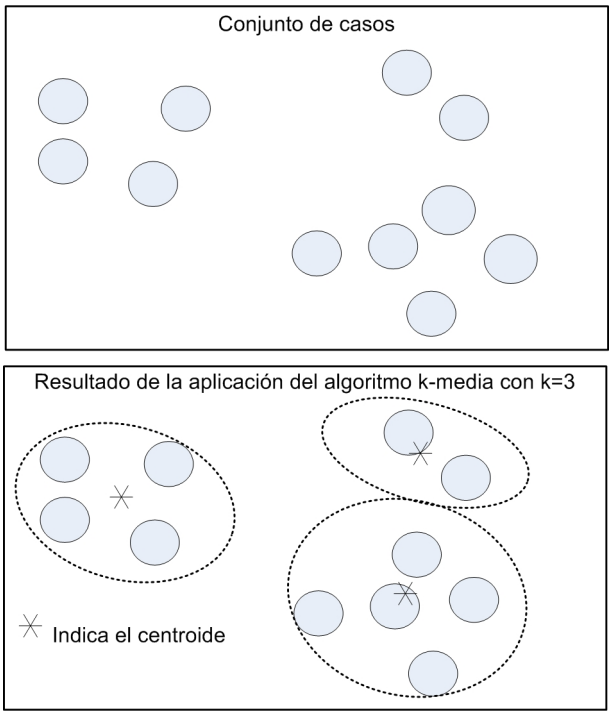

Figura B.3: Clustering K-medias.

K-Medias

El algoritmo K-medias [23] es un algoritmo interactivo donde los elementos se mueven a través de K clusters hasta que el se logra el conjunto deseado. La K representa un número fijo de clusters que debe ser definido con antelación. Es necesario elegir K semillas entre los puntos que representan los datos para ser las estimaciones iniciales de los centroides de cada uno de estos cluster, con esto se dividen los datos en K clusters iniciales. Luego se calcula la media de los atributos de los datos en cada cluster. El proceso de recalcular la media para cada nuevo dato en el cluster se realiza hasta que los centroides de los cluster dejan de moverse, es decir, hasta que la media de los atributos de los datos del cluster iguala al centroide.

Los algoritmos K-medias aunque son muy usados, tienen varias desventajas.

- En este tipo de algoritmos es necesario elegir por adelantado el número de clusters K. Por lo general no se conoce el número probable de

clusters, y el objetivo es en realidad encontrarlo. Un método es probar diferentes valores y elegir el mejor, pero para esto es imprescindible saber como evaluar la efectividad del método.

- La calidad de la solución final depende en gran parte del conjunto inicial de clusters elegido, cuando el conjunto de datos es reducido esta dependencia se acentúa. Diferentes condiciones iniciales seguramente resulten en diferentes agrupamientos.
- La representación de datos necesaria para aplicar los cálculos de distancias hacen que todos los atributos sean de igual peso, asumir esto es equivalente a perder parte de la información existente en los datos.
- Aunque el algoritmo debe converger, no hay límite para el número de iteraciones requeridas. Una técnica posible es elegir parar el algoritmo después de cierto número de iteraciones, pero ni en este caso ni en el de convergencia existen garantías de haber llegado al resultado óptimo.
- El resultado es un cluster circular porque se basa en distancias a un centroide, naturalmente los datos pueden agruparse en formas no circulares.

El cluster Ki = {ti1, ti2, ... , tim} se define como

$$m_i = (1/m) \sum_{j=1}^{m} t_{ij}. \tag{B.1}$$

Existen innumerables versiones de este algoritmo, la que se reconoce como la primera de ellas es la de J.B.MacQueen de 1967 [23].En B.4 se plantea un pseudocódigo simplificado.

ALGORITMO DE CLUSTERING K-MEDIAS
Entrada: B base de datos que contiene n elementos
k número de clusters tal que $k \leq n$
Salida: Conjunto de k clusters
Definir k clusters eligiendo arbitrariamente k centroides
repeat
Determinar las coordenadas del centroide
Hallar la distancia de cada punto a los centroides
Agrupar los puntos en base a la distancia mínima al centroide
until (no se produzcan cambios de cluster)

Figura B.4: Pseudocódigo algoritmo clustering k-medias.

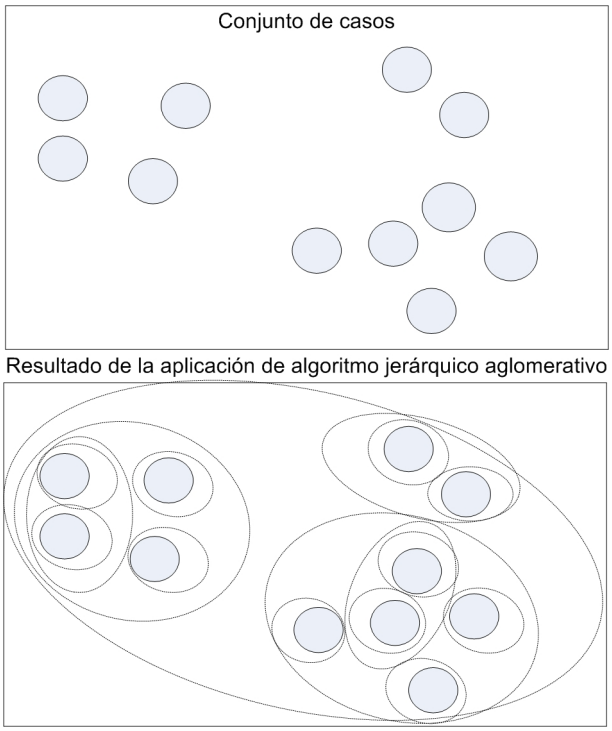

Figura B.5: Clustering jerárquico aglomerativo.

B.7.2. Métodos jerárquicos

Dado el conjunto de datos lo descomponen jerárquicamente. Los algoritmos jerárquicos pueden ser o bien aglomerativos (o bottom-up[7]) o divisivos (o top-down [8]). Los algoritmos aglomerativos parten de la separación inicial en que cada objeto pertenece a un grupo. Luego los grupos son combinados en base a similitudes de sus elementos hasta que se obtiene un único grupo o se llega a una condición de parada. Los algoritmos divisivos por el contrario parten de un único grupo que contiene todos los objetos y realiza divisiones sucesivas del grupo hasta obtener grupos unitarios o hasta una condición de parada. Los algoritmos jerárquicos son rígidos en el sentido que, luego de realizada una unión o división no puede deshacerse. Luego de generada la jerarquía el usuario puede elegir un número k de clusters, lo que provoca un corte transversal a la jerarquía en el nivel que corresponda para obtener el

[7]bottom-up: de abajo hacia arriba
[8]top-down: de arriba hacia abajo

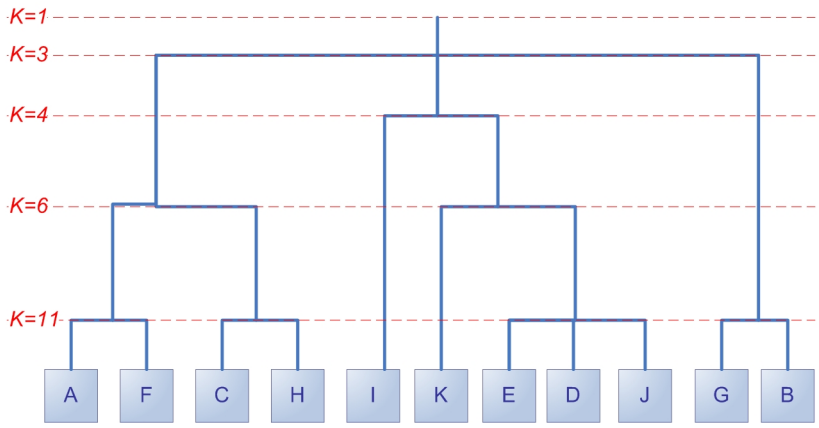

Figura B.6: Clustering jerárquico aglomerativo, vista en formato árbol.

número de clusters indicado [21].

Criterios para determinar distancia entre clusters Dado los clusters A y B existen diversas formas de calcular la distancia entre ellos. [18]

- **Acoplamiento simple,** conocido también como el vecino más cercano. Es la mínima distancia entre cualquier par de elementos de los clusters A y B. Es la distancia entre los elementos con mayores similitudes de cada cluster.
- **Acoplamiento completo,** conocido también como el vecino más lejano. Es la máxima distancia entre cualquier par de elementos de los clusters A y B. Es la distancia entre los elementos más disimiles de cada cluster.
- **Acoplamiento promedio.** Es la distancia promedio de todos los elementos del clusters A a los elementos del cluster B.

El acoplamiento simple tiende a generar clusters largos y delgados, que en ocasiones agrupa elementos heterogéneos. El acoplamiento completo por su parte tiende a formar clusters compactos de forma esférica.

No existen gran variedad de algoritmos de clustering jerárquico puros. La razón fundamental es que plantean grandes riesgos en la realización de agrupamientos, y más aún en el caso de divisiones, en etapas tempranas de procesamiento. Los errores en estas decisiones generan clusters de calidad

ALGORITMO DE CLUSTERING JERÁRQUICO AGLOMERATIVO

Entrada: X conjunto de objetos $\{x_1, ..., x_n\}$
la función de distancia $dis(c_1, c_2)$

Salida: Modelo jerárquico

Algoritmo:

for i = 1 to n
$c_i = \{x_i\}$;
end for
$C = \{c_1, ..., c_b\}$;
l = n + 1;
while (C.size $\geq$ 1) do
$(cmin1, cmin2) = min(dis(c_i, c_j))\forall c_i, c_j \in C$;
remove(C, cmin1);
remove(C, cmin2);
add(C,$\{cmin1, cmin2\}$);
l = l + 1;
end while

Figura B.7: Pseudocódigo algoritmo clustering jerárquico aglomerativo.

pobre. Otra de las razones es que no son fáciles de escalar. Dentro de los algoritmos de clustering puro se encuentran dos algoritmos presentados por Kaufman y Rousseeuw en 1990. El algoritmo aglomerativo AGNES y el algoritmo divisivo DIANA. AGNES [9] usa el método de conexión simple y une los clusters con menos disimilitudes [20]. El mayor potencial de los algoritmos que aplican métodos jerárquicos se logran cuando se combinan con otros métodos de clustering. Dentro de esta categoría se encuentran los algoritmos BIRCH, CURE, ROCK y CHAMELEON.

BIRCH[10] fue presentado por Zhang, Ramakrishnan y Livny en 1996 [24]. BIRCH aplica una técnica de clustering multifase y genera un árbol con pesos, balanceado y cuyos nodos suman cero [20].

CURE [11] fue presentado por Guha, Rastogi y Shim en 1998 [25]. Con la robustez como objetivo, CURE representa cada cluster con un conjunto de punto en vez de con un sólo punto. Esto le permite funcionar muy bien con clusters de forma irregular. El algoritmo consiste en la ejecución de dos fases,

[9] AGNES: Agglomerative Nesting
[10] BIRCH: Balanced Iterative Reducing and Clustering using Hierarchies
[11] CURE: Clustering Using Representatives

en la primera se utiliza partición sobre una muestra de datos representativa y en la segunda fase se integra clustering jerárquico aglomerativo para obtener los clusters definitivos.

En 1999 los mismos investigadores presentaron una algoritmo que extiende CURE a datos categóricos llamado ROCK. ROCK construye un grafo disperso de una matriz de similitudes dada usando un umbral de similitud y el concepto de vecinos dispersos; luego ejecuta el algoritmo de clustering sobre el grafo.

CHAMELEON, fue desarrollado por Karypis, Han y Kumar y presentado en 1999. Es una algoritmo de dos fases, en la primera fase usa partición para generar clusters de pocos elementos del conjunto de datos. En la segunda fase se aplica otro algoritmo para encontrar los clusters verdaderos por aglomeración. [26] CHAMELEON explota la cercanía de los clusters, las características internas de los elementos y la interconectividad [20]. El algoritmo obtiene buenos resultados para clusters de muchas variables. El uso de interconectividad y cercanía le permite identificar mejorar las similitudes [26].

B.7.3. Métodos basados en densidades

Los métodos basados en densidades fueron desarrollados para descubrir clusters de formas irregulares. Estos algoritmos identifican clusters como regiones de alta densidad de elementos en el espacio de datos separadas entre si por regiones de baja densidad. La idea es agrandar cada cluster hasta que la densidad en el vecindario exceda un umbral. La densidad es la cantidad de elementos de un cluster, por lo tanto el umbral se determina según este valor. Por ejemplo, el umbral puede estar dado por la cantidad de elementos en un radio dado para cada elemento de un cluster.

Para poder explicar en detalle un algoritmo basado en densidades es necesario realizar previamente algunas definiciones y establecer los parámetros que utilizan.

Centro se llama a un elemento en cuyo radio ϵ existen no menos de *Min* elementos.

Un objeto p es **directamente alcanzable por densidad** desde q con respecto a un radio ϵ si p pertenece al ϵ-vecindario de q.

Cluster basado en densidad es un conjunto de elementos conectados por densidad, la cual es máxima respecto a la densidad alcanzable y todo elemento que no pertenezca a un cluster es ruido.

Parámetros : ϵ, *Min*

Un algoritmo basado en densidades chequea el ϵ-vecindario de cada punto de la base de datos. Si el ϵ-vecindario de un punto p contiene más elementos que *Min*, se crea un nuevo cluster con p como centro. Luego reúne elementos directamente alcanzables por densidad desde p, produciendo fuciones de clusters si fuera necesario. El proceso termina cuando no se puede agregar ningún nuevo elemento a los clusters existentes. El resultado es que para cada objeto del cluster, el vecindario en un radio dado ϵ contiene por lo menos una cantidad mínima de elementos indicada en el parámetro *Min* [20].

Algunos métodos basados en densidades son DBSAN, OPTICS y DENCLUE. DBSCAN fue propuesto por Ester, Kriedgel, Sander y Xu en 1996 [27] con el objetivo de aplicarse a bases de datos espaciales de gran tamaño, recibe del usuario los valores de los parámetros. PDBSCAN es la versión paralela del algoritmo DBSCAN y GDBSCAN [12] es una versión genérica presentada en 1998 por Sander.

OPTICS[13] fue propuesto en 1999 por Ankerst, Breuning, Kriegel y Sander [28] con el objetivo de independizar los algoritmos de la elección de parámetros que realiza el usuario. La estructura de OPTICS es similar a la de DBSCAN, pero en vez de generar un clustering explícito, crea un ordenamiento de la base de datos representando la estructura de densidades. La estructura resulta versátil para análisis de clusters interactivo y automático, ya que esta no genera dependencia de los parámetros [28].

DENCLUE[14] fue desarrollado por Hinnerburg y Keim en 1998. El método se basa en estas tres ideas: la influencia de cada punto en su vecindario puede modelarse con una función de influencia, la densidad total del espacio es la sumatoria de las densidades de los puntos, los clusters pueden determinarse matemáticamente identificando la densidad de atracción. Las ventajas de este algoritmo son su base matemática tanto en metodología como en representación de los clusters, su buen comportamiento en conjuntos de datos con abundante ruido y su manejo eficiente de memoria al guardar información sólo de los puntos del espacio ocupados por elementos. [20]

[12] GDBSCAN: Generalized Density Based Spatial Clustering of Applications with Noise
[13] OPTICS: Ordering Points To Identify the Clustering Structure
[14] DENsity-based CLUstEring

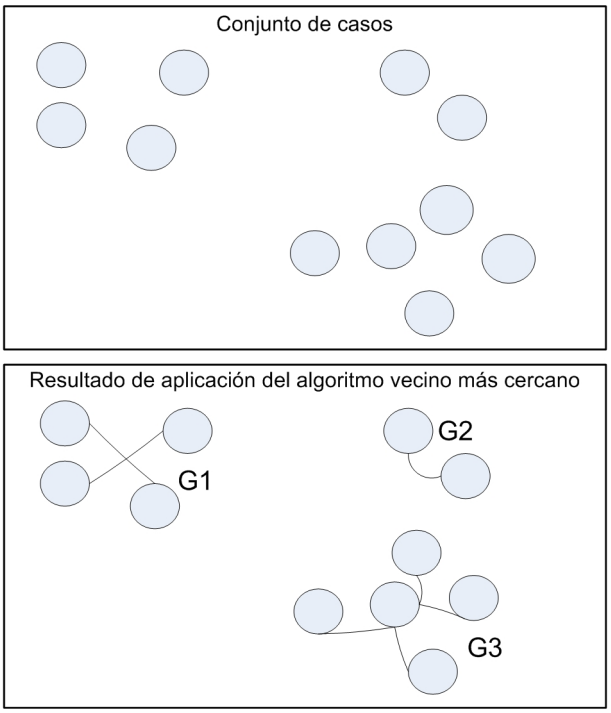

Figura B.8: Clustering K vecinos más cercanos.

K vecinos más cercanos

K vecinos más cercanos, o su abreviación k-NN [15] es una técnica de predicción adecuada para modelos de clasificación. En contra a otras técnicas de clasificación los datos no son escaneados o procesados para crear el modelo. En vez de esto, los datos de entrenamiento son el modelo. Cuando un nuevo caso o instancia se presenta al modelo, el algoritmo busca en todos los datos para encontrar un subconjunto de casos que sea los más similares al original y los usa para predecir la salida. Hay 2 datos esenciales en el algoritmo k-NN: el número de casos cercanos a ser usados (k) y la métrica para medir que se considera como cercano. Cada uso del algoritmo k-NN requiere que especifiquemos un número entero positivo k que determine la cantidad de casos que son analizados cuando se predice un nuevo caso. K-NN refiere a una familia de algoritmos que pueden ser denotados como 1-NN, 2-NN, 3-NN, etc. Por ejemplo un algoritmo 3-NN indica que se usarán los tres casos

[15] KNN: K nearest neighbors

más cercanos para predecir la salida del nuevo caso. Dado un conjunto de datos en un espacio lo podemos representar como un conjunto de puntos. El algoritmo conecta cada punto con el punto más cercano B.8.

Este algoritmo es fácil de usar, eficiente si el número de casos no es demasiado grande, se puede representar visualmente el resultado, es robusto al ruido pero no a atributos no significativos.

B.7.4. Métodos basados en grillas

Los métodos basados en grillas dividen el espacio de objetos en un número finito de celdas con una grilla. Las operaciones de clustering se realizan sobre la grilla, la principal ventaja es su velocidad de procesamiento, que depende solamente del número de celdas de la grilla y es independiente del número de elementos en los datos. [20]

STING [16], es un algoritmo basado en grillas desarrollado por Wang, Yang y Muntz en 1997 [29]. STING divide el espacio en varias capas de celdas rectangulares. La división se realiza en forma jerárquica, cada celda de un nivel alto es particionada en un conjunto de celdas en el nivel inferior. Se almacena para cada celda información estadística: medias, máximos, mínimos, desviación estándar. La calidad de STING depende de la granularidad del nivel inferior de la grilla, si es pequeña la calidad es buena pero el costo de procesamiento es más alto, si es grande la calidad es inferior [20].

CLIQUE es un algoritmo basado en grillas y densidades presentado por Agrawal en 1998, particiona el espacio en celdas rectangulares disjuntas, calcula la densidad de cada celda y encuentra los clusters comparando la densidad contra un parámetro de entrada del modelo. Maneja correctamente volumenes grandes de datos, es insensible al orden de las entradas y su escalabilidad es lineal a partir del número de dimensiones de los datos. Es un método simple, pero su efectividad no es la mejor [20].

WaveCluster fue desarrollado por Sheikholeslami, Chatterjee y Zhang en 1998 [30] , es un algoritmo basado en grillas y densidades. Primero divide el espacio en grillas y luego aplica transformaciones para encontrar regiones densas en el espacio. Maneja volumenes grandes de datos en forma eficiente, descubre cluster de forma arbitraria, es insensible a orden de las variables de entrada [20].

[16] STING: STatistical INformation Grid

B.7.5. Métodos basados en modelos

Los métodos basados en modelos intentan encontrar el mejor modelo para los datos. Se realiza una hipótesis de modelo para cada uno de los clusters y luego se optimizan en base a los datos. Se basa en la hipótesis que los datos están generados por una mezcla de distribuciones probabilísticas. Los clusters son encontrados mediante la construcción de una función de densidad que refleja la distribución espacial de los elementos. El número de clusters se determina por medio de estándares estadísticos, esto genera métodos de clustering robustos [20]. Los métodos basados en modelos pueden encararse de dos formas: estadística o redes neuronales. En la categoría estadística se encuentran COBWEB, CLASSIT y AutoClass; en la de redes neuronales SOM.

COBWEB fue presentado por Fisher en 1987[31], es un algoritmo de aprendizaje no supervisado que realiza una costrucción jerárquica top-down de clusters. Utiliza una medida de utilidad que es la sumatoria de las distribuciones de los atributos. Es un método que presenta varias limitaciones. Asume que la distribución de los atributos es estadísticamente independiente entre sí, lo cual no se cumple para todos los conjuntos de datos. Representar los clusters por su distribución hace que el método necesite muchos recursos para almacenar los valores correspondientes a cada cluster y por lo tanto no es adecuado para grandes volumenes de datos. CLASSIT fue presentado por Gennari, Langley y Fisher en 1989 como una versión incremental de COBWEB para datos continuos. Almacena una distribución normal continua para cada atributo en cada nodo y modifica la medida de utilidad de COBWEB a una integral, pero no soluciona los problemas de COBWEB para datos grandes. AutoClass, presentado por Cheeseman y Stutz en 1996, implementa un método de clustering basado análisis estadístico Bayesiano [20].

SOM [17] fue propuesto por Kohonen en 1981. SOM realiza una especie de proyecciónde una entrada de m dimensiones en un espacio de menor orden (en general 2 dimensiones). Este mapeo sirve para identificar clusters de elementos similares del espacio original [20].

B.7.6. Comparación de métodos de clustering

Dentro de los algoritmos de partición el más usado es el k-medias. Esto no significa que el mismo no tenga limitaciones, las cuales se detallaron en

[17] Self Organized Feature Map

B.7.1. El algoritmo k-medioides es más robusto que el algoritmo k-medias en presencia de ruido y valores atípicos [18] (en este caso valores muy lejanos a la media), pero es más costoso en tiempo de procesamiento[20]. k-medias es un algoritmo greedy [19] y k-medioides no lo es [21].

La principal distinción entre los métodos jerárquicos y de partición es que los métodos jerárquicos producen una serie de particiones anidadas, en tanto los de partición producen sólo una partición [32]. Los métodos de partición plantean por lo tanto la ventaja de simplicidad respecto a los jerárquicos; la construcción de los árboles o dendrogramas plantean tanto complejidad computacional como mayor necesidad de recursos.

B.8. Clasificación

La clasificación consiste en asignar etiquetas de clase a un patrón, es el mapeo de un conjunto de datos en un conjunto de clases predefinidas y disjuntas. Estas clases son clases de equivalencia.

"Dada la base de datos $B = \{t_1, ..., t_n\}$ *de registros (ítems, registros y variables) y el conjunto de clases* $C = \{c_1, ..., c_m\}$*; el problema de clasificación se puede definir como la función de mapeo de* $f : B \rightarrow C$ *donde cada* t_i *es asignado a una clase. Una clase* C_j *contiene aquellos registros que se mapearon a ella, entonces* $C_j = \{t_i / f(t_i) = C_j, 1 \leq i \leq n, t_i \in B\}$*." [5]*

En Minería de Datos, clasificación es la tarea de analizar y clasificar un conjunto de datos con el objetivo de generar un modelo que pueda ser usado en el futuro para clasificar nuevos conjuntos de datos.

La estimación y la predicción pueden ser vistas como tipos de clasificación. La predicción es un caso de clasificación donde un atributo es clasificado en uno de un conjunto de clases predefinidas y disjuntas. La estimación es un caso de clasificación en clases no conocidas.

Los resultados obtenidos por los algoritmos de clasificación pueden evaluarse analizando la exactitud de la clasificación. El valor correspondiente

[18] Ver definición de valor atípico en el Glosario

[19] Los algoritmos greedy aplican la meta-heurística de elegir el óptimo local en cada paso con la esperanza de encontrar de esta forma el óptimo global. Este método en general no alcanza el óptimo, ya que las elecciones hechas en cada paso en ocasiones alejan del óptimo y además la mayor parte de estos algoritmos no procesan exhaustivamente los datos. Su principal ventaja es la rapidez de procesamiento.

a la exactitud del algoritmo es comúnmente calculado como el porcentaje de registros de la base que son clasificados correctamente. Para un cálculo más exigente debería agregarse el peso de las clasificaciones incorrectas. Cualquiera de estas evaluaciones dependen de la opinión del usuario, tanto en si un caso se considera correctamente clasificado como en la asignación de pesos en caso de error.

B.8.1. Naive Bayes

Una red bayesiana es un modelo gráfico que encuentra relaciones probabilísticas entre las variables de un sistema. Los componentes básicos de una red bayesiana son los nodos, cada uno de los cuales representa una variable del sistema, las relaciones entre las variables, indicadas gráficamente mediante flechas, y valores de probabilidad de asociación. El uso de probabilidades y las relaciones permiten calcular probabilidades desconocidas.

Asumiendo que la contribución de todos los atributos (variables) es independiente y que cada uno contribuye de igual forma al problema de clasificación, el esquema Naive Bayes (referencia) se basa en la regla de Bayes (referencia Teorema de probabilidad de Bayes, Thomas Bayes) de probabilidad condicional. Naive, cuya traducción al español es inocente, refleja esta asunción.

Naive Bayes es una técnica de clasificación que es al mismo tiempo de predicción y descripción. Analiza las relaciones entre cada variable independiente y la variable dependiente para derivar una probabilidad condicional para cada relación. Cuando cada caso es analizado, combinando los efectos de las variables independientes en las variables dependientes da una variable de predicción (la salida que es predecida). En teoría, la predicción de Naive Bayes sólo será correcta si las variables independientes son estadísticamente independientes entre sí, esto en general no se cumple. Naive-Bayes requiere de sólo una pasada por el conjunto de datos de entrenamiento para generar un modelo de clasificación.

La técnica tiene múltiples ventajas: facilidad de uso, simplicidad por requerir sólo una pasada por los datos de entrenamiento, manejo sencillo de los datos faltantes omitiendo su probabilidad en el cálculo de pertenencia a cada clase. Por otra parte, las desventajas son que a veces no produce buenos resultados, los atributos no son usualmente independientes y a veces es necesario usar subconjuntos independientes de ellos, la técnica no maneja datos continuos. No hay modelo, es fácil de usar, eficiente y tolerante al ruido. Esto

hace que sea una muy buena técnica de Minería de Datos; pero no maneja datos continuos, por lo tanto cualquier variable independiente o dependiente que contenga datos continuos debe ser discretizados. Naive-Bayes es un proceso simple. Durante el entrenamiento la probabilidad de cada salida (valor de la variable dependiente) se calcula contando cuantas veces ocurre en los datos de prueba, esto se llama probabilidad previa. Además se calcula la frecuencia de cada valor de cada variable independiente en combinación con cada valor de la variable dependiente. Estas frecuencias se usan para calcular probabilidades condicionales que se combinan con las probabilidades previas para hacer predicciones. Naive-Bayes utiliza las probabilidades condicionales para modificar las probabilidades previas.

B.9. Arboles de decisión y reglas de decisión

Las técnicas de modelado de árboles de decisión son los métodos lógicos más usados. Pertenecen a los métodos inductivos del aprendizaje automático que aprenden a partir de ejemplos preclasificados. Se utilizan en Minería de Datos para modelar clasificaciones en los datos. Una de las mayores ventajas de los algoritmos de árboles de decisión es la posibilidad de representar gráficamente los resultados en la forma de un árbol. Otras ventajas de este tipo de algoritmo es la facilidad de uso, la posibilidad de utilizar datos discretos y continuos; la tolerancia al ruido, a atributos no significativos y a datos faltantes. Dentro de los árboles de decisión se destaca la familia de los TDIDT (Top Down Induction Trees), la mayoría de los algoritmos pertenecen a este grupo.

Un modelo de árbol de decisión es un modelo computacional que consiste de tres partes: un árbol de decisión, un algoritmo para crear el árbol, un algoritmo que aplica el árbol a los datos y resuelve el problema en consideración.

"Un árbol de decisión es un árbol donde la raíz y cada nodo interno esta etiquetado con una pregunta. Los arcos que salen de cada nodo representan cada posible respuesta a la pregunta del nodo asociada. Cada nodo hoja representa la predicción de una solución al problema en consideración." [5]

"Un árbol de decisión es una estructura que puede ser usada para dividir un conjunto grande de registros sucesivamente en subconjuntos de registros aplicando una secuencia de reglas de decisión simples." [15]

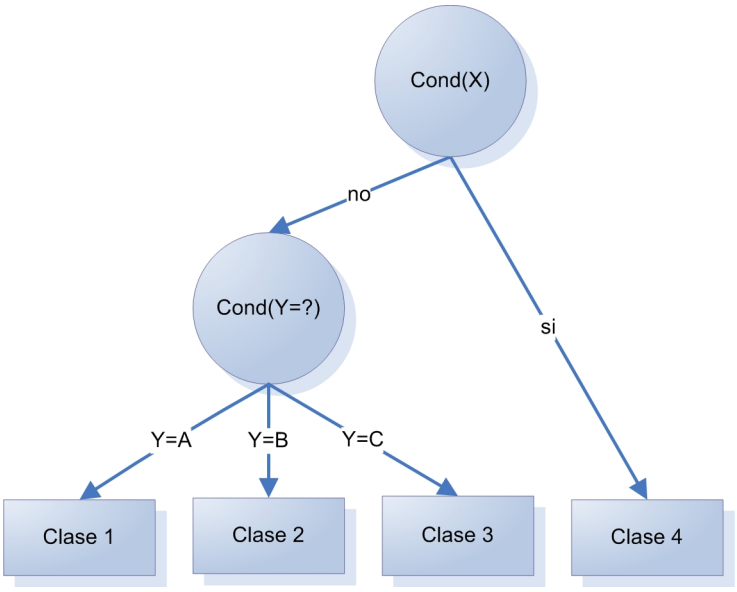

Figura B.9: Ejemplo de árbol de decisión.

La mayoría de los algoritmos de árboles de decisión se diferencian en la forma en que el árbol de decisión es construido. El árbol de decisión es generado dividiendo los registros del conjunto de datos en subconjuntos disjuntos o puede ser creado por un experto en los datos. La raíz del árbol contiene todos los registros y las preguntas en los nodos internos permiten dividirlos. Las preguntas en los nodos internos son reglas simples sobre uno o más atributos de los datos. Finalmente se obtienen subconjuntos disjuntos que corresponden a las hojas del árbol. Bajando en los niveles del árbol se debe lograr decrecer en diversidad de casos presentes en cada subconjunto o lo que es igual aumentar la pureza u homogeneidad de los mismos.

El factor más importante en el rendimiento de los algoritmos de árboles de decisión es el tamaño del conjunto de datos de entrenamiento y como son elegidos los atributos de división.

Consideraciones a tener en cuenta en los algoritmos.

- Elección de los atributos de división
- Determinar el orden de los atributos de división
- Número de divisiones
- Estructura del árbol. Para mejorar el rendimiento es mejor un árbol

balanceado con pocos niveles, pero la elección de estructura dependerá de la realidad.

- Criterio de parada. La creación del árbol termina cuando los datos de entrenamiento están perfectamente clasificados. Existen situaciones en las que se desea parar antes para evitar la creación de árboles demasiado grandes y para evitar overfitting. Esto hace necesario elegir entre rendimiento y precisión.
- Datos de entrenamiento. La estructura del árbol dependerá de los datos de entrenamiento, si el volumen de datos de entrenamiento es demasiado chico el árbol generado no funcionará adecuadamente con otros conjuntos de datos más generales. Si el volumen es demasiado grande, el árbol generado posiblemente overfitting.
- Podas. Luego de creado el árbol, es necesario mejorar su rendimiento eliminando comparaciones redundantes o subárboles completos en la fase de poda.

```
Algoritmo de Arbol de decisión
  Entrada: Arbol de decisión A, Base de datos de entrada B
  Salida: Modelo de predicción M
  Algoritmo:
  for each (t ∈ D) do
     n = raiz(T);
     while n no es una hoja do
        Obtener una respuesta a la pregunta en n aplicando t;
        Identificar arco que sale de t y contiene la respuesta correcta;
        n = nodo fin del arco identificado;
     endwhile
     Realizar una predicción de t basado en la etiqueta de n;
  endfor
```

Figura B.10: Pseudocódigo algoritmo de árboles de decisión.

Los árboles de decisión son muy útiles para la resolución de problemas de clasificación. Una vez construido el árbol de decisión, se aplica a cada registro de la base de datos, obteniendo una clasificación de la misma. Las ventajas de su uso en problemas de clasificación son su simplicidad de uso y eficiencia. Además son fácilmente escalables a bases de datos grandes ya que el tamaño del árbol de decisión es independiente del tamaño de la base

de datos. La aplicación del árbol a la base es de orden O(n) donde n es el número de registros de la base de datos. Dentro de las desventajas de su aplicación se encuentra la dificultad para manejar datos continuos, para lo que es necesario dividir el dominio en categorías manejables. Debido a que el árbol se construye a partir de datos de entrenamiento es posible que ocurra overfitting A.6.7.

Una tema de importancia que se presenta al aplicar métodos de clasificación o regresión es que los árboles finales pueden llegar a ser muy grandes. En la práctica, cuando los datos de entrada son complejos y contienen muchas categorías para los problemas de clasificación, y muchos predictores posibles para realizar la clasificación, los árboles generados pueden llegar a ser muy grandes. Esto no representa solamente un problema de cómputo, sino de lograr presentar los árboles de una manera simple que sea fácilmente accesible al analista y a los usuarios finales.

La efectividad de un árbol de decisión se mide probando como clasifica un conjunto de datos de prueba y midiendo el porcentaje de los mismos que fue correctamente clasificado.

B.9.1. ID3

La técnica ID3 fue propuesta a fines de los setenta por J. Ross Quinlan [1]. Consiste en la construcción iterativa de árboles de decisión, basándose en información teórica y en el intento de minimizar el número esperado de comparaciones. ID3 fue diseñado para casos en que existen muchos atributos y el conjunto de entrenamiento es grande, con el objetivo de obtener un árbol de decisión bueno en pocas iteraciones. El algoritmo ID3 forma parte de la familia de los TDIDT. La construcción del árbol se realiza con un método top down [20], comenzando del conjunto de objetos y la especificación de sus propiedades.

En cada nodo del árbol, una propiedad es testeada [21] y el resultado es usado para subdividir el conjunto de objetos. Este proceso es realizado en forma recursiva hasta que el conjunto en un subárbol determinado es homogéneo respecto al criterio de clasificación, entonces se convierte en una hoja del árbol. La idea básica del algoritmo de inducción es realizar las preguntas que provean más información y la estrategia es elegir primero los atributos de división que producen la mayor ganancia de información.

[20] Top Down : de arriba hacia abajo

[21] Propiedad: condición relativa a un atributo del conjunto de datos.

La cantidad de información asociada a un atributo está relacionada con su probabilidad de ocurrencia. El concepto usado para la cuantificación de la información es la entropía. La información es máxima cuando la entropía se minimiza. La entropía es una medida de la incertidumbre o aleatoriedad del conjunto de datos [5]. En cada nodo, la propiedad a testear es escogida basándose en criterios teóricos que buscan maximizar la información ganada y minimizar la entropía. En términos más simples, la propiedad que divide los candidatos en los subconjuntos más homogéneos.

Pseudocódigo de algoritmo ID3

Entrada: R: conjunto de atributos no categóricos,
C: atributo categórico,
S: conjunto datos de entrenamiento
Salida: Árbol de Decisión DT
Algoritmo:
if vacío(S)
DT = único nodo con valor FALLO;
if S está compuesto por registros con igual valor X para el atributo categórico,
DT = único nodo con valor X;
if vacío(R)
DT = único nodo con valor igual al más frecuente de los valores
del atributo categórico de los registros de S;
Sea D el atributo que produce la mayor Gain(D,S) entre los atributos de R;
Sea { d_j / j=1,2, .., m } el conjunto de valores del atributo D;
Sea { S_j / j=1,2, .., m } los subconjuntos de S que consisten de
registros con valor d_j para el atributo D;
DT = árbol con raíz etiquetada D y arcos etiquetados $d_1, d_2, .., d_m$
que salen de la raíz y llegan respectivamente a los árboles
$ID3(R - \{D\}, C, S_1), ID3(R - \{D\}, C, S_2), .., ID3(R - \{D\}, C, S_m)$;

Figura B.11: Pseudocódigo algoritmo ID3.

Existen dos métodos para construir el árbol a partir de un conjunto de datos de entrenamiento. En el primer método se construye el árbol con un subconjunto aleatorio de los datos de entrenamiento (llamado ventana). Luego de construído el árbol se clasifican con él el resto de los datos de entrenamiento. Si todos clasifican correctamente se llegó al árbol definitivo,

sino se agregan más datos a la ventana y se construye nuevamente el árbol hasta que es el mismo clasifique correctamente todos los datos de entrenamiento. El segundo método construye el árbol directamente con el conjunto completo de datos de entrenamiento. Las evidencias empíricas muestran que el método es más rápido si se aplica al conjunto completo de entrenamiento. El mismo Quinlan propone el método de la ventana como una simple alternativa al momento de trabajar con grandes volúmenes de datos.

La principal limitación del algoritmo ID3 es que sólo permite su aplicación en conjunto de datos discretos.

A continuación se plantean en detalle los cálculos en los cuales se basa este algoritmo para generar el árbol de decisión. Dada una distribución de probabilidad $P = (p_1, p_2, ..., p_n)$ donde $\sum_{i=1}^{n} p_i = 1$, entonces la información transmitida por esta distribución, llamada **Entropía** es

$$E(P) = -\sum_{i=1}^{n}(p_i * log(p_i)) \tag{B.2}$$

Cuanto más uniforme es la distribución de probabilidad, mejor es la información que la misma proporciona.

Si un conjunto de datos T se particiona en $C_1, C_2, ..., C_k$ clases disjuntas en base a un atributo categórico, la distribución de probabilidad de la partición es $P(|C_1|/|T|, |C_2|/|T|, ..., |C_k|/|T|)$. Por lo tanto la información necesaria para identificar la clase de un elemento es

$$Info(T) = E(P) \tag{B.3}$$

Si particionamos T en base a los valores de un atributo no categórico X en subconjuntos $T_1, T_2, ..., T_k$ la información necesaria para identificar la clase de un elemento es el promedio ponderado de la información necesaria para identificar un elemento de T_i que expresamos como

$$Info(X, T) = \sum_{i=1}^{n} |T_i|/|T| * Info(T_i) \tag{B.4}$$

Ahora estamos en condiciones de definir la ganancia de información debido a un atributo X. La ganancia de información es la diferencia entre la

información necesaria para identificar un elemento de T y la información necesaria para identificar un elemento de T luego de obtenido el valor del atributo X.

$$Ganancia(X,T) = Info(T) - Info(X,T) \tag{B.5}$$

La noción de ganancia es utilizada para realizar un ranking de los atributos y con esta información construir el árbol eligiendo el atributo que mayor ganancia proporciona. Esta noción presenta una tendencia a favor de atributos con mayor cantidad de valores posibles, por lo tanto Quinlan sugiere usar el cociente de ganancia que se define como

$$CocienteDeGanancia(X,T) = \frac{Ganancia(X,T)}{InfoParticion(X,T)} \tag{B.6}$$

donde InfoParticion se calcula a partir de la partición $\{T_1, T_2, ..., T_m\}$ de T en base a X

$$InfoParticion(X,T) = E(|T_1|/|T|, |T_2|/|T|, ..., |T_m|/|T|) \tag{B.7}$$

B.9.2. C4.5 y C5.0

El algoritmo C4.5 es una mejora de ID3 propuesto por J.Ross Quinlan en 1993 [2] que permite trabajar con datos continuos. El árbol de decisión se construye por particionamiento recursivo de datos usando la estrategia depth-first (DEF). El algoritmo considera todas las posibles pruebas que pueden dividir el conjunto de datos y selecciona un conjunto de pruebas que dan la mayor ganancia de información. C4.5 maneja datos faltantes, datos continuos, propone una técnica de poda y las reglas pueden derivarse del árbol. Existen diversos métodos de poda propuestos por C4.5; uno de ellos es que un subárbol puede ser sustituido por una hoja si la sustitución resulta en un error similar al del árbol original. Otro método de poda es sustituir un subárbol por su subárbol más usado, también garantizando que se mantenga el error con esta sustitución. C4.5 usa un método de ventanas similar al descripto en ID3.

El algoritmo C5.0 es una versión comercial de C4.5 donde generación de reglas es diferente. Este algoritmo no es conocido públicamente pero es ampliamente usado en los paquetes comerciales como Clementine. El método de poda de C5.0 consiste en examinar el error en cada nodo y asumir que el error real es peor. Si N registros llegan a un nodo y E de ellos se classifican mal, el error en el nodo es E/N. El objetivo del algoritmo de crecimiento del árbol es minimizar el error, por lo tanto el algoritmo toma E/N como el mejor resultado que puede obtener. C5 usa muestreo estadístico para estimar el peor error que se puede tener en una hoja. Los datos de las hojas representan los resultados de una serie de intentos, cada uno de los cuales tiene uno de dos posibles resultados.C5 asume que el número de errores observados en los datos de entrenamiento es el más bajo del rango que se puede obtener y sustituye el valor tope para predecir el error de las hojas por E/N. Cuando la estimación del número de errores en un nodo es menor que la estimación de los errores de sus hijos, entonces los hijos se podan.

B.9.3. CART

El algoritmo CART [22], es un procedimiento de árboles de decisión creado en 1984 por Leo Breiman, Jerome Friedman, Richard Olshen, y Charles Stone de la Universidad de Berkeley y el Instituto Stanford. La técnica CART genera un árbol de decisión binario. Como en ID3 se usa la entropía como medida para elegir los mejores atributos y criterios de división. CART fuerza la definición de un orden de los atributos de separación. CART facilita el manejo de datos faltantes ignorando los registros en esta situación para los cálculos en esos atributos. CART contiene también su propia estrategia de poda. El modelo CART es un modelo de predicción que emplea técnicas estadísticas de regresión para construir árboles de decisión dicotómicos. Los conjuntos de datos de entrenamiento son usados como entrada para producir los árboles de decisión.

CART construye un árbol binario dividiendo los registros en cada nodo en función de una única variable de entrada. La primer tarea es entonces encontrar cual de los campos independientes hacen mejor la tarea de separación. La medida usada para evaluar un potencial separador es llamada diversidad. Un alto índice de diversidad indica una distribución igualitaria, mientras un índice bajo indica que el conjunto contiene una distribución

[22]CART abreviación de Classification and Regression Trees. En español árboles de clasificación y regresión

igualitaria de clases. Un buen separador es aquel que más disminuye la diversidad del conjunto de registros, en otras palabras, queremos maximizar la diferencia entre la diversidad antes de la separación y la diversidad de los hijos después de la separación. CART puede clasificar en variables continuas y categóricas.

CART introduce pruebas multivariadas, los árboles se construyen a partir de pruebas que involucran más de un atributo. Los árboles construidos de esta forma son a menudo árboles más exactos y más pequeños que los árboles construidos con pruebas univariadas; pero demoran más tiempo en generarse y son más difíciles de interpretar [33].

B.9.4. CHAID

J.A. Hartigan publicó CHAID[23] en 1975. Usando el test chi-cuadrado de significancia estadística, CHAID [34] automáticamente subagrupa los datos generando un árbol de decisión no binario. El test Chi-cuadrado se define como la suma de los cuadrados de la diferencia estándar entre la frecuencia esperada y observada de alguna ocurrencia en cada muestra. El test es una medida de la probabilidad de que una asociación aparente se deba al azar o, inversamente, que una diferencia observada entre muestras se deba al azar. El predictor que genera las agrupaciones más diferentes es elegido como el separador para el nodo actual.

Para desarrollar el árbol con CHAID, una variable continua debe ser dividida en un conjunto de rangos, CHAID también intenta parar el crecimiento del árbol antes de que ocurra overfitting. Como en los dos algoritmos anteriores CHAID busca una forma de usar las variables de entrada para dividir los datos de entrenamiento en dos o más nodos hijo. Los nodos hijo son escogidos de forma tal que la probabilidad de que el campo destino tome un valor u otro difiere entre un nodo y otro.

Una modificación al algoritmo CHAID básico, llamado CHAID exhaustivo, realiza combinaciones y pruebas más cuidadosas de las variables de predicción y requiere por lo tanto de más tiempo de cálculo. La combinación de categorías continúa hasta que quedan solo dos categorías para cada predictor. El algoritmo entonces procede a seleccionar la variable de partición, y selecciona entre los predictors el que rinde la partición más significativa.

[23] Chi-squared Automatic Interaction Detector. En español: detección automática de interacción Chi-cuadrado

Para conjuntos de datos grandes, y con muchas variables de predicción continuas , esta versión del algoritmo CHAID puede requerir tiempo de cálculo significativo.

B.9.5. SLIQ

SLIQ [24]fue desarrollado por IBM[35], es un clasificador por árboles de decisión diseñado para clasificar conjuntos datos de entrenamiento grandes. Este algoritmo maneja atributos numéricos y categóricos. Utiliza una técnica de preordenamineto en la fase de crecimiento del árbol, lo cual ayuda a evitar ordenamientos costosos en cada nodo.

SLIQ mantiene una lista ordenada para cada atributo continuo y una lista llamada lista de clases. Una entrada en la lista de clases corresponde a un dato. Cada entrada tiene una etiqueta de clase y un nombre del nodo al que pertenece en el árbol de decisión. Una entrada en la lista de atributos ordenados tiene un valor de atributo y el índice de los datos en la lista de clases.

SLIQ realiza el crecimiento del árbol de decisión en forma breadth-first. Para cada atributo, escanea la lista ordenada correspondiente y simultáneamente calcula la entropía para cada valor distinto de los nodos en la frontera del árbol de decisión. Luego de que los valores de entropía de cada atributo se calculan, un atributo es elegido para dividir cada nodo de la frontera y expandirlos para generar una nueva frontera. Luego se escanea una vez más la lista ordenada de atributos para actualizar la lista de clases para los nuevos nodos. Cuando los datos residentes en disco son demasiado grandes para caber en la memoria, SLIQ necesita que algunos datos se encuentren en memoria para poder trabajar. Esto crece en proporción directa al número de registros de entrada y pone un límite al tamaño de los datos de entrenamiento que puede manejar.

B.9.6. SPRINT

Muchos de los algoritmos vistos hasta el momento requieren que todos los datos permanezcan permanentemente en memoria; esto limita su utilidad para minar datos grandes. El algoritmo SPRINT [25] elimina todas las restricciones de memoria, es rápido y escalable. Fue diseñado para ser

[24]En español: cuestionario de aprendizaje supervisado
[25]En español: inducción paralelizable escalable de árboles de decisión

fácilmente paralelizable, permitiendo que muchos procesadores trabajen al mismo tiempo para construir un modelo único y consistente.

B.9.7. Comparación de algoritmos de árboles de decisión

Diferencias CART - C4.5 El algoritmo C4.5 es muy similar a CART; ambos tratan variables continuas en casi la misma forma. Algunas diferencias entre los dos son:

- Tratan las variables categóricas en forma diferente. Para atributos categóricos C4.5 produce una rama separada para cada valor, esto genera mucha apertura en el árbol, lo normal es asociar generando rangos de valores.Cuando C4.5 llega a un campo único como separador, su comportamiento por defecto es asumir que habrá una rama para cada valor que tome esta variable.

- CART produce árboles estrictamente binarios, en cambio C4.5 no está restringido a árboles binarios.

- Los métodos para medir homogeneidad de CART y C4.5 son diferentes [18].

Diferencias CART - C4.5 - CHAID La principal diferencia entre CART, C4.5 y CHAID es que CHAID se restringe a variables categóricas.

B.10. Redes neuronales artificiales

Las redes neuronales artificiales ofrecen un medio para tratar problemas de procesamiento no simbólico de la información, en contraste con los métodos de computación tradicionales. Los computadores tradicionales proporcionan formas eficientes de tratar información simbólica, mientras que las redes neuronales en sus paradigmas más sencillos no incluyen este aspecto. Las redes neuronales son un aporte de la inteligencia artificial A.2.2, pero su estudio y aplicación en la actualidad es parte de un campo multidisciplinario.

"Una red neuronal artificial es un sistema de procesamiento de información que tiene ciertas características en común con las redes neuronales biológicas. "[36]

Una red neuronal artificial, comunmente conocida sólo como red neuronal, es un conjunto interconectado de neuronas artificiales que imita el comportamiento de las neuronas biológicas. La idea básica es que cada unidad neuronal se alimenta de datos de entrada que son procesados para generar valores de salida [15].

De las innumerables definiciones de redes neuronales artificiales, la siguiente parece ser la que mejor se adapta a este análisis enfocado a la Minería de Datos.

"Una red neuronal es un modelo computacional con un conjunto de propiedades específicas, como son la habilidad de adaptarse o aprender, generalizar u organizar la información, todo ello basado en un procesamiento eminentemente paralelo. "[37]

Las redes neuronales tienen la habilidad de aprender a traves de ejemplos, de la misma forma que los humanos aprenden de la experiencia [15]. Las redes neuronales artificiales reproducen la capacidad humana de encontrar patrones y es por eso que en Minería de Datos pueden ser aplicadas a algoritmos de mapeo de patrones, problemas de clasificación, regresión y series de tiempo. Debido a que son sensibles a cambios en la representación de datos de entrada, diferentes representaciones pueden producir resultados diferentes. Por lo tanto, el formateo de los datos es una parte significativa del esfuerzo requerido en su uso.

Una red neuronal se caracteriza por :

- Arquitectura: patrón de conexiones entre las neuronas.
- Algoritmo de entrenamiento o aprendizaje: método para determinar los pesos de las conexiones. El peso es la información utilizada por la red neuronal para resolver el problema.
- Función de activación o transferencia: cada neurona posee un función que produce un resultado en base a las entradas que recibe. Cada neurona puede emitir una salida a la vez, pero esta salida puede variar según varían los valores de las entradas.

Por lo tanto es posible clasificar los diferentes tipos de redes neuronales en base a estas caracaterísticas [36].

Las redes neuronales tienen muchas ventajas [13]

- No linealidad. Aún cuando el procesamiento puede ser tanto lineal, como no lineal, los modelos de redes neuronales son básicamente no

lineales. Esta característica es muy importante, pues refleja los mecanismos reales de aprendizaje.

- Aprendizaje de los datos o aprendizaje de ejemplos, esta propiedad se conoce también como organización automática. Se utilizan algoritmos de entrenamiento para aprender automáticamente de la estructura de datos. No es necesario diseñar modelos previos.
- Flexibilidad. Le es posible manejar sin problemas variaciones en los datos de entrada, pueden diseñarse para tomar los parámetros de entrada en tiempo real.
- Tolerancia a fallos. Tiene capacidad para sobrellevar problemas debidos a redundancia de información, datos faltantes o ruido.
- Escalabilidad, se pueden escalar fácilmente a múltiples procesadores, lo cual es útil para conjuntos de datos almacenados en servidores multiprocesadores.
- Aplicación a problemas de tiempo real

Una de sus mayores cualidades es su amplio campo de aplicación, pueden ser adaptadas a una gran variedad de problemas de alta complejidad. Las redes neuronales modelan los problemas en función a las variables de entradas con las que se alimenta y las variables de salida que genera. Han sido satisfactoriamente aplicadas a una gran gama de problemas, incluyendo clasificación y aproximación de funciones. Son especialmente útiles porque no requieren de información previa de la distribución de los datos.

B.10.1. Arquitectura

Es común organizar las neuronas en capas, donde las neuronas de una misma capa tienen comportamientos similares. Para determinar el comportamiento de una neurona se consideran su función de activación y el patrón de conexiones por las cuales recibe y envía señales. Las capas de neuronas se conforman de una capa de entrada, una capa de salida y tantas capas ocultas como sean necesarias para representar la red. Para determinar el número de capas de una red neuronal no se tiene en cuenta la capa de entradas, porque en las neuronas que la componen no se realizan cálculos. Por esta razón es común que las redes se diferencien en redes de una capa y redes multicapa.

- Red de una capa.

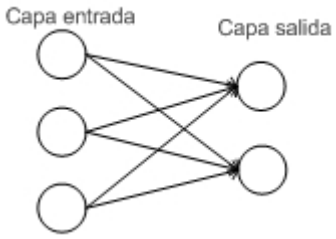

Figura B.12: Red neuronal de una capa.

- Red multicapa.

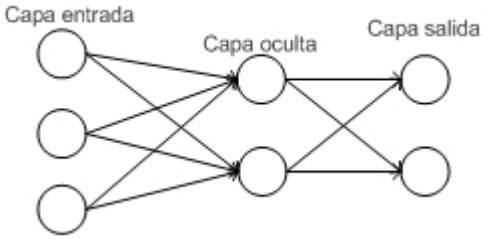

Figura B.13: Red neuronal multicapa.

- Red competitiva: las neuronas de salida compiten entre sí para activarse

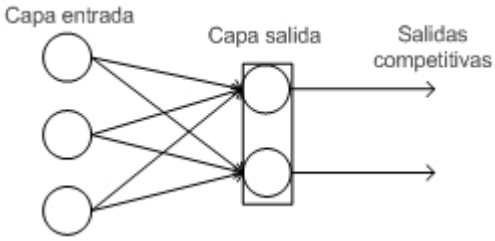

Figura B.14: Red neuronal competitiva.

Basándose en el patrón de conectividad las redes neuronales pueden clasificarse en dos tipos [13]

- Alimentación hacia adelante [26] : el procesamiento se propaga sólo desde la entradas a las salidas; sin bucles, ni retrocesos, ni conexiones entre neuronas que pertenecen a la misma capa.
- Recurrentes o retoalimentadas [27] : Si existen en la representación de la red neuronal aristas de retroalimentación que forman caminos circulares (ciclos) en la misma, la red se retroalimenta.

B.10.2. Algoritmo de entrenamiento o aprendizaje

" Aprendizaje es el proceso por el cual los parámetros libres de una red neuronal se adaptan a traves de un proceso de estimulación por el ambiente en que se encuentra embebida la red. El tipo de aprendizaje está determinado por la manera en la cual los parámetros cambian. " [13]

El proceso de aprendizaje de una red neuronal actualiza los pesos de las conexiones de tal manera que la red mejore la forma en que realiza la tarea para la que fue diseñada, por lo tanto puede ser visto como un proceso de optimización. La determinación de los pesos de las aristas es un proceso clave en una red neuronal, y para ello existe dos posibilidades, usar pesos fijos o pesos variables. Dadas estas dos posibilidades se pueden distinguir dos tipos de redes neuronales:

- Red de pesos fijos o estáticas.
- Red de pesos variables o dinámicas.

Redes estáticas. Este tipo de red neuronal, una vez establecido el valor de las entradas las salidas alcanzan un valor estacionario independientemente de las entradas en el instante anterior, y en un tiempo siempre por debajo de una determinada cota. Estas redes se pueden caracterizar estructuralmente por la inexistencia de bucles de realimentación y de elementos de retardo entre los distintos elementos de proceso que las forman. Debido a su modo de funcionamiento, estas redes tienen una capacidad limitada para sintetizar funciones dependientes del tiempo en comparación con las redes dinámicas. Ejemplo de red estática: debido su estructura las redes hacia adelante son estáticas.

[26] del término en inglés feedforward.
[27] del término en inglés feedback.

Redes dinámicas. Este tipo de redes responde de manera diferente ante diferentes secuencias de entradas, haciendo uso de manera implícita o explícita de la variable tiempo. Este aspecto las hace en principio más idóneas que las redes estáticas para la síntesis de funciones en las que aparezca de alguna manera el parámetro tiempo. La inclusión del elemento tiempo se puede llevar a cabo de varias maneras. Ejemplo de red dinámica: debido a su estructura las redes retroalimentadas son dinámicas.

A su vez las redes neuronales pueden utilizar distintas fuentes de información para actualizar sus pesos. Una opción es proporcionale información previamente validada y en función de esta se actualizan los pesos y otra es valerse unicamente de un conjunto de datos para esta tarea. Es por eso que se distinguen por lo menos dos formas de redes dinámicas según el método o información usados para determinar los pesos de las conexiones: aprendizaje supervisado y aprendizaje no supervisado. Como existe cierta ambigüedad en la clasificación de algoritmos en los tipos supervisado y no supervisado, algunos autores definen además un tercer tipo supervisado por sí mismo o híbrido. Finalmente se puede llegar a la siguiente clasificación:

- Red de aprendizaje supervisado: se proporcionan las resultados correctos para un conjunto de datos de entrenamiento y los pesos se determinan de forma que las salidas de la red neuronal sean tan cercanas a los resultados como sea posible.
- Red de aprendizaje no supervisado: la red explora un conjunto de datos de entrenamiento y determina patrones y relaciones existentes en los mismos.
- Red de aprendizaje híbrido: combinación de los dos anteriores.

El cálculo del error en los algoritmos de aprendizaje se realiza usando alguna de estas opciones

1. $\mid y_i - d_i \mid$
2. Media $\frac{(y_i - d_i)^2}{2}$

ALGORITMO DE REDES NEURONALES ARTIFICIALES : PROPAGACIÓN

Entrada: Red neuronal N,
X=$(x_1, ..., x_n)$, atributos de entrada
Salida: Y=$(y_1, ..., y_n)$, atributos de salida
Algoritmo:
for each nodo i en la capa de entrada do
salida x_i en cada arco de salida de i;
for each capa oculta do
for each nodo i do
$S_i = \sum_{j=1}^{k}(w_{hi} * x_{hi})$;
for each arco desde i do
$salida = \frac{1-e^{-S_i}}{1+e^{-c*S_i}}$;
for each nodo i en la capa de salida do
$S_i = (\sum_{j=1}^{k}(w_{ji} * x_{ji}))$;
$salida = \frac{1}{1+e^{-c*S_i}}$;

Figura B.15: Pseudocódigo algoritmo de propagación para RN.

ALGORITMO DE REDES NEURONALES ARTIFICIALES : APRENDIZAJE SUPERVISADO

Entrada: Red neuronal N,
X=(x_1, ... ,x_n) , registro de entrada de los datos de entrenamiento
D=(d_1, ... ,d_n) , registro de salida esperado
Salida: Red neuronal N , red neuronal mejorada
Algoritmo:
Propagar X a través de N produciendo la salida Y(y_1, ... ,y_n);
Calcular el error comparando D con Y;
Modificar los pesos de los arcos de N para reducir el error;

Figura B.16: Pseudocódigo algoritmo de aprendizaje supervisado para RN.

ALGORITMO DE REDES NEURONALES ARTIFICIALES : PROPAGACIÓN HACIA ATRÁS

Entrada: Red neuronal N,

X=(x_1, ...,x_n) ,registro de entrada de los datos de entrenamiento

D=(d_1, ...,d_n) ,registro de salida esperado

Salida: Red neuronal mejorada N

Algoritmo:

//Propagar X a través de N produciendo la salida Y

Propagar(N,X);

//Calcular el error comparando D con Y usando media al cuadrado

$E = \frac{1}{2}\sum_{i=1}^{m}(d_i * y_i)^2$;

//Modificar los pesos de los arcos de N para reducir el error encontrado

//un conjunto de pesos que minimicen el error medio al cuadrado.

Gradiente(N,E)

Figura B.17: Pseudocódigo algoritmo de propagación hacia atrás para RN.

ALGORITMO DE REDES NEURONALES ARTIFICIALES : GRADIENTE

Entrada: Red neuronal N,

E, error encontrado en el algoritmo para atrás

Salida: Red neuronal mejorada N

Algoritmo:

for each nodo i en la capa de salida do

for each nodo j entrante a j do

$\Delta w_{ji} = \eta(d_i - y_i)x_j(1 - (\frac{1}{1+e^{-Sj}}))\frac{1}{(1+e^{-Sj})}$;

$w_{ji} = w_{ji} + \Delta w_{ji}$;

capa = capa anterior;

repeat

for each nodo i en esta capa do

$\Delta w_{ji} = \eta\sum_{l=1}^{n}(d_i - y_i)w_{il}x_j\frac{1-(\frac{1-e^{-Si}}{1+e^{-Si}})^2}{2}$;

$w_{ji} = w_{ji} + \Delta w_{ji}$;

capa = capa anterior;

until (capa = capa entrada)

Figura B.18: Pseudocódigo algoritmo del gradiente para RN.

B.10.3. Funciones de activación o de transferencia

La operación básica de una neurona está determinada por su función de activación o transferencia. Para hacer que una red neuronal realice una tarea específica, es necesario elegir como se conectan las neuronas entre sí y determinar los pesos de las conexiones de forma apropiada. Las conexiones determinan si una neurona influencia a otra, los pesos de las conexiones establecen el valor de esa influencia.

/medskip La función de activación pertence generalmente a una de las siguientes clases:

- Función lineal.
- Función umbral.
- Función sigmoid. F.3

A continuación se describen algunas funciones comunmente usadas.

- Función identidad.

$f(x) = x \forall x$ Para las entradas la función de activación es la identidad, puede darse también en neuronas internas de la red.

- Función binaria.

$$f(x) = \begin{cases} 1 & x \geq \alpha \\ 0 & x < \alpha \end{cases}$$

- Función binaria sigmoid. F.3

$f(x) = \frac{1}{1+exp(\theta x)}$

- Función bipolar sigmoid. F.3

$g(x) = 2 * f(x) - 1 = \frac{2}{1+exp(\theta x)} - 1$ La función bipolar sigmoid está relacionada con la tangente hiperbólica, que se usa en casos en que el rango de salida esta entre -1 y 1.

B.10.4. Ejemplos de redes neuronales

Perceptrón multicapa

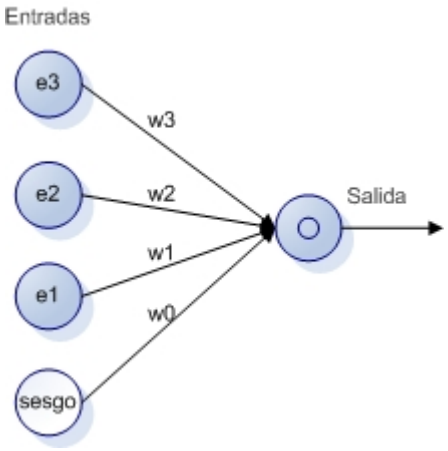

Figura B.19: Perceptrón original.

El perceptrón multicapa (MLP [28]) es uno de los tipos de redes neuronales más importantes y uno de los más populares en aplicaciones de redes neuronales a la realidad. El modelo clásico fue descrito por Rosenblatt en 1958 [38] junto con un teorema de aprendizaje para el perceptrón. En esta primer versión la salida del perceptrón es una única neurona que cuenta con una función lineal a partir de las entradas. B.19 El perceptrón multicapa es una red construída usando como componentes perceptrones y conectándolos de forma jerárquica, esto permite representar funciones no lineales de salida. [33] La red consiste entonces de un conjunto de entradas (capa de entrada), una o más capas de ocultas, y una capa de salida. El procesamiento se realiza hacia adelante capa por capa. [13]

Características de un perceptrón multicapa son:

- Cada neurona tiene asociada una función de activación generalmente no lineal, puede ser sigmoidal o hiperbólica.
- La red contiene una o más capas ocultas.
- Alto nivel de conectividad de una capa a la siguiente.

[28] MLP: Siglas en inglés para multilayer perceptron

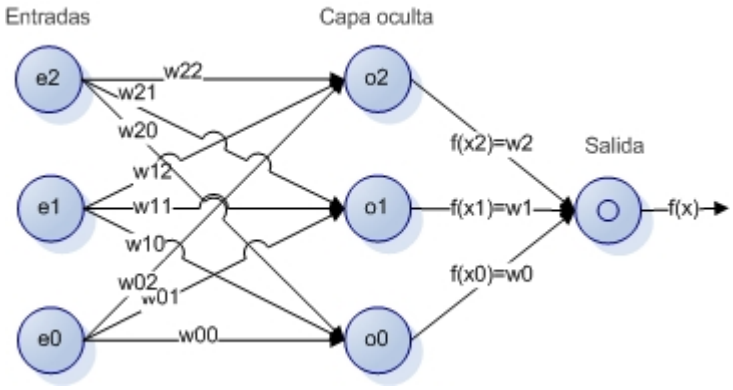

Figura B.20: Perceptrón multicapa.

Mapas auto-organizados

El modelo de red neuronal de mapa auto-organizado [29] fue propuesto por T.Kohonen en 1982 [39]. A ganado popularidad debido a su capacidad para representar de manera automática y eficiente conjuntos de datos multidimensionales por medio de una estructura bidimensional.

La arquitectura del SOM está constituída por un conjunto de neuronas N con las mismas propiedades, conectadas de forma idéntica a la entrada. Durante el proceso de entrenamiento la entrada se considera como una variable de tiempo discreto que toma valores del conjunto de entradas X. Las neuronas se distribuyen en una retícula bidimensional, donde cada una constituye un nodo de la retícula. La localización de la neurona en la retícula está dada por un vector de localización $r_i = (p_i, q_i)$ y cada neurona tiene asociado un vector de referencia w_i (vector de pesos).

[29] Conocido por sus siglas en inglés SOM, Self Organizing Map.

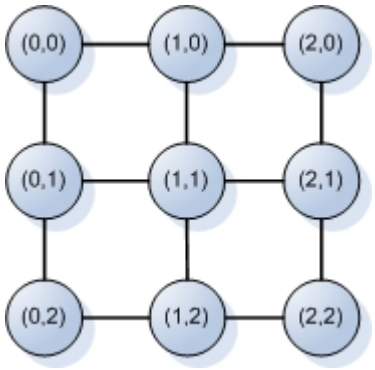

Figura B.21: Mapa auto organizado.

El algoritmo SOM es un algoritmo aprendizaje no supervisado. El entrenamiento se realiza mediante un proceso conocido como *aprendizaje competitivo*, donde las neuronas se hacen sensibles gradualmente a diferentes datos de entrada.

B.11. Algoritmos genéticos

Los algoritmos genéticos fueron propuestos por primera vez en 1975 por John Holland [40]. Los algoritmos genéticos son un ejemplo de métodos de computación evolutiva, su nombre viene de la similitud con el proceso natural de selección. Dada una población de soluciones individuales potenciales de un problema, la computación evolutiva expande la población con nuevas soluciones que son potencialmente mejores. La base de los algoritmos de computación evolutiva es la evolución biológica donde a medida que el tiempo pasa produce la mejor adaptación de los individuos. El objetivo es encontrar una solución óptima de muchas soluciones posibles basándose en un criterio definido.

Los algoritmos se basan en tres mecanismos que tienen relación con la selección genética. El primero preselecciona las soluciones más fuertes. El segundo es similar a cruzamiento donde son generadas nuevas soluciones mezclando aspectos de los conjuntos padres. El tercero es similar a la mutación donde accidentalmente se generan cambios en la población. Luego de que un número de nuevas generaciones de soluciones son generadas se identifica una solución que no puede ser mejorada. Esta solución es tomada como la solución final. Los algoritmos genéticos tienen dos puntos débiles.

El primero es que la formulación del problema priva de toda oportunidad de estimar el significado estadístico de la solución obtenida. El segundo es que se necesita un especialista para formular el problema en forma efectiva. Los algoritmos genéticos son un instrumento para la investigación científica más que una herramienta para el análisis de datos.

Un algoritmo genético es un modelo computacional que consiste de cinco componentes:

- un conjunto inicial de individuos P
- una técnica de cruzamiento, que indica como se reproducen los individuos
- un algoritmo de mutación, que como en la naturaleza indica cambios en las características de los individuos
- una función de aptitud, que determina los mejores individuos del conjunto
- algoritmos para aplicar la técnica de cruzamiento y el algoritmo de mutación al conjunto de individuos P de forma iterativa y elegir mediante la función de aptitud los mejores individuos del conjunto P. El algoritmo reemplaza un número determinado de individuos en cada iteración y termina cuando se alcanza el thresold.

```
Algoritmo Genético
  Entrada: Población actual P
  Salida: Población mejorada Q
  Algoritmo:
  Repetir
     Evaluar la aptitud de los individuos de un subconjunto de la población
     Seleccionar pares de los mejores individuos para reproducir
     Aplicar cruzamiento
     Aplicar mutación
  hasta (condicion fin)
```

Figura B.22: Pseudocódigo algoritmo genético.

Cuando se usan algoritmos genéticos para resolver problemas, la primera tarea a realizar es determinar como se puede modelar el problema como un conjunto de individuos. La abstracción del problema para llegar a este tipo de representación es una de las tareas más compleja dentro del proceso. Como el

espacio de búsqueda es muy grande, incluso infinito, los algoritmos genéticos probablemente no encuentren la mejor solución. Los algoritmos podan del espacio de individuos aquellos que no resuelven el problema y por otra parte sólo generan nuevos individuos diferentes a los ya examinados. La mayor ventaja de los algoritmos genéticos es que son fácilmente paralelizables. Las desventajas son que en general son difíciles de entender y más aún de explicar a los usuarios, la abstracción de los problemas es compleja y no siempre es posible, es difícil definir las funciones de aptitud, cruzamiento y mutación.

Los algoritmos genéticos se han usados esencialmente como una técnica para solucionar problemas de combinación o de optimización. En Minería de Datos los algoritmos genéticos son usados para clustering, predicción y reglas de asociación. La técnica consiste en encontrar el modelo más adecuado de un conjunto de modelos que representan los datos. Se elige un modelo para iniciar el proceso y luego a medida que se avanza en las iteraciones, se combinan los modelos para crear nuevos. El mejor modelo es usado en la siguiente iteración. Los diferentes algoritmos genéticos usados en Minería de Datos se diferencian en la forma de representación del modelo, la forma de combinación de los individuos y la función de aptitud.

B.12. Comparación de algoritmos

El clustering es similar a la clasificación en el sentido en que los datos son agrupados. El clustering pertenece a aprendizaje no supervisado, donde las clases no están predefinidas, es dirigido a los datos y se basa en las similitudes entre los valores de los atributos. La clasificación pertenece al aprendizaje supervisado, las clases en que se agrupan los datos están predefinidas.

Los árboles de decisión son modelos simples, de fácil interpretación y generación en general rápida, además no dependen de hipótesis realizadas sobre los atributos. Los métodos basados en árboles de decisión tienden a ser más robustos que la mayoría de los métodos estadísticos. Por otra parte necesitan de un mayor volumen de datos para usar en el entrenamiento, necesidad que no debe ser subestimada al momento de elegir este tipo de algoritmos [13].

B.13. Uso combinado de los algoritmos

Técnicas de aprendizaje supervisado y no supervisado se usan comúnmente asociadas para producir mejores resultados. Las técnicas de clustering por ejemplo rara vez se utilizan en forma aislada. Encontrar clusters es por lo general forma parte de la búsqueda de la solución de un problema, pero es poco común que sea el objetivo final. Un caso excepcional es la identificación de segmentos de mercado, en este caso la identificación de los clusters es el objetivo final. En general se usa la técnica de clustering para descomponer un problema en subproblemas a los cuales se les aplicará técnicas de aprendizaje supervisado, esto ayuda a mejorar los resultados que se obtendrían si se aplicarán desde el inicio estás técnicas sobre el conjunto completo de datos.

Es posible utilizar un algoritmo del árbol de regresión para proporcionar el pronóstico financiero y un algoritmo basado en reglas (un algoritmo CART) para el análisis de carrito de compras. También se puede utilizar el algoritmo de árbol de decisión como manera de reducir el número de columnas en un conjunto de datos. Se puede utilizar un algoritmo de agrupación (un algoritmo de reconocimiento de patrones) para descomponer los datos en grupos más o menos homogéneos, y después utilizar los resultados para crear un modelo mejor para aplicar un algoritmo de árboles de decisión.

Apéndice C

Herramientas de Minería de Datos

C.1. Introducción

El mercado de productos de software de Minería de Datos es muy diverso, en él existen varias categorías de productos de software: herramientas de minería de datos genéricas, paquetes de Business Intelligence [1] extendidos, productos independientes especializados y soluciones propietarias. A pesar de cierta consolidación, el espacio de productos de Minería de Datos todavía se encuentra altamente fragmentado. La diversidad de propuestas complica perceptiblemente el proceso de planeamiento y de compra de los interesados. Una genuina herramienta de Minería de Datos debería soportar el descubrimiento semiautomático de patrones. Los usuarios deben informarse tanto de los requisitos como de las ventajas de cada una de las herramientas disponibles antes de decidirse por una para su proyecto de Minería de Datos. Por supuesto que además de las consideraciones técnicas de las herramientas es necesario realizar consideraciones financieras. Cuando una institución pretende llevar a cabo un proyecto de Minería de Datos, habitualmente uno de los factores de mayor peso a tener en cuenta es el costo que esto implica . En el mercado existen variaciones enormes en los precios de las herramientas de minería de datos que va desde software libre hasta productos que rondan los miles de dólares.

Para que la Minería de Datos sea efectiva, es necesario contar con software confiable, una computadora suficientemente potente y buenos datos.

[1] Ver definición en Glosario G

Incluso contando con buenas herramientas de software, la Minería de Datos no van a ser exitoso si no se tienen buenos datos. En lo que refiere a la Minería de Datos, el viejo dicho *"garbage in, garbage out"* [2] se aplica muy bien. En lo que concierne al usuario, su participación es fundamental; es necesario que posea conocimientos mínimos de análisis estadístico y de los algoritmos que el software utiliza. Muchas de las herramientas de software para Minería de Datos no son amigables; son herramientas de gran alcance pero requieren que el usuario tenga amplios conocimientos estadísticos y de los algoritmos que se utilizan.

C.2. Clasificación de herramientas

Con el objetivo de apoyar en el proceso de decisión de las organizaciones que buscan un producto de software para sus proyectos de Minería de Datos, Gartner Group [10] ha propuesto la siguiente clasificación de herramientas:

- Herramientas genéricas
- Herramientas para algoritmos específicos
- Herramientas específicas según su aplicación
- Herramientas embebidas
- Herramientas de programación analítica
- Consultoría externa en Minería de Datos

A continuación se describen en detalle cada una de ellas [3] y se analiza en cada una de las categoría alguna herramienta representantiva.

C.2.1. Herramientas genéricas

Las herramientas genéricas son la clase más conocida. Una de las principales ventajas de este tipo de herramientas es que permiten tratar muchos tipos de problemas y ofrecen una amplia gama de técnicas de Minería de Datos y de visualización de los datos. Su flexibilidad y facilidad de uso les permiten alcanzar buenos resultados dentro de un marco de tiempo razonable. Se podría considerar una desventaja el que requieran ser operados por personal experto, que sepa como preparar los datos, cuales técnicas son

[2] En español: "basura entra, basura sale".
[3] Excepto la categoría "Consultoría externa", que no es de interés para este estudio.

las más apropiadas para cada problema y cómo utilizarlas, validar los resultados y, finalmente, cómo aplicarlos al negocio. Todo depende del usuario destinado a usar la herramienta.

Dentro de esta categoría de herramientas se encuentran los siguientes productos:

- Clementine®[41], de SPSS®[42]
- Enterprise Miner™[43], del Instituto SAS®[44]
- Insightful Miner™de Insightful.
- WEKA[45], de la Universidad de Waikato [46], Nueva Zelanda

Clementine®de SPSS®

Producto:	Clementine®
Vendedor:	SPSS®
Funciones:	Reglas de asociación, clasificación, clustering, factor analysis, predicción, descubrimiento de secuencias
Técnicas:	Reglas de asociación(Apriori), BIRCH, CARMA, arboles de decisión (C5.0, C&RT (variación de CART), CHAID, CHAID exhaustivo, QUEST), reglas de decisión (C5.0, GRI), clustering K-medias, redes neuronales (Kohonen, MLP, RBFN), regresión (lineal, logística y logística multinominal)
Plataformas:	Cliente Windows, Servidor: Windows Server™, Sun™Solaris™, HPUX , IBM AIX®, OS/400
Bases de datos:	Oracle, Ingres, Sybase, Informix, SQL Server, DB2.
WebMining:	Si
Versión actual:	9
Precio :	desconocido

Clementine®es el primer producto comercial de Minería de Datos del mercado, fue lanzado en 1994 por la empresa Integral Solutions Ltd., que en 1998 fue adquirida por SPSS®. Las herramientas de SPSS®comenzaron

como paquetes estadísticos y Clementine®fue el primer paquete de Minería de Datos en usar programación gráfica. Clementine®soporta el proceso completo de Minería de Datos a través del modelo CRISP-DM A.7.1, lo cual permite acortar los tiempos de obtención de una solución. Proporciona amplia gama de técnicas de Minería de Datos y soluciones verticales pre-construidas, de forma integrada y comprensiva, da un foco especial en la visualización y facilidad de uso. Clementine®es un ambiente de trabajo integrado para Minería de Datos, permite desarrollar modelos de predicción usando la experiencia que el usuario posee del negocio y poner esa experiencia en uso para mejorar la toma de decisiones.

La siguiente es una evaluación de las principales características de Clementine según Bloor Research [47].

- Apoya la metodología de CRISP-DMA.7.1. El modelo CRISP-DM hace que el proceso de Minería de Datos sea un proceso de negocios, enfocando la tecnología de Minería de Datos a resolver problemas de negocio específicos.
- Soporta amplia gama de algoritmos de que se pueden utilizar individualmente o combinados.
- El uso es altamente intuitivo, incorpora técnicas de programación visual para definir procesos y para probar ideas. El "constructor" es relativamente simple y la visualización estadística y gráfica es fácil de entender e interpretar.
- Los usuarios no tienen que saber como trabajan los algoritmos, apenas deben saber que hace. Igualmente no es fácil para los principiantes y para ellos disponen de ayuda en línea muy completa.
- La conectividad con bases de datos que ofrecen incluye a los manejadores más usados del mercado. Permite generar SQL nativo para cada base de datos, de manera transparente para el usuario.

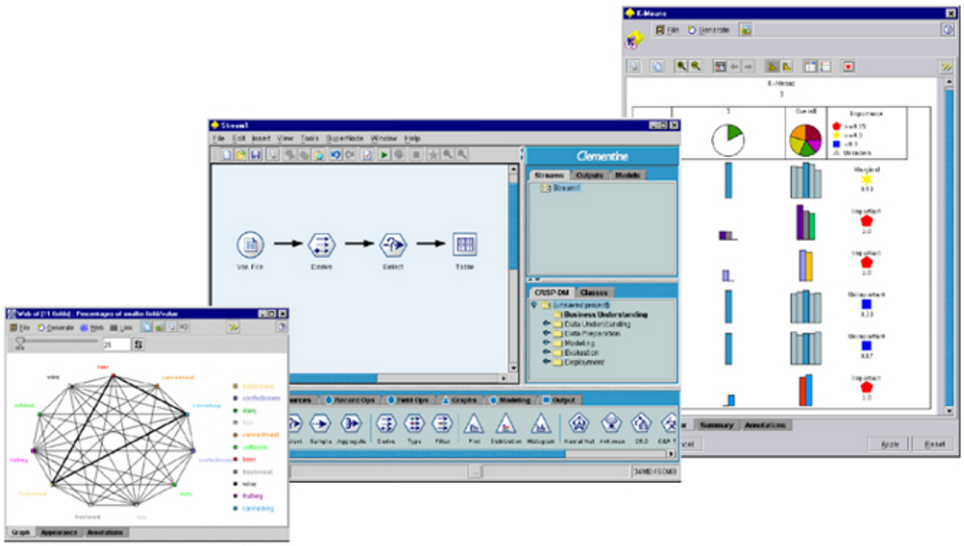

Figura C.1: Clementine.

Enterprise Miner™del Instituto SAS®

Producto:	Enterprise Miner™
Vendedor:	Instituto SAS®
Funciones:	Reglas de asociación, clasificación, clustering, predicción, series de tiempo
Técnicas:	Arboles de decisión (CART, CHAID), K vecinos más cercanos, regresión (lineal y logística), razonamiento basado en memoria, redes neuronales (Kohonen, MLP, RBF,SOM)
Plataformas:	Cliente Windows, Servidor Unix o Windows
Bases de datos:	
Versión actual:	5.1
Precio :	desconocido

Enterprise Miner™provee funcionalidades para apoyan todas las tareas necesarias en la realización de Minería de Datos, desde al acceso a los datos hasta la puesta en producción. Soporta el proceso completo de Minería de

Datos usando su propio modelo de proceso llamado SEMMA A.7.2. La interface gráfica basada en iconos, crea un flujo de proceso que representa la tarea de Minería de Datos a realizar. Además, contiene muchas herramientas para empaquetamiento, muestreo y visualización, puesta en producción, transformaciones, y validación del modelo. Genera una fórmula de evaluación para cada etapa del desarrollo del modelo.

Insightful Miner^{TM}de Insightful

Producto:	Insightful Miner^{TM}[48]
Vendedor:	Insightful [49]
Funciones:	Clasificación, clustering, predicción, series de tiempo
Técnicas:	Arboles de decisión, redes neuronales, Naive Bayes, regresión (lineal y logística), k-medias, Cox
Plataformas:	Cliente Windows, Servidor Windows o Solaris
Bases de datos:	Oracle, DB2, SQL Server, Sybase, archivos ASCII y otro formatos planos, Excel, SAS, acceso a otras bases de datos via ODBC
Versión actual:	3.0
Precio :	desconocido

Insightful Miner^{TM}es un ambiente integrado para Minería de Datos, proporciona acceso simple a los datos, técnicas de visualización en 2D y 3D, técnicas para la manipulación y transformación de datos. La integración con S-PLUS®[4] le da mucho potencial en cuanto a funciones disponibles y en cuanto a potencial extensibilidad. Las funciones disponibles a través de S-PLUS®son

- Regresión no lineal y spline
- Combinación de modelos lineales y no lineales

[4]S-PLUS producto de análisis estadístico de Insightful

- Regresión ribusta y análisis de varianza
- GAMs (Generalized Additive Models)
- Modelos de series de tiempo

Las funcionalidades pueden ser extendidas a través de S-PLUS®Script, mediante el cual es posible diseñar funciones propias.

Insightful Miner^{TM}presenta un ambiente de trabajo visual en forma de WorkFlow [5], que simplifica y acelera el proceso de análisis de datos. La interface estilo drag-and-drop le da facilidad de uso. Una de las principales ventajas de este formato es que el proceso queda documentado en forma simple a medida que se construye. Los flujos diseñados pueden almacenarse para ser usado como platillas en diseños futuros y pueden compartirse con otros usuarios, permitiendo la realización de modificaciones simultáneas de los modelos. Otra característica importante consiste en la programación de los procesos de minería para su ejecución automática.

[5]

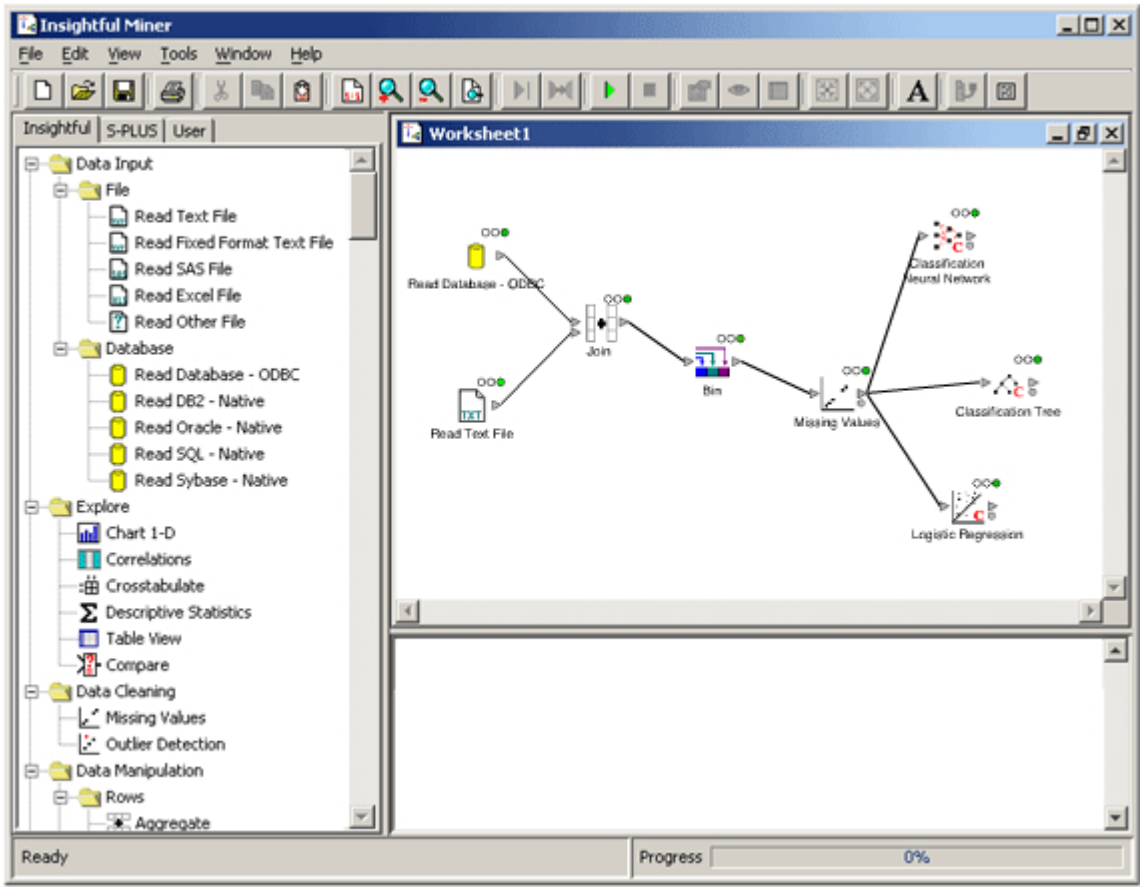

Figura C.2: WorkFlow de Insightful Miner.

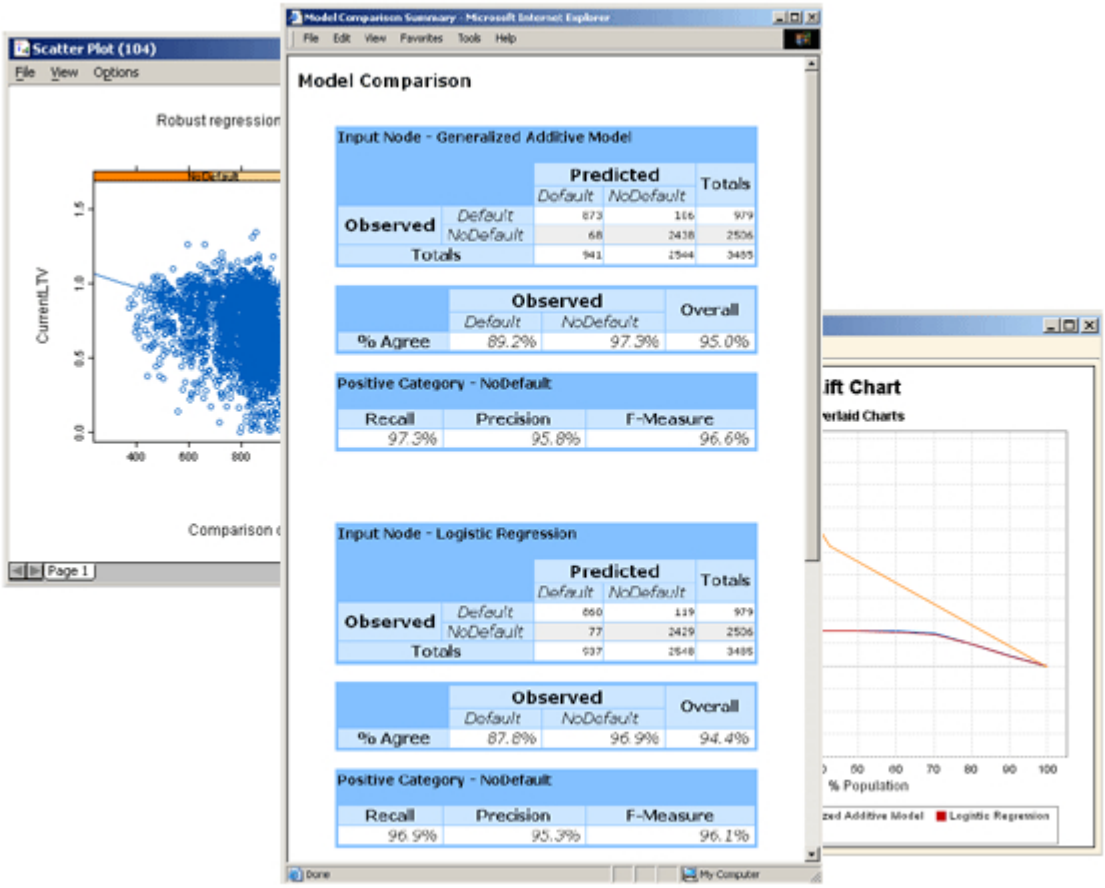

Figura C.3: Presentación de resultados de Insightful Miner.

WEKA de la Universidad de Waikato

Producto:	WEKA
Vendedor:	Universidad de Waikato
Funciones:	Clustering, clasificación, reglas de asociación, selección de atributos, visualización
Técnicas:	Clustering k-media , Clasificación (Bayes, Naive Bayes y otros), reglas de asociación (A priori y otros)
Plataformas:	todas las que dispongan de máquina virtual Java
Bases de datos:	
Versión actual:	3.4.5
Precio :	gratis

WEKA[45], Waikato Environment for Knowledge Analysis, es una colección de algoritmos de aprendizaje de máquina para tareas de Minería de Datos desarrollado en el marco del "Proyecto Weka de aprendizaje de máquina" de la Universidad de Waikato[46]. Los objetivos del proyecto son hacer accesibles las técnicas de aprendizaje de máquina, aplicarlas a problemas de interés para la industria, desarrollar nuevos algoritmos de aprendizaje de máquina y hacerlos públicos y contribuir al desarrollo de un marco teórico para el campo del aprendizaje de máquina. Weka es software de código abierto[6] publicado bajo de la licencia pública general de GNU. El software está escrito enteramente en Java e incluye un interfaz uniforme a un número de técnicas estándares de aprendizaje de máquina. Java asegura su portabilidad a múltipes plataformas, Weka ha sido probado en sistemas operativos Linux, Windows y Macintosh. Los algoritmos reunidos en WEKA se pueden aplicar directamente a un conjunto de datos desde línea de comando o llamándolo desde código Java. WEKA contiene herramientas para el proceso de pre-procesamiento de los datos, clasificación, regresión, agrupación, reglas

[6]del inglés Open Source, se refiere al software cuyo código fuente está disponible públicamente. Los términos de licenciamiento del mismo pueden variar.

de asociación, y visualización. Además está bien adaptado para desarrollar nuevos esquemas de aprendizaje de máquina.

Existen varios niveles en los cuales WEKA puede ser utilizado. Una forma es aplicar un método de aprendizaje a un conjunto de datos y analizar su resultado para extraer información sobre los datos. Otra forma es aplicar a varios métodos de aprendizaje y comparar su funcionamiento para elegir uno para la predicción. La implementaciones de esquemas de aprendizaje reales y las herramientas para preprocesar los datos son los recursos más valiosos que proporciona WEKA. Si un desarrollador desea programas sus propios algoritmos, WEKA proporciona bibliotecas para resolver temas comunes como la lectura de los archivos de datos, implementar filtros, evaluar resultados. Para hacer uso eficiente de estas utilidades el programador debe familiarizarse con las estructuras básicas de datos. Un recurso importante de WEKA es la documentación en línea, que se genera automáticamente del código de fuente y refleja su estructura. La documentación es esencial si se desea pasar al siguiente nivel y acceder a la biblioteca desde programas Java desarrollados por el usuario, o escribir y probar esquemas propios.

La ventana de navegación de WEKA se utiliza para ejecutar los 4 ambientes gráficos C.4. 1. CLI Simple. Proporciona una línea de comandos simple que permite la ejecución directa de los comandos de WEKA para sistemas operativos que no proporcionan su propia línea de comando. 2. Explorador. Es un ambiente para explorar datos. 3. Experimentador. Es un ambiente para realizar experimentos y conducir pruebas estadísticas entre esquemas de aprendizaje. 4. Flujo de conocimiento. Este ambiente es similar al Explorador pero con un interfaz drag-and-drop.

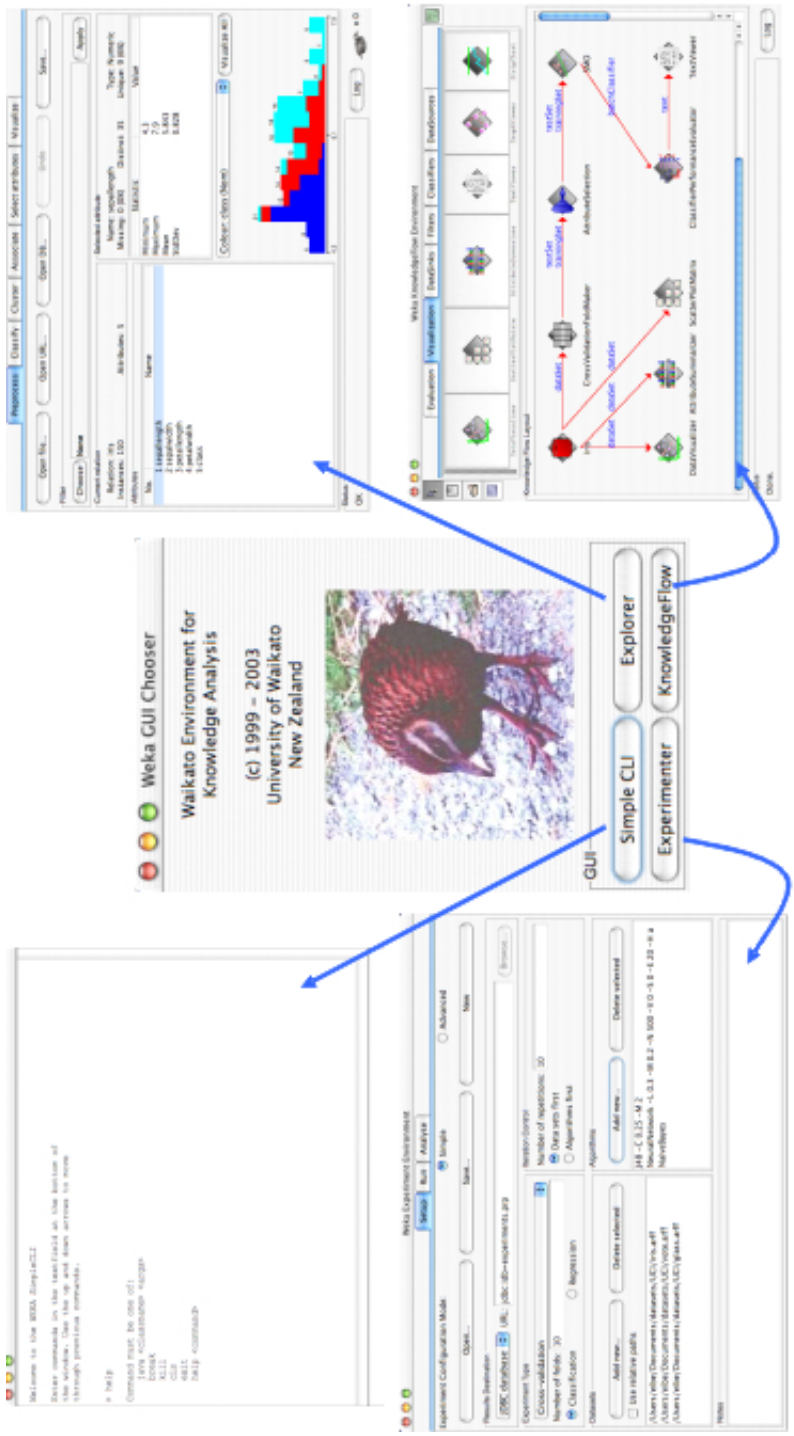

Figura C.4: Imágenes de los componentes de WEKA.

Todas las tareas que componen el proceso de Minería de Datos pueden ser llevadas a cabo por el usuario sin escribir ni una línea de código. La interfase gráfica presenta todas las opciones disponible en forma de botones y menúes. Debido a la inmensidad de opciones puede resultar poco intuitiva para los principiantes; es conveniente tener a mano el tutorial y la guía del explorador ya que no tiene ayuda en línea. También está disponible en el sitio de WEKA una guía paso a paso en forma de diapositivas que resulta muy útil para las primeras pruebas.

C.2.2. Herramientas para algoritmos específicos

Las herramientas para algoritmos específicos son similares a las herramientas genéricas pero se enfocan a un subconjunto particular de algoritmos. Su ventaja respecto a las anteriores es que pueden ofrecer mejores resultados que las herramientas genéricas si las técnicas son las adecuadas al problema de minería de datos abordado. Algunas de estas herramientas se especializan en poner en ejecución lo más eficientemente posible algoritmos específicos y su funcionamiento se degrada si los algoritmos puestos en ejecución no son apropiados.

Dentro del grupo de herramientas para algoritmos específicos se pueden distinguir dos clases: para árboles de decisión y para redes neuronales.

Dentro de la categoría de árboles de decisión se encuentran las siguientes herramientas:

- CART®[50] y MARS®[51] de Salford Systems [52]
- Alice[53] de Isoft[54]

Dentro de la categoría de redes neuronales se encuentran las siguientes herramientas:

- NeuralWorks Professional II de NeuralWare

Las consideraciones más importantes al elegir una herramienta de este tipo es que la misma se adapte correctamente a las condiciones de cada algoritmo. Si no se eligen adecuadamente los resultados serán pobres y sin valor. Los algoritmos de árboles de decisión tienen resultados pobres en problemas de dominio lineal o casi lineal. Los algoritmos de redes neuronales son pobres para tareas de segmentación. Los modelos K vecinos cercanos tienen comportamiento pobres en problemas de dominio de grandes dimensiones y más aún ante la presencia de valores altos de ruido.

CART®

Producto:	CART®[50]
Vendedor:	Salford Systems [52]
Funciones:	
Técnicas:	Arboles de decisión CART
Plataformas:	
Bases de datos:	archivos de texto
Versión actual:	5
Precio :	desconocido

CART [7], es una herramienta que permite aplicar técnicas de árboles de decisión automáticamente a bases de datos grandes, con el fin de encontrar patrones y relaciones significativos. Este conocimiento se utiliza luego para generar modelos de predicción confiables. Por lo tanto, CART sirve como método de pre-procesamiento para otras técnicas de análisis de datos. Por ejemplo, las salidas de CART se pueden utilizar como entradas para mejorar la exactitud de algoritmos de redes neuronales y regresión logística.

La metodología CART resuelve algunos problemas comunes a los algoritmos de árboles de decisión relacionados a su precisión y operación. Las innovaciones propuestas por CART son

- resuelve el problema de crecimiento del árbol
- utiliza únicamente la división binaria del árbol
- incorpora testeo y validación del árbol automática
- provee métodos para manejas valores faltantes

MARS®

Producto:	MARS®[51]
Vendedor:	Salford Systems [52]
Funciones:	Clasificación
Técnicas:	Arboles de decisión CART
Plataformas:	CMS, MVS, Unix, Linux y Windows
Bases de datos:	
Versión actual:	
Precio :	desconocido

[7] Classification and Regression Trees

MARS®, es una herramienta que permite desarrollar fácilmente modelos que proporcionen pronóstico exactos. MARS automatiza el desarrollo y puesta en producción de modelos de regresión.

Alice de Isoft

Producto:	Alice [53]
Vendedor:	Isoft [54]
Funciones:	Clustering, correlación, segmentación
Técnicas:	Arboles de decisión
Plataformas:	Versión standalone para Windows NT/98/2000/2003/ XP, TSE, Metaframe. Versión Cliente-Servidor para: Windows NT/2000/2003 o Solaris
Bases de datos:	
Versión actual:	6.1
Precio :	desconocido

C.2.3. Herramientas específicas según su aplicación

Las herramientas específicas según aplicación ofrecer un ambiente adaptado para requerimientos particulares según su uso. Existen múltiples casos para marketing, comercialización y CRM. Su fortaleza consiste en proveer al usuario la guía para llevar adelante un proyecto de Minería de Datos a través de un asistente y diálogos de preguntas y respuestas. Debido a su ambiente especializado, el usuario necesita menos experiencia. Sin embargo, al utilizar estas herramientas, las empresas deben tener objetivos claros que coincidan con las herramientas elegidas para que los resultados sean útiles.

Dentro de esta categoría de herramientas están los siguientes productos:

- IBMs Intelligent Miner para mercadeo
- Unicas Model 1
- SLP InfoWares Churn/CPS
- Quadstones Decisionhouse para CRM

C.2.4. Herramientas de Minería de Datos embebidas

Las herramientas de Minería de Datos embebidas se pueden encontrar en productos como manejadores de bases de datos y paquetes de Business Intelligence [8] . La Minería de Datos se puede beneficiar utilizando la estructura de cubos diseñada para los productos OLAP como plataforma para el diseño de una solución o como fuente de datos. Es así como muchos proveedores de herramientas OLAP han diseñado soluciones para Minería de Datos embebidas en sus productos. Las ofertas embebidas para Minería de Datos son aún triviales, la mayoría de estas herramientas ofrecen funcionalidades limitadas. Cuentan con la gran ventaja de que son mucho más fáciles de utilizar que las herramientas genéricas, lo que hace a esta clase particularmente popular para usuarios menos especializados o usuarios que están familiarizados con las herramientas originales. Como desventajas se destaca su poca flexibilidad, la cual puede afectar la calidad y exactitud de los resultados.

Dentro de esta categoría de herramientas se encuentran:

- Business Miner de Business Objects

  Cognos®BI [55](mejor conocido por Scenario) de Cognos®[56]

- Oracle Data Mining Suite (antes conocido como Oracle Darwin) de Oracle [57]

- DB2 Intelligent Miner, de IBM para DB2[35]

- SQL Server Analysis Server (SSAS) para SQL Server 2005[58] de Microsoft [59]

Oracle Data Mining Suite

[8] Ver definición en Glosario G

Producto:	Oracle Data Mining Suite
Vendedor:	Oracle Corporation
Funciones:	Clasificación, clustering, predicción, reglas de asociación
Técnicas:	Clasificación (Bayes para árboles de decisión, Naive Bayes, Model seeker), Clustering (K-medias, O-Clusters), reglas de asociación (A priori)
Plataformas:	Windows, Sun Solaris, HP-UX
Bases de datos:	Oracle
Versión actual:	10g (coincide con la versión de la base de datos Oracle)
Precio licencia 1 año:	U$ S 4.000
Precio licencia perpetua:	U$ S 20.000
Precio usuario nominado:	U$ S 400

Oracle Data Mining Suite [60] (ODM), es una herramienta de Minería de Datos embebida en de la base de datos de Oracle. Esto permite a Oracle proporcionar una infraestructura que integra fácilmente Minería de Datos con la base de datos. Las funciones de Minería de Datos tales como la construcción del modelo, prueba del modelo y evaluación se proveen mediante una API Java que da control programático completo de las funciones de Minería de Datos. Estos permiten procesar la Minería de Datos dentro de la base de datos.

Oracle Data Mining tiene dos componentes: Oracle Data Mining (ODM) API y el Servidor de Data Mining (DMS). ODM API usa el nuevo estándar Java Data Mining (JDM), permite a los usuarios escribir programas de Minería de Datos en lenguaje Java. El DMS es un componente de la base de datos que realiza las operaciones de Minería de Datos en la base de datos. El DMS provee un repositorio de meta datos que consiste de los objetos de entrada de Minería de Datos y los objetos resultados, además del espacio de nombres dentro del cual se almacena y se recuperan estos objetos.

ODM soporta los siguientes tipos de problemas de Minería de Datos: clasificación, predicción, regresión, clustering, asociaciones, importancia de atributos, extracción de características, búsquedas de semejanzas y análisis de secuencia.

DB2 Intelligent Miner de IBM

Producto:	DB2 Intelligent Miner
Vendedor:	IBM Corporation
Funciones:	Reglas de asociación, clasificación, clustering, predicción, series de tiempo, patrones secuenciales
Técnicas:	Arboles de decisión (CART modificado), K-medias, regresión lineal, redes neuronales (MLP, RBF,propagación hacia atrás)
Plataformas:	Windows NT/2000, Solaris, AIX, OS/390, OS/400
Base de datos:	DB2, archivos, otras BD via ODBC
Versión actual:	5.1
Precio software y mantenimiento por 1 año :	U$S 74.950

Dentro de la familia de productos de minería de IBM para DB2 se encuentra también Intelligent Miner for Text, especializado en minería de texto.

C.2.5. Herramientas de programación analíticas

Las herramientas de programación analíticas apuntan a tareas analíticas genéricas, específicamente Minería de Datos. Incluyen un enorme conjunto de funcionalidades gráficas, acceso a base de datos y estadística. Estas herramientas ofrecen gran flexibilidad, pero funcionan sólo para conjuntos de datos pequeños, lo cual no es lo común en los problemas de Minería de Datos. Con su flexibilidad pueden tratar tareas Minería de Datos de borde, pero esto puede ser aburrido, propenso al error y puede requerir personal adecuado.

Para analistas de negocio

- SPSS Base
- Herramientas SQL
- RISK para Microsoft Excel y Microsoft Project

Para analistas estadísticos

- SAS Macro Language y otros paquetes
- S-PLUS®de Insightful

- R, lenguaje y ambiente de análisis estadístico gratuito, extensible y con extras para bajar de diferentes sitios
- Módulos extra para Matlab, como por ejemplo NetLab

C.3. Como elegir la herramienta

No existe una herramienta de Minería de Datos que sea mejor que todas las demás. Cada una de las herramientas disponibles en el mercado tiene cualidades y carencias. La herramienta de Minería de Datos adecuada para un proyecto es la que mejor se ajuste a las necesidades particulares del proyecto y la empresa que pone en marcha el proyecto.

"Contrario a la opción general, la mejor herramienta para ti puede no ser la herramienta más avanzada, puede no ser aquella con más algoritmos de minería de datos ni aquella que de la mayor exactitud de predicción." [61]

Los atributos más deseables en una herramienta de Minería de Datos son la facilidad de uso, exactitud aceptable, posibilidad de desarrollar con ella todas las tareas del proceso de Minería de Datos. La facilidad de uso es un atributo en toda herramienta de software, al menos debe ser posible que el usuario de los primeros pasos sin necesidad de pasar por un proceso de aprendizaje complejo. En casos en que la curva de aprendizaje de la herramienta es muy pronunciada, la herramienta se hace menos popular. La exactitud es los que da confiabilidad a la herramienta, pero la relación costo-beneficioso puede inclinar la balanza hacia herramientas menos certeras pero de menor costo.

"Elegir el producto de Minería de Datos adecuado es encontrar una herramienta con buenas capacidades básicas, una interfase que se corresponda con las habilidades de las personas que la usarán, y características relevantes a los problemas de negocio específicos." [62]

C.4. Comparación de herramientas

Finalmente, para complementar la clasificación, Gartner Group proporciona una evaluación de los diferentes tipos de herramientas según las cualidades más deseables en un producto de software de Minería de Datos: fa-

cilidad de instalación, calidad de resultados, tiempo necesario para obtener la solución y flexibilidad.

Tipo de herramienta	**Facilidad de instalación**	**Calidad de resultados**	**Tiempo necesario para obtener la solución**	**Flexibilidad**
Genéricas	Regular / Bueno	Bueno / Muy Bueno	Regular / Muy bueno	Bueno / Muy Bueno
Para algoritmos específicos	Regular / Bueno	Muy Bueno / Excelente	Regular / Muy bueno	Regular / Bueno
Específicas por aplicación	Muy Bueno / Excelente	Regular / Muy bueno	Muy Bueno / Excelente	Malo / Regular
Embebidas	Bueno / Muy Bueno	Malo / Bueno	Bueno / Excelente	Regular / Bueno
De programación analíticas	Malo / Excelente	Regular / Excelente	Malo / Excelente	Excelente

Tabla C.1: Comparación de tipos de herramientas según GartnerGroup.

Obviamente hay una compensación entre la calidad de resultados, los niveles de habilidad requeridos, la flexibilidad y el tiempo necesario para llegar a la solución. Cuando los objetivos de la Minería de Datos son confusos y múltiples; las herramientas genéricas deben ser consideradas como la mejor opción. Por otra parte, diversas clases no son mutuamente excluyentes, sino que pueden complementarse, particularmente para las empresas con objetivos de Minería de Datos complejos o numerosos.

Las características más importantes de una herramienta de Minería de Datos son[63]

- Los tipos de problemas que pueden resolver.
- Los tipos de modelos que se pueden implementar.
- Utilidades de la herramienta para soportar el proceso completo de Minería de Datos
- Facilidad de uso: interfaz amigable, utilidades visuales para programación o para análisis de los resultados y ayuda en línea.
- Requerimientos del sistema: arquitectura, plataforma y sistema operativo.
- Bases de datos soportadas.

- Respaldo: quien es el fabricante y quien es el proveedor de la herramienta. Quien proporcionará el soporte técnico y las actualizaciones de versiones del producto. Existencia de versiones de evaluación, demostraciones del producto o documentación de casos de éxito.
- Precio.

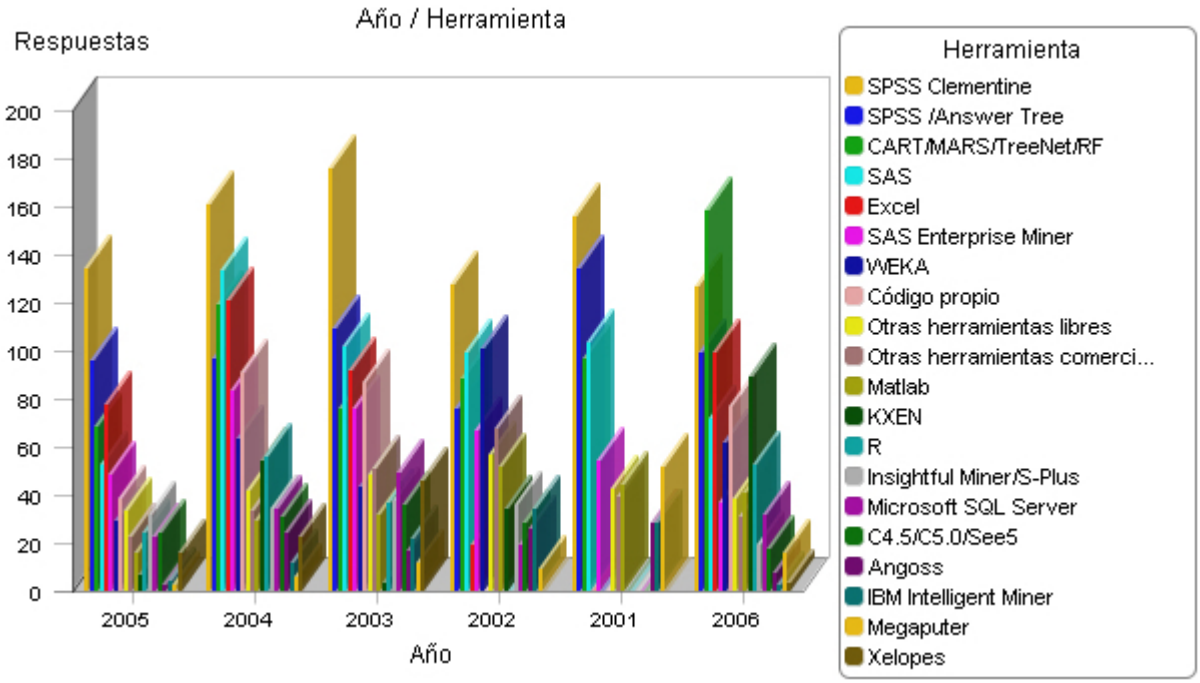

Figura C.5: 20 productos más usados en 2003, 2004, 2005 y 2006 según encuesta de KDnuggets™

Desde el año 2000 KDnuggets realiza anualmente una encuestaron la finalidad de determinar cuales son las herramienta de Minería de Datos más usadas. En la figura C.5 se grafican los 20 productos más utilizados según los resultados obtenidos para los años 2003, 2004, 2005 y 2006. Los datos para construir las gráficas provienen de las encuestas de uso de herramientas de Minería de Datos realizadas por KDnuggets™[64]; que se encuentran disponibles en *http://www.kdnuggets.com/polls*, último acceso octubre 2005. [9]

[9]Total de votos año 2003:1252, año 2004:1324, año 2005:860, año 2006:8596. Máximo 2 votos por persona.

C.4.1. Comparación por precio

Las herramientas de Minería de Datos tienden a cambiar de precio según su aplicación, siendo más baratas para uso académico que para su uso comercial en una relación generalmente de un 50 % y disponen en algunos casos de licencias gratuitas para investigadores. De acuerdo a los precios del mercado entre agosto y octubre del 2005, las herramientas se pueden agrupar en los siguientes rangos según su precio para uso comercial.

- Más de U$ S 10.000
  - SPSS Clementine
  - SAS Enterprise Miner
  - IBM Intelligent Miner
  - Insightful
  - KXEN
  - Fair Isaac
  - Oracle Data Mining Suite
  - IBM DB2 Intelligent Miner.
- De U$ S1.000 a U$ S9.999
  - Angoss
  - CART/MARS/TreeNet/Random Forests
  - Equbits
  - GhostMiner
  - Gornik
  - Mineset
  - Megaputer
  - Statsoft Statistica
  - ThinkAnalytics
- De U$ S1 a US$ 999
  - Excel
  - See5/C5.0 U$ S900 (http://www.rulequest.com/price.html)
- Gratis
  - C4.5
  - R
  - Weka
  - Xelopes
  - Yale

Apéndice D

Aplicaciones de Minería de Datos

D.1. Introducción

Las aplicaciones de la Minería de Datos son tan variadas como la variedad de datos disponible en el mundo. Los avances en tecnología están haciendo posible la generación, en muchos casos automática, de enormes volumenes de datos. En muchas disciplinas científicas, tales como astronomía, proyección de imagen médica, bioinformática, química y física estos avances se están usando intensamente, así como también en organizaciones comerciales. Naturalmente, el punto de partida es que los datos sean de valor, de lo contrario no vale la pena realizar Minería de Datos.

Por otra parte la diversidad de problemas y de áreas de aplicación que se pueden beneficiar de la Minería de Datos son tantas como los objetivos de las personas. El interés científico en los datos dista bastante del interés comercial en los mismos; las técnicas se puede aplicar igualmente a los datos de negocio como a los datos científicos. Las prácticas de investigación científica están cambiando, incrementalmente los científicos han incorporado tecnologías y técnicas nuevas como la Minería de Datos, que les permiten aprovechar mejor los datos de los cuales disponen. La alta competencia y posibilidades de mercado han propiciado que muchas empresas usen la Minería de Datos como un arma para múltiples objetivos: obtener información del mercado, adelantarse a las decisiones de sus competidores, lanzar nuevos productos.

Para llevar adelante un proyecto de Minería de Datos se necesita un equipo de personas e infraestructura adecuada. El primer paso es evaluar

los recursos, objetivos y necesidades. No basta con contar con un enorme volumen de datos si no se sabe que se desea obtener de ellos o no se cuenta con técnicos capaces de llevar a cabo la tarea.

De la gran variedad de problemas que pueden ser resueltos usando Minería de Datos, se pueden identificar tipos de problemas típicos y sugerir en función de la categoría el tipo de algoritmo más conveniente.

Predecir un atributo discreto Como por ejemplo predecir si un cliente es apto para otorgarle un crédito o no, predecir si ocurrira una tormenta en los próximos 10 días. En ambos casos la repuesta será "si" o "no". Los algoritmos convenientes en este caso son árboles de decisión, Naive Bayes, clustering y redes neuronales.

Predecir un atributo continuo Como por ejemplo predecir las ventas a producirse en el siguiente año, predecir el monto de las compras que realizará un cliente. En ambos casos la respuesta es un valor dentro de un dominio continuo. Los algoritmos convenientes en este caso son árboles de decisión y series de tiempo.

Predecir una secuencia Los algoritmos convenientes en este caso son los de clustering.

Encontrar grupos de elementos similares en transacciones Por ejemplo para sugerir productos a un cliente en base a lo que contiene su carrito de compras. Los algoritmos convenientes en este caso son árboles de decisión y reglas de asociación.

Encontrar grupos de elementos similares Los algoritmos convenientes en este caso son los de clustering.

D.2. Aplicación al mercadeo

Hasta hace poco tiempo, los datos de las empresas estaban orientados principalmente a alimentar sus sistemas contables, financieros, de ventas, de inventarios, de producción, de recursos humanos. En la medida que los negocios se hacen cada día más complejos y competitivos, los datos han cobrado más vida y se han convertido en información vital para la toma de

decisiones empresariales. Entender al consumidor es una tarea cada vez más compleja; la antigua noción de desarrollar un producto e inducir su compra a un cliente potencial mediante el uso de la publicidad masiva ya murió. Para cada producto o servicio hay numerosas opciones de mercados posibles. Seleccionar el mercado y luego segmentarlo es una tarea compleja que puede ocupar mucho tiempo. Tras la aparente similitud de los consumidores existe una heterogeneidad derivada de las diferencias en educación, ocupación, ingresos, etnias, culturas, estilos de vida, percepciones, necesidades y deseos.

El mercadeo mediante Minería de Datos, convierte una plataforma tecnológica en un sistema de información sobre el que se construyen soluciones de negocios. La minería de datos se integra a los procesos modernos de ventas de muchas formas y con diversos objetivos. Facilita la comunicación con la cartera de clientes mediante marketing directo (comunicación en fechas especiales para el cliente como su cumpleaños o aniversario) o marketing masivo (comunicación en fechas especiales como feriados o nuevos lanzamientos). Posibilita vender más a los clientes de la empresa aplicando técnicas de venta cruzada. Permite capturar nuevos clientes aplicando técnicas como networking. Existen numerosas aplicaciones de la Minería de Datos a las actividades de mercadeo, como por ejemplo:

- Segmentación del mercado
- Tendencias de deserción de clientes
- Descubrimiento de transacciones fraudulentas
- Mercadeo directo
- Mercadeo interactivo
- Análisis de canasta
- Análisis de tendencias
- Perfiles de clientes

Ejemplos de uso de Minería de Datos en mercadeo:

Venta cruzada, es una técnica basada en mercadeo concéntrico que consiste en hacer múltiples ofertas a un mismo cliente. A mayor cantidad de transacciones o relaciones que sostenga una cliente con una empresa, mayor será la capacidad de la empresa de retenerlo con el paso del tiempo. Este cruce de productos puede ser personal o masivo. Cada uno requiere de la segmentación de la clientela para adaptar la oferta a las necesidades

del cliente o grupo de clientes. La existencia de alguna matriz que identifique cuáles productos se han vendido a cuáles clientes (y cuáles no) facilitar ofertas futuras.

Networking, cada cliente puede ser una fuente de negocios adicionales. Esta técnica busca crear redes de cuentas potenciales. Por ejemplo: Pueden ser nuestros clientes los empleados, los propios clientes, los proveedores de los clientes, afiliados, etc.

Análisis del carrito de compras, es ejemplo típico de minería de datos en mercadeo y uno de los más utilizados. Este proceso aplica métodos de asociación para analizar los hábitos de compra de los clientes y así deduce asociaciones entre los artículos que los clientes ponen en sus carritos de compra. El descubrimiento de tales asociaciones ayuda a desarrollar estrategias de comercialización aprovechando entrar al mercado un producto asociados al artículo junto con el cual es frecuentemente comprado. Tal información puede conducir a las ventas crecientes ayudando a la comercialización selectiva y a planificar la distribución de los artículos en el comercio. Por ejemplo, la colocación de productos relacionados físicamente cerca en el comercio, puede incrementar la venta de estos artículos. Internet y el comercio en línea le ha dado una nueva dimensión a este método. Esta es seguramente la aplicación más conocida de Minería de Datos; cualquier persona que haya navegue frecuentemente en la web o realice compras en la web ha sido objeto de análisis de este tipo.

D.3. Aplicación a CRM

La Minería de Datos aplicada a CRM [1] sirve para evaluar y desarrollar un conjunto de reglas de negocio relacionadas a la interacción de la empresa y los clientes. Algunos ejemplos concretos del uso de Minería de Datos en CRM son:

- Modelar la probabilidad de respuesta ante una oferta de un nuevo producto o servicio. Dado un conjunto de clientes, su historial de compras y su respuesta a ofertas y promociones pasadas, determinar la probabilidad de respuesta a nuevas ofertas y promociones.

[1] CRM, siglas en inglés de Customer Relationship Management, en español, gestión de la relación con el cliente

- Basándose en las reglas obtenidas las campañas de marketing de una empresa pueden tener como objetivo la máxima respuesta para generar un nivel deseado de respuesta, ingresos o beneficios.

- Desarrollo de técnicas de fidelización de los clientes. Para las empresas es más fácil retener un cliente, que captar uno nuevo; por lo tanto es muy importante entender como esto es posible. Una entidad financiera puede, analizando los datos provenientes de diferentes canales (teléfono, internet, cajero automático) detectar patrones de comportamiento de los clientes e identificar a aquellos que pretenden cancelar una cuenta. Ello les permite anticiparse e intentar evitarlo enviando una carta promocional al cliente o realizando una acción comercial directa, como una llamada teléfonica para consultarlo sobre su satisfacción con el servicio.

D.4. Aplicación en la detección de fraude

Una de las áreas comerciales que incorporó más tempranamente la Minería de Datos es el sector financiero, que comprende a bancos y aseguradoras. En el sector financiero se usa desde hace más de 10 años para modelar y predecir fraude, evaluar riesgo, analizar tendencias y ganancias. La aplicación más extendida es la detección de fraude y existen productos de software específicos para esa área. Actualmente la mayoría de los bancos emisores de tarjetas de crédito cuentan con los medios necesarios para detectar cuando una transacción de una tarjeta de crédito es fraudulenta. Para alimentar los sistemas que realizan estos controles es necesario crear bases de datos o datawarehouse que contengan además de la información transaccional, información específica para la detección de fraude. Algunos de los datos a tales efectos son el promedio de transacciones por mes, el número de transacciones fuera de la ciudad origen, la existencia de fraude previo.

Están disponibles modelos de detección de fraude para:

- Transferencia de fondos por Internet.

- Retiro de fondos en cajeros automáticos.

- Uso de Tarjeta de Débito o de Crédito.

D.5. Aplicación a bioinformática

Bioinformática se puede definir como la aplicación de tecnologías computacionales a la administración de información médica. (anónimo)

Bioinformática es un campo emergente cuyo desarrollo responde a los desafíos planteados por la investigación y uso de la biología moderna. Su objetivo es proveer a los investigadores de herramientas adecuadas para la investigación biomédica, se ocupa de la utilización y almacenamiento de grandes cantidades de información biológica. El uso de las computadoras como herramientas para la adquisición, consulta y análisis de la información biológica es fundamental. Las nuevas tecnologías, basadas en la genética molecular e informática, proveen de potentes instrumentos para la obtención y el análisis de la información genética.

Algunas de las metas fundamentales de la bioinformática son: la predicción de la estructura tridimensional de las proteínas a partir de su secuencia, la predicción de las funciones biológicas y biofísicas a partir de secuencias o estructuras, simular el metabolismo y otros procesos biológicos. Actualmente, la mayoría de los proyectos de bioinformática que se desarrollan en el mundo, requieren de la aplicación de técnicas de minería de datos para poder determinar qué es realmente importante dentro del enorme volumen de información que se genera para estos proyectos. El mejor ejemplo es el proyecto Genoma Humano.

Aplicaciones concretas de los algoritmos de Minería de Datos a bioinformática son:

Experimentos microarray de ADN [2], miden todos los genes de un organismo, proporcionando un punto de vista genómico de la expresión del gene. La mayoría de las herramientas del análisis usadas actualmente se basan en algoritmos de clustering. Estos algoritmos procuran localizar grupos de genes que tienen patrones similares de la expresión sobre un sistema de experimentos.

Alineación de la secuencia, es una herramienta importante en bioinformática. Es capaz de identificar las regiones similares y divergentes entre dos secuencias, por ejemplo secuencias biológicas de ADN o proteínas.

Lectura recomendada: The GeneMine system for genome/proteome annotation and collaborative data mining de C.Lee y K.Irizarry [65].

[2] Ácido desoxirribonucleico, material genético de los organismos

D.6. Aplicación a detección de intrusos

Los sistemas informáticos de red desempeñan un papel cada día más importante, por lo cual se han convertido en el objetivo de criminales e inescrupulosos. Es necesario encontrar formas eficientes para protegerlos cuando la seguridad se ve comprometida cuando por una intrusión. Una intrusión se puede definir como, la tentativa de comprometer la integridad, el secreto o la disponibilidad de un recurso.

Las técnicas de la prevención de intrusiones son autenticación de usuarios (contraseñas, huellas digitales, huellas oculares), control de errores programación (revisión, prueba de debilidades) y protección de los datos (cifrado de datos). La prevención de la intrusión no es suficiente, no existen los sistemas invulnerables. La detección de la intrusión por lo tanto otra forma de proteger los sistemas informáticos. Los elementos centrales en la detección de la intrusión son: los recursos de un sistema a ser protegidos; los modelos que describen el comportamiento normal o legítimo de recursos; las técnicas que comparan las actividades reales del sistema con los modelos e identifican los que es anormal. Las técnicas de la detección de la intrusión se pueden categorizar en detección del uso erróneo, que utiliza los patrones de ataques bien conocidos o los puntos débiles del sistema para identificar intrusiones; y detección de anomalías, que intenta determinar si la desviación de los patrones de uso normales pueden identificar como intrusiones.

La Minería de Datos proporciona múltiples algoritmos útiles para la detección de intrusos. Los algoritmos de clasificación identifican cada elemento con una categoría del conjunto de categorías predefinidas. La aplicación típica a detección de intrusos tendrá dos grupos etiquetados como *normal* o *anormal*. Para que el sistema de Minería de Datos pueda ser usado con confianza para la detección de intrusos deberá ser adecuadamente entrenado con conjuntos de datos que contengan casos para ambas categorías *normal* y *anormal*. Los algoritmos de reglas de asociación determinan relaciones entre datos; descubrir la relación entre las intrusiones puede ayudar a identificar la flaqueza del sistema. Los algoritmos de análisis secuencial pueden ayudar a entender la secuencia de acontecimientos de de las intrusiones. Estos patrones son elementos importantes que pueden alcanzar para describir el comportamiento de un usuario o de un programa.

D.7. Aplicación a la industria

La Minería de Datos puede aplicarse a la optimización de procesos industriales, una de las áreas de mayor interés es la búsqueda de patrones en históricos que puedan ayudar a tomar decisiones o a mejorar los procesos productivos [66].

En la industria manufacturera: muchos procesos son tan automatizados que comprenden miles de mediciones por cada artículo producido. La Minería de Datos puede identificar problemas en la producción alimentándose de las mediciones que se obtienen durante la producción.

Dentro del proceso industrial es usual encontrar bases de datos llenas de series temporales. Las mediciones son capturadas automáticamente con una frecuencia de muestre constante dentro de un período de tiempo en general largo. Las metodología se basan en extraer tramos de series temporales previamente filtradas para eliminar ruido y obtener la forma básica. Estas series son observadas y procesadas y se utilizan luego para buscar repeticiones de la misma en otras series temporales. El objetivo es determinar reglas asociativas de los casos que aparecen con mayor frecuencia.

Apéndice E

Anexo WEKA

E.1. Archivo de formato ARFF

@relation indica el comienzo del archivo.

@atribute define los atributos que componene los datos. Los atributos nominales van seguidos de los valores que pueden tomar encerrados por llaves. Los atributos numéricos van seguidos por la palabra clave *numeric*. Los atributos de texto van seguidos por la palabra clave *string*. Los atributos de fecha van seguidos por la palabra clave *date*. El archivo ARFF simplemente proporciona los datos, no especifica cual o cuales son los atributos a predecir.

@data indica el comienzo de la zona de datos. Cada instancia se escribe a continuación, una por línea, separando por comas los valores de cada uno de los atributos anteriormente definidos. Los valores deben ajustarse a los tipos especificados en la sección *atribute*, si hay un valor faltante en el registro para un atributo se indica con el símbolo *?*.

% Las líneas que comienzan con % son comentarios.

```
%% ARFF file for the weather data with some numeric features
% @relation clima
@attribute prónostico  soleado, cubierto, lluvioso
@attribute temperatura numeric
@attribute humedad numeric
@attribute ventoso  verdadero, falso
@attribute jugar  sí, no
@data
% 6 instancias
soleado, 85, 85, falso, no
soleado, 80, 90, verdadero, no
cubierto, 83, 86, falso, sí
lluvioso, 70, 96, falso, sí
lluvioso, 68, 80, falso, sí
lluvioso, 65, 70, verdadero, no
cubierto, 64, 65, verdadero, sí
soleado, 72, 95, falso, no
soleado, 69, 70, falso, sí
lluvioso, 75, 80, falso, sí
soleado, 75, 70, verdadero, sí
cubierto, 72, 90, verdadero, sí
cubierto, 81, 75, falso, sí
lluvioso, 71, 91, verdadero, no
```

Figura E.1: Ejemplo de archivo ARFF.

E.2. Definición de medidas de WEKA

Muchas medidas pueden ser usadas para evaluar el éxito o fracaso de una predicción numérica. Sean los valores predecidos la secuencia de valores $p_1, p_2, ..., p_n$ y los valores reales $r_1, r_2, ..., r_n$. A continuación se definen algunas de las posibles medidas de desempeño que es posible utilizar para medir este éxito o fracaso. [33]

La decisión de cual medida usar dependerá de estudiar para el caso en que se quiere aplicar cual es el valor de mayor interés o el que más aporta en la interpretación de los resultados. Afortunadamente sucede que en la mayoría de las situaciones prácticas la mejor predicción es la mejor sin importar la medida de error elegida [33].

Media del error al cuadrado ***(Mean squared error)*** es la medida más común y más usada como medida de error en muchas técnicas matemáticas, pero en Minería de Datos no existe especial preferencia por ella sobre las demás medidas existentes. Cuenta con el defecto de que tiende a exagerar los efectos de valores atípicos [33].

$$\frac{(p_1 - r_1)^2 + ... + (p_n - r_n)^2}{n} \tag{E.1}$$

Media del error absoluto ***(Mean absolute error)*** es una medida alternativa en la que no se toma en cuenta el signo del error. Evita el problema de los valores atípicos que presenta la *Media del error al cuadrado*[33].

$$\frac{|p_1 - r_1| + ... + |p_n - r_n|}{n} \tag{E.2}$$

Error relativo al cuadrado ***(Relative squared error)*** calcula el error relativo a lo que pudo haber sido si se hubiese usado un predictor simple. El predictor simple es el promedio de los valores de los datos de entrenamiento. Por lo tanto se normaliza el error absotuto al cuadrado dividiéndolo entre error absotuto al cuadrado del predictor[33].

$$\frac{(p_1 - r_1)^2 + ... + (p_n - r_n)^2}{(r_1 - r)^2 + ... + (r_n - r)^2} \tag{E.3}$$

Error absoluto relativo ***(Relative absolute error)*** es el error absoluto normalizado de la misma forma que el error relativo al cuadrado. [33]

$$\frac{|p_1 - r_1|^2 + ... + |p_n - r_n|^2}{|r_1 - r|^2 + ... + |r_n - r|^2} \tag{E.4}$$

donde r es el promedio de los valores de los datos de entrenamiento.

Coeficiente de correlación ***(Correlation coefficient)*** Mide la correlación estadística entre los valores predecidos p_i y los correspondientes valores reales. Vale 1 para resultados perfectamente correlativos, 0 cuando no existe correlación y -1 para correlación negativa.

Raíz del error medio al cuadrado ***(root mean squared error)*** [33]

$$\sqrt{\frac{|p_1 - r_1|^2 + ... + |p_n - r_n|^2}{|r_1 - r|^2 + ... + |r_n - r|^2}} \tag{E.5}$$

donde r es el promedio de los valores de los datos de entrenamiento.

Raíz del error relativo al cuadrado *(Root relative squared error)* [33]

$$\sqrt{\frac{(p_1 - r_1)^2 + ... + (p_n - r_n)^2}{(r_1 - r)^2 + ... + (r_n - r)^2}} \tag{E.6}$$

donde r es el promedio de los valores de los datos de entrenamiento.

Estadística Kappa *(Kappa stadistic)* Esta medida fue introducida por Cohen en 1960, se utiliza para medir la concordancia entre los valores predecidos y los observados en un conjunto de datos restando las predicciones que pudieron ser correctas con un predictor al azar. El valor máximo que puede tomar es 100 %, se da cuando el predictor aleatorio no predice correctamente ningún valor [33].

Ejemplo de cálculo de Kappa :

Valores	Predecidos			
	a	b	c	Total
Reales	88	14	2	100
	14	40	6	60
	18	10	12	40
Total	120	60	20	200

[a]

[a]Predicción obtenida con un predictor a evaluar.

Valores	Predecidos			
	a	b	c	Total
Reales	60	30	10	100
	36	18	6	60
	24	12	4	40
Total	120	60	20	200

[a]

[a]Predicción obtenida con un predictor aleatorio.

Las instancias correctamente predecidas por el predictor aleatorio suman 82 y las instancias correctamente predecidas por el predictor que se quiere

evaluar suman 140. Entonces el éxito del predictor que se está evaluando es la diferencia entre ambos $140 - 82 = 58$ de un total de $200 - 82 = 118$. El valor de Kappa es entonces $(58 * 100)/118 = 49{,}15\,\%$

TP (True Positive).
En base a una clasificación en dos valores (positivo y negativo), es la cantidad de valores clasificados correctamente como positivos. [33]

FP (False Positive).
En base a una clasificación en dos valores (positivo y negativo), es la cantidad de valores clasificados positivos cuando su valor real es negativo. [33]

TP Rate (True Positive Rate).
En base a una clasificación en dos valores (positivo y negativo), es el porcentaje de clasificaciones correctas para el valor positivo. Se calcula dividiendo TP entre la cantidad total de positivos (TP + FN). [33]

TN (True Negative).
En base a una clasificación en dos valores (positivo y negativo), es la cantidad de valores clasificados correctmente como negativos. [33]

FN (False Negative).
En base a una clasificación en dos valores (positivo y negativo), es la cantidad de valores clasificados negativos cuando su valor real es positivo. [33]

TN Rate (True Negative Rate).
En base a una clasificación en dos valores (positivo y negativo), es el porcentaje de clasificaciones correctas para el valor negativo. Se calcula dividiendo TP entre la cantidad total de negativos (TN + FP). [33]

El éxito total de la clasificación está dado por el valor $\frac{TP+TN}{TP+FP+TN+FN}$ y el error es $1 - \frac{TP+TN}{TP+FP+TN+FN}$.

Precisión (Precision)
Es el porcentaje de positivos sobre el total de clasificaciones positivas.
$Precision = (\frac{TP}{TP+FP}) * 100\,\%$.

Recall
Es el porcentaje de positivos clasificados correctamente sobre el total de positivos, igual a TP Rate.
$Recall = \frac{TP}{TP+FN}$.

F-Measure
$F - Measure = \frac{2*recall*precision}{recall+precision} = \frac{2*TP}{2*TP+FP+FN}$.

E.3. Interpretación de resultados de WEKA

La salida estándar de Weka (si no se eligen opciones especiales), se divide en cinco partes :

- Información de ejecución. *(Run Information)*
- Modelo de clasificación. *(Classifier model (full training set))*
- Sumario. *(Summary)*
- Precisión detallada por clase. *(Detailed Accuracy By Class)*
- Matriz de confusión. *(Confusion Matrix)*

Un ejemplo de salida se muestra en la Figura E.5

Información de ejecución
Esta sección contiene información general del algoritmo ejecutado y la fuente de datos. Bajo la etiqueta de nombre *Scheme* se especifica el algoritmo ejecutado con todos sus parámetros. Bajo las etiquetas de nombre *Relation*, *Instances* y *Attributes* se detalla la estructura del archivo de entrada ARFF utilizado. La etiqueta *Test mode* aparece si existen modos especiales de testeo, como ser validación cruzada, corresponde con las opciones elegidas al momento de la ejecución del algoritmo [33].

Modelo de clasificación
En esta sección se detalla el modelo de clasificación obtenido y el tiempo utilizado para su construcción. En la Figura E.5 se muestra un modelo de árbol de decisión que se puede dibujar como muestra la Figura E.3

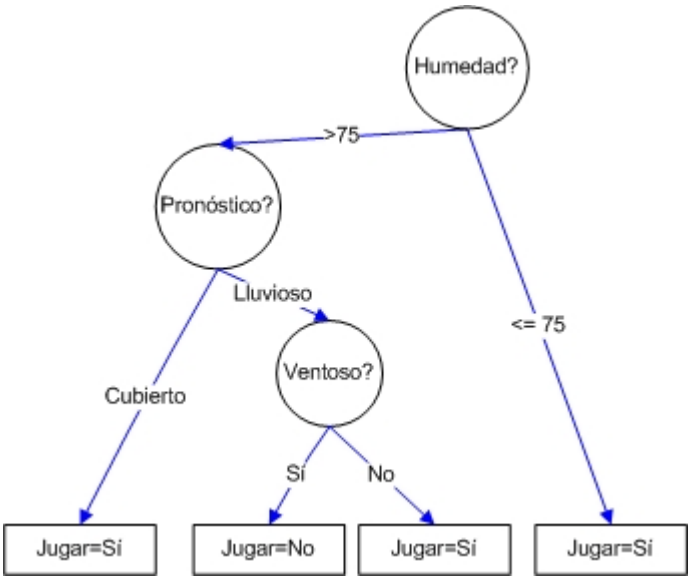

Figura E.2: Representación arbol de decisión de E.5.

Si se tratase de una red neuronal el modelo de clasificación descripto en la salida de WEKA sería del estilo del que se muestra en la Figura E.3 . La notación *Sigmoid Node* se usa para indicar un nodo sigmoidal (cuya función es sigmoidal) y pueden corresponder tanto a nodos de entrada como a nodos de la capa oculta. Los nodos 0 y 1 son los nodos de entrada, los nodos 2,3 y 4 son los nodos de la capa oculta y los nodos A y B son los nodos de salida. Entonces la red neuronal correspondiente se puede dibujar como se muestra en la Figura E.3 , donde se omiten los pesos de las aristas para simplificar el gráfico.

```
=== Classifier model (full training set) ===
Sigmoid Node 0
    Inputs Weights
    Threshold -1.7
    Node 2 -4.7
    Node 3 -1.3
    Node 4 1.9
Sigmoid Node 1
    Inputs Weights
    Threshold -0.1
    Node 2 -1.8
    Node 3 -2.7
    Node 4 -0.0
Sigmoid Node 2
    Inputs Weights
    Threshold -0.7
    Attrib X 3.0
    Attrib Y -1.4
Sigmoid Node 3
    Inputs Weights
    Threshold -0.5
    Attrib X 2.4
    Attrib Y -0.8
Sigmoid Node 4
    Inputs Weights
    Threshold -1.0
    Attrib X -0.8
    Attrib Y 1.2
Class A
    Input
    Node 0
Class B
    Input
    Node 1
```

Figura E.3: Ejemplo salida de red neuronal en WEKA.

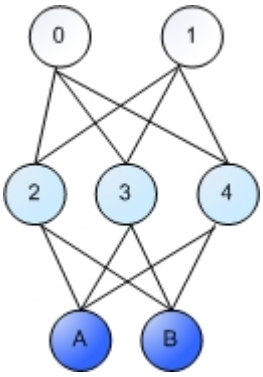

Figura E.4: Representación de red neuronal de E.3.

Sumario

Si se eligió realizar validación cruzada aparecerá bajo el título *Stratified cross-validation*, en cambio si se eligió probar el modelo con datos de entrenamiento aparecerá bajo el título *Evaluation on training set*. Esta sección comienza con la discriminación de las instancias correctamente clasificadas *(Correctly Classified Instances)* e incorrectamente clasificadas *(Incorrectly Classified Instances)* del conjunto de datos en cantidad y porcentajes sobre el total de datos. A continuación se indican los valores de estadística Kappa *(Kappa stadistics)*, *(Mean absolute error)*, *(Root mean squared error)*, *(Relative absolute error)*, *(Root relative absolute error)*. El significado de estas medidas se explicó en la sección anterior. Finalmente se indica la cantidad de instancias procesadas bajo la etiqueta *(Total Number of Instances)* [33]

Precisión detallada por clase

En esta sección se presenta una tabla cuyas filas son las clases de la clasificación y cuyas columnas son las siguientes: TP Rate, FP Rate, Precision, Recall, F-Measure, Class. A continuación se explica que significado tiene cada una de ellas. [33] Los valores de estas medidas tienen sentido en una clasificación en dos clases, para una clasificación multiclase se utiliza la matriz de confusión para visualizar los resultados. El significado de estas medidas se explicó en la sección anterior.

Matriz de confusión

Ver definición de matriz de confusión en el Anexo matemático-estadístico F

```
=== Run information ===
Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2
Relation: clima
Instances: 14
Attributes: 5
   pronóstico
   temperatura
   humedad
   ventoso
   jugar
Test mode: 10-fold cross-validation
=== Classifier model (full training set) ===
J48 pruned tree
------------------

pronóstico = soleado
— humedad ≤ = 75: yes (2.0)
— humedad > 75: no (3.0)
pronóstico = cubierto: yes (4.0)
pronóstico = lluvioso
— ventoso = TRUE: no (2.0)
— ventoso = FALSE: yes (3.0)

 Number of Leaves : 5
 Size of the tree : 8
 Time taken to build model: 0.27 seconds
 === Stratified cross-validation ===
 === Summary ===
 Correctly Classified Instances 9 64.2857 %
 Incorrectly Classified Instances 5 35.7143 %
 Kappa statistic 0.186
 Mean absolute error 0.2857
 Root mean squared error 0.4818      Relative absolute error 60 %
 Root relative squared error 97.6586 %
 Total Number of Instances 14
 === Detailed Accuracy By Class ===
 TP Rate FP Rate Precision Recall F-Measure Class
 0.778 0.6 0.7 0.778 0.737 yes
 0.4 0. 222 0.5 0.4 0.444 no
 === Confusion Matrix ===
 a b ≤ – classified as
 7 2 — a = yes
 3 2 — b = no
```

Figura E.5: Ejemplo salida de WEKA.

Apéndice F

Anexo matemático - estadístico

Modelo matemático. Un modelo matemático es el uso de lenguaje matemático para describir el comportamiento de un sistema.

Sistema. Es una porción del universo compuesta por un conjunto de elementos y relaciones entre los mismos desde el punto de vista de un observador.

Sistema estocástico. Un sistema estocástico es aquel cuyo funcionamiento se define mediante leyes probabilísticas, no determinísticas. Las leyes causa-efecto no explican cómo actúa el sistema, sino una función de probabilidad.

Proceso estocástico. Un proceso estocástico se define como un conjunto de variables aleatorias cuya distribución varía de acuerdo a un parámetro de tiempo t. Las variables aleatorias toman sus valores de un conjunto denominado espacio de estados.

Variables categóricas o de clase. Son variables simples que contienen códigos o valores de texto para distinguir clase. Por ejemplo: la variable Sexo tiene las categorías Femenino y Masculino.

Valor atípico Elemento de los datos cuyo valor cae fuera de los límites que encierran a la mayoría de los valores de la muestra. Puede indicar datos anormales, por lo tanto deben ser examinados detenidamente; pues pueden dar información importante de la muestra.

Matriz Hessiana. Sea F una función escalar $F : \Re^n \to \Re$ tal que $F(x) = F(x_1, x_2, ..., x_n)$. La matriz hessiana de $F(x)$ es la matriz simétrica cuyo elemento (i, j) es $\frac{\partial^2 F(x)}{\partial x_i \partial x_j}$.

$$\text{Entonces } H(w) = \nabla_x^2 F(X) = \begin{bmatrix} \frac{\partial}{\partial x_1}\left(\frac{\partial F}{\partial X}\right)^T \cdot \\ \cdot \\ \cdot \\ \frac{\partial}{\partial x_n}\left(\frac{\partial F}{\partial X}\right)^T \end{bmatrix} = \begin{bmatrix} \frac{\partial^2 F(x)}{\partial x_1^2} & \frac{\partial^2 F(x)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 F(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 F(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 F(x)}{\partial x_2^2} & \cdots & \frac{\partial^2 F(x)}{\partial x_2 \partial x_n} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \frac{\partial^2 F(x)}{\partial x_n \partial x_1} & \frac{\partial^2 F(x)}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 F(x)}{\partial x_n^2} \end{bmatrix}$$

La matriz hessiana de $F(x)$ es la matriz Jacobiana del gradiente $\nabla F(x)$. Una aplicación típica de la matriz hessiana es la optimización no lineal (minimización o maximización) de la función de costo $J(w)$.

Matriz de confusión. Una matriz de confusión es una tabla bidimensional NxN que indica el número de predicciones correctas e incorrectas que un modelo de clasificación hizo en un conjunto de datos de prueba específico. Proporciona una medida de precisión del modelo y de donde pueden encontrarse los errores. En ella se puede observar fácilmente los errores cometidos por los algoritmos de predicción [74]. En la diagonal se observan los valores correcamente clasificados y en los demás casilleros los valores incorrectamente clasificados. De ella se pueden obtener entonces los porcentajes de clasificaciones correctas e incorrectas con las cuales es posible evaluar el desempeño del proceso de clasificación [33].

Ejemplo

En base a los valores de la matriz de confusión deducimos:
Clasificaciones correctas $(88 + 40 + 12)/200 = 70\,\%$
Clasificaciones incorrectas $(200 - (88 + 40 + 12))/200 = 30\,\%$

Valores		Predecidos		
	a	b	c	Total
	88	14	2	100
Reales	14	40	6	60
	18	10	12	40
Total	120	60	20	200

Tabla F.1: Matriz de confusión.

Entropía. Es una medida que caracteriza la homogeneidad de un conjunto arbitrario de muestras.

$$E(P) = -\sum_{i=1}^{n}(p_i * log(p_i)) \tag{F.1}$$

donde n es la cantidad de muestras y p_i es la probabilidad de cada una.

Fórmula de Bayes.

$$P(C_i|X) = \frac{P(X|C_i) * P(C_i)}{P(X)} \tag{F.2}$$

Función sigmoidal. La función sigmoidal, también llamada curva sigmoidal o función logística, es la función

$$f(x) = \frac{1}{1 + \exp(a * x)} \tag{F.3}$$

Esta función produce una curva con forma de "S" como la de la figura F.1. Las funciones sigmoidales se usan frecuentemente en redes neuronales para introducir el concepto de no linealidad del modelo. También se usan para asegurar que ciertas señales permanezcan en determinado rango de valores o "suavizar"la función original. Una de las razones de su popularidad en redes neuronales es que satisface la siguiente propiedad entre la función y su derivada, lo cual simplifica los cálculos. $\frac{d}{dx}sig(x) = sig(x) * (1 - sig(x))$

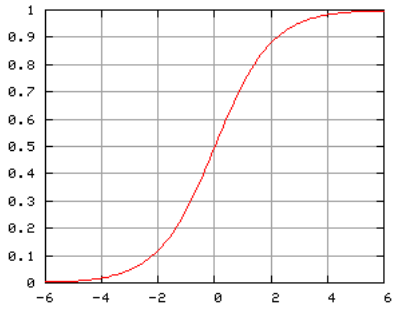

Figura F.1: Función sigmoidal.

Distribución chi-cuadrado. La distribución chi-cuadrado fue desarrollada por Karl Pearson en 1900.

$$chi - cuadrado = \sqrt{\frac{(valorObservado(x) - valorEsperado(x))^2}{valorEsperado(x)}} \quad \text{(F.4)}$$

Chi-cuadrado depende del grado de libertad. La idea de los grados de libertad es cuantas variables diferentes se necesitan para describir la tabla de valores esperados. Si la tabla tiene F filas y C columnas, la cantidad de celdas es F*C. Si no hay restricciones en la tabla, este es el número de variables necesarias. Como la suma de los valores esperados en las columnas y las filas coincide con el valor real los grados de libertad son F*C-F-C. Además la suma de todas las filas y todas las columnas coincide, por lo tanto, el valor final es (F-1)*(C-1).

Precisión. En base a la matriz de confusión se calcula la precisión del modelo de la siguiente forma. Se por ejemplo la matriz de confusión

Real vs. Predecida	A	B
A	250	6
B	21	506

la precisión del modelo se calcula como (250+506)/(250+506+21+6)=96.6 % [74]

Error de Predicción. Sea Y una variable aleatoria. El error que se comete al predecir un valor particular de Y mediante Y' es la diferencia entre el

valor observado de Y y el valor predicho. En base a la matriz de confusión se calcula la precisión del modelo de la siguiente forma. Se por ejemplo la matriz de confusión

Real vs. Predecida	A	B
A	250	6
B	21	506

el error es (21+6)/(250+506+21+6)=3.4 % [74]

Valor esperado. Si c es constante entonces $E(c) = c$ Si $Y_1 e Y_2$ son variables aleatorias, g es un función de Y_1, Y_2 y la c es constante; entonces el valor esperado $E[c.g(Y_1, Y_2)] = c.E[g(Y_1, Y_2)]$ Si $Y_1 e Y_2$ son variables aleatorias con la función de densidad conjunta $f(y_1, y_2)$ y $g_1, g_2, ..., g_n$ son funciones de Y_1, Y_2; entonces $E[g_1(Y_1, Y_2) + g_2(Y_1, Y_2) + ... + g_n(Y_1, Y_2)] = E[g_1(Y_1, Y_2)] + ... + E[g_n(Y_1, Y_2)]$

Covarianza. Intuitivamente pensamos en la dependencia de dos variables aleatorias Y_1 e Y_2, como el caso en el que una variable crece o decrece cuando la otra cambia. Suponiendo conocidos los valores $E(Y_1) = \mu_1 y E(Y_2) = \mu_2$; el valor medio de $(y_1 - \mu_1)(y_2 - \mu_2)$ da una medida de la dependencia lineal de Y_1 e Y_2. Esta cantidad, definida sobre la población bivariable asociada a Y_1 e Y_2 se denomina *Cobarianza* de Y_1 e Y_2. A mayor valor absoluto de la cobarianza corresponde mayor dependencia lineal entre las variables. Valores positivos indican que Y_1 crece cuando Y_2 crece; valores negativos que Y_1 decrece cuando Y_2 crece y valor cero indica que no existe dependencia lineal. La covarianza de Y_1 e Y_2 se define como el valor esperado de $(Y_1 - \mu_1)(Y_2 - \mu_2)$.

$$Cov(Y_1, Y_2) = E[(Y_1 - \mu_1)(Y_2 - \mu_2)] \tag{F.5}$$

en donde $y_1 = E(Y_1)$ y $y_2 = E(Y_2)$.

Coeficiente de correlación. El coeficiente de correlación lineal se relaciona con la covarianza y se define así

$$\rho = \frac{Cov(Y_1, Y_2)}{\theta_1 \theta_2} \tag{F.6}$$

donde $\theta_1 y \theta_2$ son las desviaciones estándares de Y_1 e Y_2 respectivamente. El coeficiente de correlación satisface la desigualdad $-1 \leq \theta \leq 1$. Los valores

-1 o +1 representan la correlación perfecta en tanto el valor 0 representa la no existencia de correlación. El signo se corresponde con el de la covarianza y mantiene su significado.

Apéndice G

Glosario

Business Intelligence. (BI) es una categoría de tecnologías que permite recopilar, almacenar, acceder y analizar datos para apoyar a los usuarios a tomar mejores decisiones en el ámbito de negocios, en sentido de optimizar las operaciones de negocio y fomentar su crecimiento.

Decision Support. En inglés se conoce con la sigla DS bajo varios términos: Decision Support, Decision Science, Decision Systems. A pesar de que el término "toma de decisiones" parece ser intuitivo y simple, da lugar a varias interpretaciones dependiendo de la persona y el contexto. Además, su significado ha cambiado en los últimos años. Desde hace muchos años se le asocia con Investigación de Operaciones y Análisis de Decisiones. Una década atrás se le asociaba con sistemas de soporte de decisiones (en inglés Decision Support Systems, sigla DSS). Hoy en día se le asocia también con data warehouses, OLAP y Minería de Datos. Decision Support es un campo que concierne a brindar apoyo a las personas para tomar decisiones y pertenece a las llamadas Ciencias de Decisión. Abarca varias disciplinas que incluyen la Investigación de Operaciones y se espera que en los próximos años el uso de Minería de Datos este completamente integrado dentro de las disciplinas usadas. [67]

OLAP. (Online Analytical Processing) Es un conjunto de metodología y tecnologías que permite a los usuarios ver, navegar, manipular y analizar bases de datos multidimensionales.

Data Warehouse. Sistema para el almacenamiento y distribución de cantidades masivas de datos.

Dimensión. En una base de datos relacional o plana, cada campo en un registro representa una dimensión. En una base de datos multidimensional, una dimensión es un conjunto de entidades similares; por ejemplo en una base de datos de ventas podría identificarsee las dimensiones Producto, Cliente y Fecha de Venta.

Bibliografía

[1] J.R.Quinlan. Induction of decision trees. Publicación desconocida, 1986.

[2] J.R.Quinlan. C4.5: Programs for machine learning. Morgan Kaufmann, 1999.

[3] A.Maira S.Deshpande. Intelligent information in action. Technical report, Computer Associates International, Inc., 2003.

[4] D.Hand P.Smyth, H.Mannila. *Principles of DataMining, ISBN: 0-262-08290-X*. The MIT Press, 2001.

[5] Margaret Dunham. *Data Mining: Introductory and Advanced Topics*. Prentice Hall. ISBN: 0-130-88892-3, 2002.

[6] P. Bradley O. Mangasarian, U. Fayyad. *Mathematical programming for Data mining: Formulations and Challenges*. Technical report, Microsoft Research y Universidad de Wisconsin, 1998.

[7] N. Indurkhya S. Weiss. *Predictive Data Mining: A Practical Guide*. Morgan Kaufmann. ISBN: 1-558-60403-0, 1998.

[8] Peter Cabena. Pablo Hadjinian. Rolf Stadler. JaapVerhees. Alessandro Zanasi. *Discovering Data Mining: From Concept to Implementation*. Prentice Hall. ISBN: 0-137-43980-6, 1998.

[9] Jerome Friedman. *Datamining and stadistics.Whats the connection?* Universidad de Stanford, 1997.

[10] Grupo Gartner. *http://www.gartner.com*. Ultima visita 17/01/2006.

[11] P.Yu M.Chen, J.Han. *Datamining: An Overview from Database Perspective. Publicación desconocida*, 1996.

[12] Sushmita Mitra y Tinku Acharya. *Data Mining Multimedia, Soft Computing, and Bioinformatics*. John Wiley & Sons, Inc. ISBN: 0-471-46054-0, 2003.

[13] Mehmed Kantardzic. *Data Mining Concepts, Models, Methods and Algorithms*. John Wiley & Sons, Inc. ISBN: 0-471-22852-4, 2003.

[14] W. Thornthwaite. *Are you missing potential business opportunities because you're not exploring your data?* Intelligent Enterprise. *http://www.intelligententerprise.com*, 2005.

[15] Michael Berry y Gordon Linoff. *Data Mining Techniques for Marketing Sales and Customer Support*. Wiley Publishing, Inc. ISBN: 0-471-47064-3, 2da edition, 2004.

[16] CRoss Industry Standard Process for DataMining. *http://www.crisp-dm.org*. Ultima visita 31/10/2005.

[17] SAS Institute. Data Mining, The SEMMA Methodology and SAS Software, disponible en *http://www.sas.com*.

[18] Daniel T. Larose. *Discovering knowledge in data: an introduction to data mining*. John Wiley & Sons, Inc. ISBN: 0-471-66657-2, 2005.

[19] R.Agrawal R.Srikan. Fast algorithms for mining association rules. 20th VLDB Conference Santiago, Chile, 1994, 1994. IBM Almaden Research Center.

[20] J.Han M.Kamber. *Data Mining: Concepts and techniques*. Morgan Kaufmann. ISBN: 1-558-60489-8, 2000.

[21] Xiangji Huang. Clustering analysis and algorithms. In John Wang, editor, *Encyclopedia of Data Warehousing and Mining*, pages 159–164. Information Science Publishing, Idea Group Reference, ISBN: 2-59140-557-2, 2005.

[22] R.Ng y J.Han. Clarans: A method for clustering objects for spatial data mining. Technical report, IEEE Transactions on knowledge anda data engineering, Vol. 14, Nro. 5, Set/Oct 2002, 2002.

[23] J. B. MacQueen. *Some Methods for classification and Analysis of Multivariate Observations*. Technical report, Berkeley, University of California Press, 1967.

[24] T.Zhang M.Livny, R.Ramakrishnan. Birch: An efficient data clustering method for very large databases. Technical report, International Conference on Management of Data (SIGMOD),pag. 103-114, 1996.

[25] K.Shim S.Guha, R.Rastogi. Cure: an efficient clustering algorithm for large databases. Technical report, International Conference on Management of Data (SIGMOD),pag. 73-84, 1998.

[26] V.Kumar G.Karypis, E.Han. Chameleon: Hierarchical clustering using dynamic modeling. Technical report, IEEE Computing Society Vol.32 Nro.6 Pag 68-75, 1999.

[27] M.Ester et. al. A density-based algorithm for discovering clusters in large spatial databases with noise. Technical report, 2nd International Conference on Knowledge Discovery and Data Mining, 1996. Martin Ester, Hans-Peter Kriegel, Jorg Sander, Xiaowei Xu.

[28] Ankerst et. al. Optics: Ordering points to identify the clustering structure. Technical report, International Conference on Management of Data (SIGMOD),pag. 49-60, 1999. M.Ankerst, M.Breunig, H.Kriegel, J.Sander.

[29] W.Wang R.Muntz, J.Yang. Sting: A statistical information grid approach to spatial data mining. 23rd VLDB Conference Athens, Greece, 1997, 1998.

[30] Sheikholeslami Zhang, Chatterjee. Wavecluster: A multi-resolution clustering approach for very large spatial databases, 1998. 24th Very Large Data Base Conference, New York.

[31] Douglas Fisher. Knowledge acquisition via incremental conceptual clustering. In *Machine Learning 2*, pages 139–172. Kluwer Academic Publishers, 1987.

[32] P.Flynn A.Jain, M.Murty. Data clustering: A review. In *ACM Computing Surveys, Vol. 31, No. 3.* Setiembre 1999.

[33] Ian Witten and Eibe Frank. *Data mining: Practical Machine Learning Tools and Techniques.* Morgan Kaufmann, Inc. ISBN: 0-12-088407-0, 2da edition, 2005.

[34] Chaid. *http://www.statistics.com/content/glossary/c/chaid.php.* Ultima visita 31/10/2005.

[35] IBM. *http://www.ibm.com/us.* Ultima visita 31/10/2005.

[36] Lauren Fausset. *Fundamentals of Neural Networks.* Prentice Hall. ISBN: 0-133-34186-0, 1994.

[37] Ben Krose y P. Patrik van der Smagt. An introduction to neural networks. Publicación desconocida, 1993.

[38] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386-408, 1958.

[39] T. Kohonen. Selforganizad formation of topologically correct feature maps. Technical report, Biological Cybernetics, 1982.

[40] John H. Holland. Adaptation in natural and artificial systems. Universidad de Michigan, 1975.

[41] SAS®. Clementine®. *http://www.spss.com/clementine/*. Ultima visita 31/10/2005.

[42] SPSS Inc. *http://www.spss.com*. Ultima visita 31/10/2005.

[43] SAS®. Enterprise miner™. *http://www.sas.com/technologies/analytics/datamining/miner/* Ultima visita 31/10/2005.

[44] SAS®Institute. *http://www.sas.com*. Ultima visita 31/10/2005.

[45] Universidad de Waikato. Waikato environment for knowledge analysis. *http://www.cs.waikato.ac.nz/ml/weka/index.html*. Ultima visita 31/10/2005.

[46] Universidad de Waikato. *http://www.cs.waikato.ac.nz*. Ultima visita 31/10/2005.

[47] Bloor Research. *http://www.bloor-research.com*. Ultima visita 31/10/2005.

[48] Insightful. Insightful miner. *http://www.insightful.com/products/iminer/default.asp*. Ultima visita 10/11/2005.

[49] Insightful. *http://www.insightful.com/*. Ultima visita 10/11/2005.

[50] Salford Systems. Cart®. *http://www.salford-systems.com/cart.php*. Ultima visita 31/10/2005.

[51] Salford Systems. Mars®. *http://www.salford-systems.com/mars.php*. Ultima visita 31/10/2005.

[52] Salford Systems. *http://www.salford-systems.com*. Ultima visita 31/10/2005.

[53] Isoft. Alice. *http://www.isoft.fr/html/prod_alice.htm*. Ultima visita 31/10/2005.

[54] Isoft. *http://www.isoft.fr*. Ultima visita 31/10/2005.

[55] Cognos. Cognos b.i. *http://www.cognos.com*. Ultima visita 31/10/2005.

[56] Cognos. *http://www.cognos.com*. Ultima visita 31/10/2005.

[57] Oracle Corporation. Oracle. *http://www.oracle.com*. Ultima visita 31/10/2005.

[58] Microsoft Corporation. Sql server 2005. *http://www.microsoft.com/sql/prodinfo/overview/datasheet.mspx*. Ultima visita 01/12/2005.

[59] Microsoft Corporation. Microsoft. *http://www.microsoft.com*. Ultima visita 01/12/2005.

[60] Oracle. Oracle data mining suite. *http://www.oracle.com*. Ultima visita 31/10/2005.

[61] Robert Nisbet. How to choose a data mining suite. DM Direct Special Report, Marzo 2004.

[62] Two Crows Corporation. *Introduction to Data Mining and Knowledge Discovery*. Two Crows Corporation. ISBN: 1-892095-02-5, 3era edition, 1999.

[63] C. GiraudCarrier y O. Povel. Characterising data mining software. *Intelligent Data Analysis. IOS Press*, 7:181-192, 2003.

[64] KDnuggets^{TM}. *http://www.kdnuggets.com*. Ultima visita 31/10/2005.

[65] C.Lee y K.Irizarry. Adaptation in natural and artificial systems. IBM Systems Journal, Vol. 40, Nro. 2, 2001.

[66] Francisco Martínez et. al. Minería de datos en series temporales para la búsqueda de conocimiento oculto en hisóricos de procesos industriales. *Publicación desconocida*. Universidad de La Rioja, 2005.

[67] Marko Bohanec. *What is Decision Support?* Proc. Information Society IS-2001: Data Mining and Decision Support in Action!, 2001.

[68] Dorian Pyle. *Business Modeling and Data Mining*. Morgan Kaufmann. ISBN: 1-558-60653-X, 2003.

[69] Soumen Chakrabarti. *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan Kaufmann. ISBN: 1-558-60754-4, 2003.

[70] I. Davidson T. Soukup. *Visual Data Mining : Techniques and Tools for Data Visualization and Mining*. John Wiley & Sons, Inc. ISBN: 0-471-14999-3, 2002.

[71] Olivia Parr Rud. *Data Mining Cookbook: Modeling Data for Marketing, Risk and Relationship Management*. John Wiley & Sons, Inc. ISBN: 0-471-38564-6, 2001.

[72] T.Blaxton C.Westphal. *Data Mining Solutions: Methods and Tools for Solving Real-World Problem*. John Wiley & Sons, Inc. ISBN 0-471-25384-7, 1998.

[73] Sander et. al. Density-based clustering in spatial databases: A new algorithm and its applications. Technical report, Data Mining and Knowledge Discovery, Kluwer Academic Publishers, Vol.2, No. 2, 1998. Jorg Sander, Martin Ester, Hans-Peter Kriegel, Xiaowei Xu.

[74] JSR-73 Expert Group. *JavaTM Specification Request 73:JavaTM Data Mining (JDM). Versión 1.1*. Technical report, Oracle Corporation, 2005.

[75] Moshe Leshno Yoav Benjamni. *Statistical Methods for Data mining*. Universidad de Tel Aviv, Año de publicación desconocido.

[76] SAS®. Text miner^{TM}. *http://www.sas.com/technologies/analytics /datamining/textminer/*. Ultima visita 31/10/2005.

[77] IBM. Intelligent miner for data. *http://www.ibm.com/software/data/iminer/fordata/*. Ultima visita 31/10/2005.

[78] Statsoft e book. *http://www.statsoft.com/textbook/stathome.html*. Ultima visita 15/11/2005.