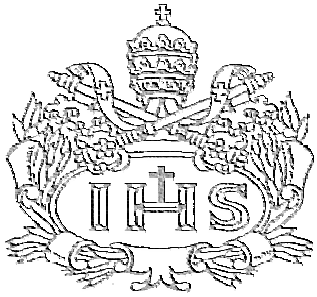
CONFRONTACIÓN DE DOS TÉCNICAS DE MINERÍA DE DATOS APLICADAS A UN DOMINIO ESPECÍFICO

MARIO GALVIS

FABRICIO MARTÍNEZ

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA DE SISTEMAS
BOGOTÁ D.C.
2004

CONFRONTACIÓN DE DOS TÉCNICAS DE MINERÍA DE DATOS APLICADAS A UN DOMINIO ESPECÍFICO



MARIO GALVIS

FABRICIO MARTÍNEZ

Proyecto de grado para optar el título de Ingeniero de Sistemas

Director:

Ing. Jorge E. Salazar Polanía

Ingeniero de Sistemas y Computación Mgr.

PONTIFICIA UNIVERSIDAD JAVERIANA

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA DE SISTEMAS

BOGOTA D.C.

2004

PONTIFICIA UNIVERSIDAD JAVERIANA

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA DE SISTEMAS

Rector Magnífico

Padre Gerardo Remolina Vargas S.J.

Decano Académico Facultad de Ingeniería

Ingeniero Roberto Enrique Montoya Villa

Decano del Medio Universitario Facultad de Ingeniería

Padre José Sarmiento Nova S.J.

Director Carrera de Ingeniería de Sistemas

Ingeniera Hilda Cristina Chaparro López

Director Departamento de Ingeniería de Sistemas

Ingeniero Germán Alberto Chavarro Flórez

Nota de Aceptación:

_____

_____

_____

_____

Director del proyecto

_____

JORGE E. SALAZAR POLANÍA

Jurado

_____

NOMBRE

Jurado

_____

NOMBRE

Bogotá D.C., 02 de diciembre de 2005

Artículo 23 de la Resolución No. 1 de Junio de 1946

"La Universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado.

Sólo velará porque no se publique nada contrario al dogma y la moral católica y porque no contengan ataques o polémicas puramente personales. Antes bien, que se vean en ellos el anhelo de buscar la verdad y la Justicia"

“Dedicamos este Proyecto de Investigación a nuestras familias a todos aquellas personas involucradas en el desarrollo de este proyecto”

Mario Galvis

Fabricio Martínez

AGRADECIMIENTOS

Damos nuestros agradecimientos a:

A nuestro tutor Ing. Jorge E. Salazar P., por acompañar siempre el desarrollo de este proyecto y por depositar su confianza en nosotros.

A Proyecto Orbis, por facilitarnos el acceso a la información que poseían y por su confianza en nosotros en el manejo de la misma.

A Ubiquando, por brindar la orientación técnica necesaria durante la elaboración del proyecto.

Mario Galvis

Fabricio Martínez

TABLA DE CONTENIDOS

INTRODUCCIÓN

Por ser la Minería de Datos una herramienta que día a día cobra mayor importancia en diversos entornos de nuestra sociedad, desde el académico, pasando por el comercial, hasta el de investigación y desarrollo, este Proyecto de Investigación es importante pues presenta una oportunidad a quienes deseen hacer uso de esta herramienta que ofrece la tecnología.

Este Proyecto de Investigación pretende mostrar las principales diferencias entre dos técnicas de Minería de Datos, árboles de decisión y reglas de asociación.

Para esto se tienen a disposición los siguientes recursos. Primero, un conjunto de datos con información histórica de estudiantes de la Pontificia Universidad Javeriana y de su desempeño académico detallado. Segundo, una herramienta de Minería de Datos llamada WEKA (Waikato Envioronment for Knowledge Analysis) desarrollada en el entorno académico que permitirá la ejecución de los algoritmos seleccionados y la posterior comparación de los resultados obtenidos para cada uno de ellos.

El primer capítulo contiene un marco teórico del proyecto, que informa acerca del estado del arte de la Minería de Datos y el cual contiene la información teórica necesaria para la elaboración de esta investigación. El siguiente capítulo es una compilación de la metodología utilizada para la realización del proyecto, la cual comprende una descripción de los datos de origen utilizados, las pruebas a realizar sobre los mismos y cómo la herramienta de Minería de Datos utilizada fue implementada en este proceso. Así mismo se da una reseña de la manera como se formularon las preguntas o consultas para el proyecto.

Posteriormente se muestra en el capítulo tres los resultados obtenidos tras la aplicación de las pruebas descritas en el anterior capítulo, así como un análisis a nivel global de estos resultados. Los resultados obtenidos en el capítulo tres, permitirán la extracción de conclusiones que serán el contenido del capítulo cuatro.

1. MARCO TEÓRICO

A lo largo de este capítulo se pretende mostrar los conceptos básicos de la Minería de Datos, así como de las técnicas y tecnologías que permiten su desarrollo y aplicación.

Primero, se hará una breve introducción al tema y del porque se ha convertido en una herramienta de importancia en la actualidad. Posteriormente, se hará un recuento de los antecedentes que llevaron al desarrollo de la Minería de Datos. También se revisará el estado del arte, es decir, cual es la situación actual de la Minería de Datos y cuales son los últimos avances en esta materia. Se repasarán sus aplicaciones más comunes en ambientes industriales y comerciales, entre otros.

Se mostrará cuál es el proceso de aplicación de la Minería de Datos en un caso de estudio real, así como algunas de las técnicas utilizadas para su implementación, específicamente, las técnicas de árboles de decisión y de reglas de asociación, que serán el objeto de este estudio. Por último, se hará un recuento de los criterios que permiten la comparación de algoritmos y que serán utilizados posteriormente para efectuar la comparación de las dos técnicas, que son en esencia algoritmos y como tales, pueden ser comparados por estos criterios.

1.1 INTRODUCCIÓN

A través de la Historia se puede observar la evolución de la humanidad bajo la adquisición del conocimiento. Cada vez se mejoran las actividades humanas gracias al conocimiento que se adquiere a través de la experiencia del mismo ser humano. Esta evolución brinda un mejor modo de vivir, buscando satisfacer todo tipo de necesidad. Todo surge de la relación causa – efecto que es la base de toda experiencia. Y todo esto proviene de los datos que nacen a partir de las acciones que se toman. Estos datos se convierten en información que es al final la que sirve para la toma de decisiones en cualquier tipo de escenario. Este es el proceso normal que se hace diariamente en la vida del ser humano.

Hablando más en términos informáticos, se tienen bases de datos donde se guardan los datos y la información de una empresa o persona. De la misma manera se construyeron las bodegas de datos donde se guardan todas las transacciones que se tengan a través del tiempo, formando así una base de datos histórica donde se pueden hacer profundos análisis de la información.

¿Pero qué se puede conocer a partir de la información suministrada por una gran cantidad de datos? La empresa sería conciente de todos los movimientos que se tienen, gustos de los clientes, frecuencia de compras, temporadas de transacciones, qué tipo de productos se vende con otros productos. Claramente se puede notar que se está hablando como ejemplo de un supermercado, algún negocio de venta de productos tangibles, pero este tipo de conocimiento abarca todas las áreas.

Ahora bien, todo este conocimiento va a dar como resultado el objetivo mismo de la información, la toma de decisiones. Se realizan estudios y transformaciones de datos para que se pueda tener una mejor capacidad de tomar decisiones a partir de los resultados.

¿Pero se tiene el mayor provecho de la información que se almacena en estos grandes repositorios de datos? En este punto es donde surge la Minería de Datos como tecnología para el desarrollo y el descubrimiento de la información muchas veces oculta en los mismos datos. Información que muchas veces no se tiene en cuenta y que en determinados casos puede ser valiosa y crítica para un mejor conocimiento del negocio y aportar mayor base a la toma de decisiones en cualquier tipo de escenario.

1.2. ANTECEDENTES

La Minería de Datos reúne unas cuantas áreas como la Estadística, la Inteligencia Artificial, la Computación Gráfica, las Bases de Datos y el procesamiento masivo y usando como materia prima las bases de datos.

La idea de Minería de Datos no es nueva. Ya desde los años sesenta los estadísticos manejaban términos como *data fishing*, *data mining* o *data archaeology* con la idea de encontrar relaciones existentes en los datos en bases de datos con ruido. A principios de los años ochenta se empezaron a fortalecer los términos de la Minería de Datos. A principios de los años ochenta sólo existían un par de empresas dedicadas a este estudio. En el 2002 existen más de 100 empresas en todo el mundo que utilizan la Minería de Datos. Las listas de discusión sobre este tema las forman investigadores de más de ochenta países. Esta tecnología ha sido un buen punto de encuentro entre personas pertenecientes al ámbito académico y al de los negocios.

La tecnología informática se ha convertido en fundamental para las grandes organizaciones. Actualmente permite registrar con lujo de detalle, los elementos de todas las actividades con facilidad. Las bases de datos permiten almacenar cada transacción, y otros elementos que reflejan la interacción de la organización con todos sus integrantes, ya sean otras organizaciones, sus clientes, sus divisiones o sus empleados. Se tiene un registro bastante completo

del comportamiento de la organización. Pero, ¿cómo traducir estos datos en experiencia, sabiduría corporativa y conocimiento que apoye la toma de decisiones, especialmente al nivel gerencial que es el destino de las grandes organizaciones? ¿Cómo se comprende este fenómeno, si se toman en cuenta grandes volúmenes de datos?

Como ejemplo: ¿Cuántas transacciones se realizan en un banco? Actualmente, un banco cuenta (posiblemente) con cientos de sucursales. Al mismo tiempo, con diferentes medios en los que se realizan transacciones, compra de tarjeta de crédito, manejo de inversiones y herramientas de mercados electrónicos. Se puede notar una gran diferencia con los primeros bancos de la historia, hoy en día es imposible para los tesoreros guardar en una libreta o en la memoria los datos de cada transacción que realiza un cliente. Se tienen que basar en sistemas de información para guardar este tipo de información. Estos sistemas de información permiten obtener resúmenes, reportes y distintas formas de ver el estado de cuentas de la empresa para así tener un mayor conocimiento y cubrimiento del estado del negocio. Seguramente, muestren también reportes que sugieren estrategias a futuro o condiciones del nicho de mercado del negocio, que en definitiva proveen argumentos para generar la planeación y estrategias de la empresa.

En este volumen de datos, ¿qué conoce la organización sobre sus clientes?. Puede conocer todo o puede que no conozca nada. El no conocer nada se debe a que un cliente que va a una sucursal a la que no acostumbra ir normalmente puede que obtenga información vaga e impersonal de su cuenta que como conclusión genera que el banco le brinde un servicio impersonal al cliente. No tendría elementos para tratarlo como cliente fiel. ¿No se podría inferir ciertos elementos para dar un trato más personal a cierto tipo de clientes? Según el cliente y sus condiciones, se esperan distinto tipo de servicio, la edad o la situación económica de la persona podría ser un aspecto que diferenciaría un cliente de otro.

La Minería de Datos surge del análisis de grandes volúmenes de información, con el fin de obtener conocimiento que apoye la toma de decisiones y que contribuya a la construcción de la experiencia a partir de millones de transacciones que registra una corporación en sus sistemas de información.

La Minería de Datos es más efectiva cuando los datos tienen características que permitan una interpretación de acuerdo a la experiencia humana, espacio y tiempo. Por ejemplo, qué productos se venden mejor en la temporada de vacaciones, en qué regiones es productivo sembrar maíz.

La tecnología ofrece analizar estos grandes volúmenes de datos y reconocer patrones en tiempo y espacio, que resultará en un modelo claro para soportar la toma de decisiones.

1.3. ESTADO DEL ARTE

La Minería de Datos es el proceso de examinar exhaustiva y minuciosamente inmensas cantidades de datos a fin de identificar, extraer y descubrir nuevos conocimientos, de forma automática. La Minería de Datos es una herramienta exploradora y no explicativa. Es decir, explora los datos para sugerir hipótesis.

La Minería de Datos también puede ser entendida como proceso analítico diseñado para explorar grandes volúmenes de datos (generalmente datos de negocio y mercado) con el objeto de descubrir patrones y modelos de comportamiento o relaciones entre diferentes variables. Esto permite generar conocimiento que ayuda a mejorar la toma de decisiones en los procesos fundamentales de un negocio.

La Minería de Datos permite obtener valor a partir de la información que registran y manejan las empresas, lo que ayuda a dirigir esfuerzos de mejora respaldados en datos históricos de diversa índole.

Los objetivos de la Minería de Datos son:

Descripción de clases: provee una clasificación concisa y resumida de un conjunto de datos y los distingue unos de otros. La clasificación de los datos se conoce como caracterización, y la distinción entre ellos como comparación o discriminación.

Asociación: es el descubrimiento de relaciones de asociación o correlación en un conjunto de datos. Las asociaciones se expresan como condiciones *atributo-valor* y deben estar presentes varias veces en los datos.

Clasificación: analiza un conjunto de datos de entrenamiento cuya clasificación de clase se conoce y construye un modelo de objetos para cada clase. Dicho modelo puede representarse con árboles de decisión o con reglas de clasificación, que muestran las características de los datos. El modelo puede ser utilizado para la mayor comprensión de los datos existentes y para la clasificación de los datos futuros.

Predicción: esta función de la minería predice los valores posibles de datos faltantes o la distribución de valores de ciertos atributos en un conjunto de objetos.

Clustering o agrupación: identifica *clusters* o grupos en el conjunto de datos, donde un *cluster* es una colección de datos "similares". La similitud puede medirse mediante funciones de distancia, especificadas por los usuarios o por expertos. La Minería de Datos trata de encontrar *clusters* de buena calidad que sean escalables a grandes bases de datos y a bodegas de datos multidimensionales.

Análisis de series a través del tiempo: analiza un gran conjunto de datos obtenidos con el correr del tiempo para encontrar en él regularidades y características interesantes, incluyendo la búsqueda de patrones secuenciales, periódicos, modas y desviaciones.

La Minería de Datos se encuentra en pleno desarrollo y aplica a varias disciplinas como las bases de datos, estadística, bodegas de datos, visualización de datos y obtención de información. También se utilizan métodos de las áreas de reconocimiento de patrones, redes neuronales, análisis espacial de datos, y procesamiento de señales. La Minería de Datos se muestra como un proceso interdisciplinario donde diferentes ramas pueden intervenir para obtener un mayor provecho de este conocimiento.

Para procesar grandes volúmenes de datos donde deben extraerse patrones automáticamente, se debe contar con una gran capacidad computacional y equipos de alto desempeño.

1.4. APLICACIONES

Algunas de las aplicaciones más comunes o reconocidas a nivel industrial o comercial se encuentran en los siguientes segmentos:

Sector Industria

- ***Optimización de Centrales Eléctricas***

  Aplicación de Minería de Datos al control y optimización de centrales térmicas mediante el desarrollo de un sistema de optimización/control de procesos complejos. El sistema ha sido aplicado con éxito en varias centrales térmicas, consiguiendo una mejora superior al 2% en el consumo de combustible.

- ***Control de Trenes de Laminado en la Industria del Acero***

  Aplicación de Minería de Datos en trenes de laminado de acero consistente en la determinación precoz de la fuerza necesaria para laminar una bobina de acero en un tren de bandas en caliente a partir de ciertas propiedades del acero entrante y de condiciones de salida deseadas

- ***Optimización de la producción de cartón en la Industria Papelera***

  Optimización del proceso de producción de cartón mediante el mejoramiento del rendimiento de la fabricación de cartón optimizando el control de la velocidad de la línea, mediante la utilización de técnicas de Minería de Datos.

Sector Farmacéutico y Sanitario

- ***Predicción de Ventas de Productos Farmacéuticos***

  Predicción de ventas a través del desarrollo de un modelo para predecir las ventas de un producto en un determinado mes, basándose en datos sobre las ventas en meses previos. La Minería de Datos es ampliamente utilizada en esta área por empresas comerciales y existe una amplia gama de aplicaciones implementadas.

Administración Pública y Servicios

- ***Análisis y Control de Tráfico de Vehículos***

  Aplicación de Minería de Datos para análisis del estado del tráfico en carretera con el desarrollo de sistemas para clasificación del estado del tráfico.

Sector Financiero y del Seguro

- ***Estimación de Riesgos en la Concesión de Seguros***

  Los estudios para realizar la concesión de pólizas de seguros origina grandes gastos para las compañías de esta rama. Mediante el desarrollo de sistemas para análisis de créditos que modele la forma en que los expertos humanos analizan las empresas, con técnicas de Minería de Datos, permite la óptima asignación de estos recursos.

- ***Detección y Control de Fraude en el uso de Tarjetas de Crédito***

  Aplicación de Minería de Datos en detección y control del fraude en el uso de tarjetas de crédito mediante el análisis de los atributos característicos de las transacciones fraudulentas y desarrollo de sistemas para su identificación y detección.

- ***Segmentación de Clientes de Entidades Financieras***

  Aplicación de Minería de Datos para segmentación de clientes de entidades financieras.

  Esta segmentación de los clientes (por ejemplo de un banco) mediante un modelo basado en agrupamiento (*clustering*), permite discernir el comportamiento de los clientes en la actualidad, así como de las tendencias que se han presentado en el tiempo.

1.5. PROCESO DE IMPLEMENTACIÓN DE MINERÍA DE DATOS

1.5.1. Importancia de los datos

El primer punto importante en la implementación de la Minería de Datos, es tener la conciencia de la importancia de los datos y de su almacenamiento. Si no se tiene desarrollada esta conciencia, la calidad de los datos que se puedan almacenar no será la mejor y por tanto, no se podrá efectuar el proceso de Minería de Datos de manera eficiente, útil y rápida.

Problemas comunes con los datos
Algunos de los problemas que se pueden presentar en el almacenamiento de los datos son:

- El mantenimiento e ingreso de grandes cantidades de información, puede ser un proceso difícil y engorroso.

- Los datos se encuentran dispersados en distintas ubicaciones, tanto físicas como lógicas (cuando por ejemplo pertenecen a distintas organizaciones).

- Se pueden tener distintos métodos y dispositivos de recopilación.

- Solo una porción de los datos que se tienen darán una verdadera utilidad para la organización. Comúnmente se cumple que el 20% de los datos, dé un 80% del total de la utilidad.

- A pesar de tener grandes cantidades de datos, pocas veces se logra una adecuada interpretación de los mismos que facilite la toma de decisiones a quienes no les interesa el detalle de los datos.

Debido a estos problemas y a todos los que se pueden presentar en una organización, estas deben tener una solución de administración de datos efectiva y eficiente. En la administración de datos, se manejan dos conceptos de importancia: primero, el concepto de carga, que se refiere a lo en este proyecto se llama datos de origen y el segundo concepto se refiere al uso en información y conocimiento que se les pueda dar a estos datos y que se traduce en poder. Es en este concepto que se enfoca la Minería de Datos y por tanto esta investigación.

El ciclo de vida de los datos dentro de una organización y del descubrimiento de conocimiento, puede ser dividido en las siguientes etapas.

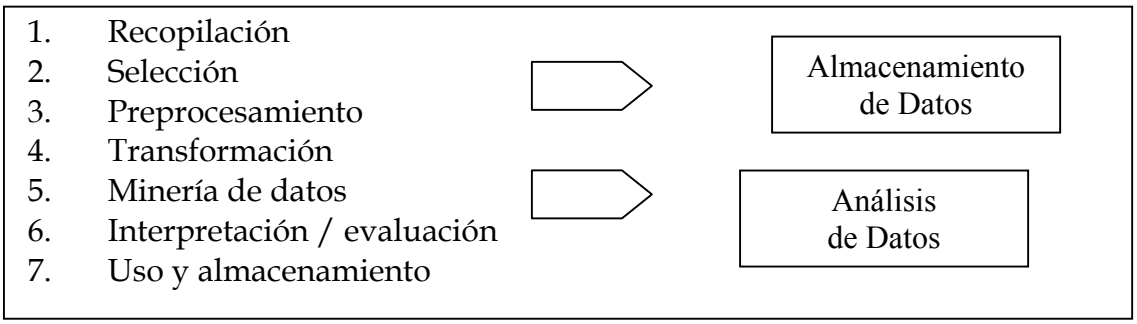


Ilustración 1 - Etapas del descubrimiento del conocimiento

Calidad de los datos

Otro punto de gran importancia en el campo de los datos de origen, es su calidad. A continuación se da una definición precisa de lo que significa el término "calidad de los datos" (también conocida como CD).

La calidad de los datos puede estar dada en cualquiera de las siguientes cuatro categorías:

1. Calidad intrínseca de los datos: hace referencia a la exactitud, objetividad, credibilidad y reputación de los datos.

2. Calidad de los datos por su accesibilidad: analiza los datos según accesibilidad y la seguridad en su acceso.

3. Calidad contextual de los datos: toma de los datos su pertinencia, valor agregado, oportunidad, consistencia y cantidad.

4. Calidad de los datos de representación: interpretabilidad, facilidad, comprensión, representación concisa y consistente de los datos recopilados.

Tecnologías Relacionadas

Algunas tecnologías modernas que se relacionan con la Minería de Datos, son las siguientes:

1. Data Warehouse: almacenes de datos, que resultan útiles para organizaciones que guardan grandes cantidades de información, comúnmente información relacionada con transacciones, para que sean utilizadas directamente o para que se haga algún proceso de transformación o análisis sobre los mismos. Muchas veces son utilizados y recomendados para la implementación de la Minería de Datos.

2. Bases de Datos Multimedia: aparte de los tipos de datos tradicionales que permiten almacenar las bases de datos relacionales que actualmente se encuentran en el mercado (cadenas de caracteres, enteros, booléanos, etc.), estas permiten almacenar tipos de datos multimedia que facilita su visualización.

1.5.2. Pasos del proceso

La Minería de Datos como ya se ha mencionado con anterioridad se refiere al proceso de extracción de patrones de alguna manera inesperados o al menos no fácilmente discernibles a simple vista.

El proceso de Minería de Datos consta de tres etapas fundamentales:

1. Exploración, integración y limpieza de los datos:

   En primer lugar se tiene el proceso de exploración de datos, que consiste en un proceso en el que se hace uso de una metodología estructurada con el fin de descubrir y evaluar problemas apropiados, definir soluciones de manera que al final del proceso se tengan resultados de utilidad.

   Dentro de esta etapa se tiene una serie de actividades. Dentro de estas actividades se debe tener en cuenta que las que revisten mayor importancia para el éxito de un proyecto de Minería de Datos son las de exploración de los espacios tanto del problema como de la solución y la especificación del método de implementación, aunque es proporcionalmente menor el tiempo necesario para llevarlas a cabo que la actividad de minería propiamente dicha.

   - Explorar el espacio del problema: dentro de esta actividad se pueden encontrar algunos de los pasos más importantes para la realización de la Minería de Datos.

Primero, se deben identificar con precisión los problemas que se desean resolver mediante la implementación de la Minería de Datos. Muchas veces puede ocurrir que las personas u organizaciones tengan ideas preconcebidas de cuál es el problema existente, por lo tanto se realizan modelos de los datos que pueden ser muy precisos y útiles para otros temas, pero no para el de interés. Por lo anterior, un análisis concienzudo en este aspecto al inicio del proyecto, ahorrará tiempo, recursos y esfuerzos en etapas posteriores. En caso tal que se presente una gran variedad de problemas a resolver y se tengan recursos limitados para afrontarlos, se debe elegir un subconjunto de problemas a resolver. Para hacerlo, se puede utilizar técnicas como *parwise ranking* o la utilización del método de Paretto, para seleccionar este subconjunto.

Posteriormente a la identificación del problema, se debe realizar una definición precisa del mismo. Esta definición debe contener todos los componentes o partes de la organización que se ven involucrados en el mismo, variables que intervienen, marcos de tiempo, y en general, cualquier aspecto de relevancia para la definición formal del problema.

- Explorar el espacio de la solución: después de decidir qué tipo de problema se desea resolver, se debe poder establecer clara y completamente, qué tipo de resultados se desean obtener a partir de la utilización de la Minería de Datos. Estos resultados no son objetivos específicos. Pueden incluir cuadros, gráficas, reportes, listas de registros, código de programación, fórmulas algebraicas, etc.

  En el caso de esta investigación, los resultados de la Minería de Datos, se darán en forma de reportes o informes del comportamiento de los estudiantes de la facultad o de la influencia de algunos factores en su desempeño, entre otros.

- Especificar el método de implementación: es el paso final para especificar en detalle, cómo las distintas soluciones a los problemas seleccionados van a ser aplicadas en la práctica. Normalmente, debe haber una persona de alto nivel en la organización involucrada en el proyecto, de manera que no se quede en intenciones e investigaciones, sino que se realice un plan de implantación para que se lleve a cabo.

- Realizar la minería sobre los datos

i. Preparar los datos:
Prepara los datos para lograr una extracción más eficiente y rápida de los resultados deseados. Es una de las etapas más importantes e incluye la integración, la limpieza de los datos y la discretización de las tablas de origen.

- Integración
  Primero, la integración comprende el reunir los datos de origen si se encuentran físicamente separados o si se trata de bases de datos distintas. En segundo lugar, como origen de datos casi siempre se tiene una base de datos relacional, que puede tener cualquier tipo de información. Sin embargo, para lograr que las herramientas de Minería de Datos hagan su trabajo, se les debe proporcionar un conjunto de datos "pre-procesado". En el caso de esta investigación, se deben realizar sentencias SQL que permitan la extracción y unión de los datos que se consideran relevantes para la resolución de una de las preguntas planteadas.

- Limpieza
  Los datos no son precisos o puros por varias razones. Pueden presentarse errores de medición (en el caso que una regla que no se ubique bien), errores de precisión (cuando se trunca la medida a metros, centímetros o milímetros, en algún punto se pierde precisión).

  Por esta razón debe implementarse un proceso de examen y limpieza de los datos, a fin de lograr que reflejen con la mayor precisión posible, el dominio de aplicación al que corresponden.

ii. Investigar los datos:
Se pretende contestar tres preguntas:

- ¿Qué hay en el conjunto de datos?
  Se parte de una base de tablas que contienen un conjunto de vasta información. Sin embargo, parte del proceso es encontrar la porción de los datos que será de real utilidad para lograr los

objetivos del proyecto. Es en este punto donde es de vital importancia el que se tenga un amplio conocimiento del dominio, de tal manera que se pueda determinar qué tablas y columnas serán relevantes.

- ¿Qué riesgos plantea el conjunto de datos?
  El hecho de tener a disposición los datos no quiere decir (como se vio en la sección "Limpieza de los datos") que estos sean aptos para extraer conclusiones que sean de utilidad para los interesados. Si los datos de origen tienen una "calidad" muy deficiente, la ejecución de un proyecto de Minería de Datos sobre los mismos resultará una tarea muy compleja.

iii. Modelar los datos:
Aplicación de las técnicas o herramientas de minería, para lograr los resultados deseados.

2. Definición de patrones o construcción de modelos

   Es en esta etapa que se aplican los algoritmos y técnicas que se explican a fondo en los siguientes apartados de este marco teórico.

   La forma de aplicación y su funcionamiento, están dados por cada uno de los algoritmos y se estudiarán en su descripción correspondiente.

3. Validación y verificación de los modelos

   Una vez se hayan creado los modelos a partir de los datos de origen, utilizando las herramientas de Minería de Datos a disposición, se debe pasar a la etapa de verificación y validación de dichos modelos.

   Esta etapa consiste en el contraste de estos modelos con los datos de origen, para saber en qué medida se ajustan a la realidad (no se

puede afirmar que los datos de origen –una mera base de datos- sean un fiel reflejo de la realidad).

1.6. TÉCNICAS PARA LA EJECUCIÓN DE MINERÍA DE DATOS

Existen dos tipos de técnicas para la implementación de la Minería de Datos:

1. Técnicas clásicas: estadística, vecindades (*neighborhoods*) y *clustering*
2. Técnicas de nueva generación: árboles, redes y reglas

A continuación, se explicarán estas técnicas y algunas de sus implementaciones más conocidas.

1.6.1. Técnicas Clásicas:

- **Estadística**

Aunque la Estadística en su definición estricta difiere de la Minería de Datos, ésta estaba siendo utilizada para este fin, mucho antes de que existiera el término "Minería de Datos".

La idea era utilizar técnicas estadísticas para descubrir patrones y construir modelos predictivos. Hoy en día, las aplicaciones de Minería de Datos que utilizan estadística, quieren resolver preguntas como estas:

- ¿Qué patrones existen en los datos?
- ¿Cuál es la probabilidad de que un evento ocurra?
- ¿Qué patrones son significativos?
- ¿Cuál es un resumen de alto nivel de los datos, que da una idea de lo que está contenido en la base de datos?

Por medio del uso de histogramas, se puede hacer un resumen, que generalice lo que está ocurriendo en los datos.

La predicción también es un elemento importante en la estadística, y es por eso que se utilizan métodos como la regresión, que permiten este tipo de análisis. El ejemplo más claro, es el uso de la regresión lineal para predecir valores de una variable dada.

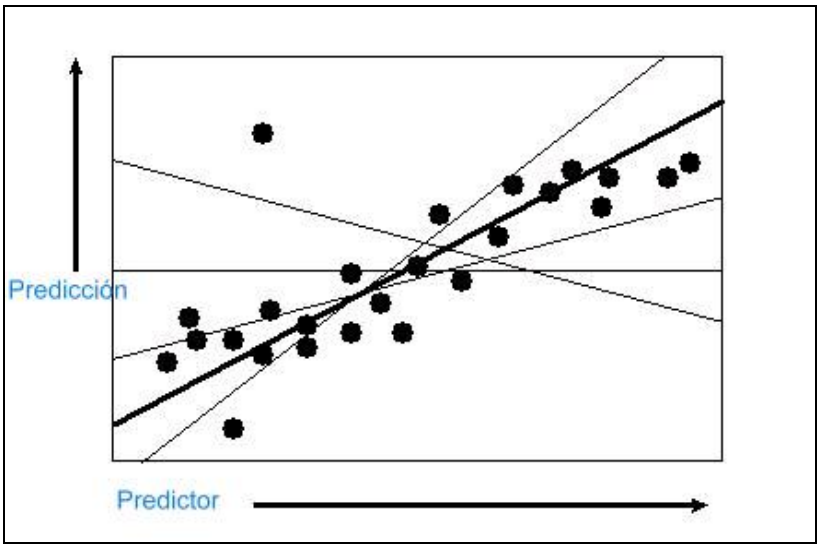


Ilustración 2 - Regresión lineal. Predicción vs. Predictor

La regresión lineal es similar a la tarea de encontrar la línea que minimice la distancia total a un conjunto de puntos.

El método toma un valor para el predictor y encuentra el valor correspondiente para su predicción. La idea es tener un modelo que minimice el error para cualquier estimación.

Sin embargo, este método es útil cuando los valores contenidos en la base de datos siguen una tendencia lineal. Para los casos en que esto no se cumple, se pueden hacer estas modificaciones:

- Se puede usar más de un predictor
- Se pueden aplicar transformaciones a los predictores
- Se puede multiplicar varios predictores y usarlos como términos de una ecuación

Agregar más predictores, puede hacer líneas más complicadas, que tienen en cuenta más información y por tanto, pueden hacer mejores predicciones (líneas en varias dimensiones).

- **Vecino más cercano**

La técnica del "vecino más cercano" es muy parecida al "clustering", que será explicado más adelante. Su esencia es que para poder predecir el valor de un registro, busca por registros con valores similares de predictor en la base de datos. El método también debe agrupar los registros de la base de datos, para encontrar cuáles podrían ser vecinos y cuáles no.

Un ejemplo sencillo, sería un barrio de la vida real. Si se quiere saber cuales con los ingresos de una persona en especial, se podrá hacer una estimación bastante aproximada, conociendo el valor de los ingresos de las personas que viven cerca de él.

Este tipo de técnicas están entre las más fáciles de entender y usar. Ha sido ampliamente usado en algoritmos de recuperación de texto. Cuando un usuario encuentra un documento de su interés y después solicita otros documentos parecidos, se debe encontrar similitudes en los documentos y después encontrar el más parecido de acuerdo a estos criterios.

Una mejora usual a este algoritmo, es tener en cuenta un número "k" de vecinos para clasificar un registro no clasificado.

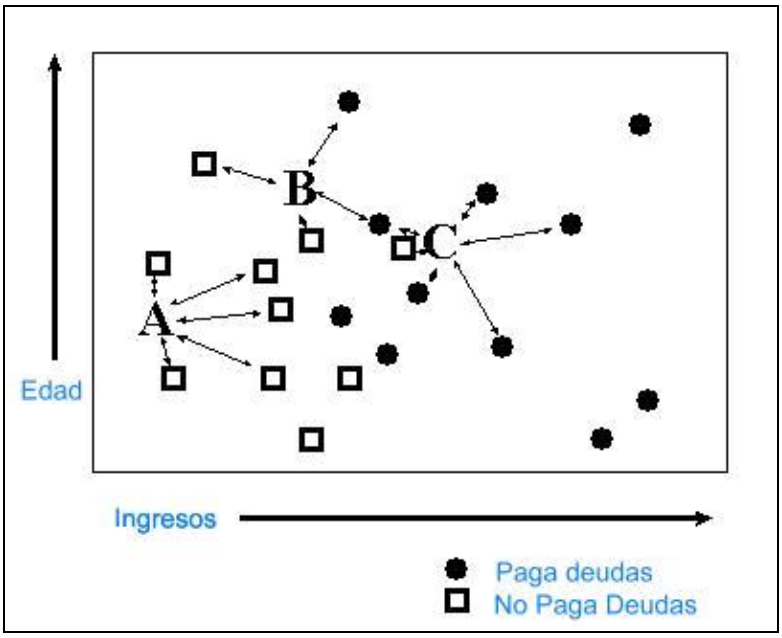


Ilustración 3 - Vecino más cercano. Ejemplo de aplicación.

Puede ser aplicado para saber si otorgarle un crédito a un cliente C, o no, a partir de la información de sus vecinos más cercanos (de acuerdo a edad e ingresos por ejemplo).

- ***Clustering***

Clustering es el método mediante el cual se agrupan registros. Usualmente este método es utilizado para darle al usuario final una visión más general de lo que ocurre en la base de datos. Después de haber trabajado algún tiempo con estas clasificaciones, se podrán a empezar a hacer predicciones de lo que pasará con uno de los grupos ante un estimulo determinado.

Esta técnica también ayuda a encontrar los registros que se destacan del resto. Si se hace un grupo (cluster) con ciertas características, quienes pertenezcan al grupo generalmente cumplirán con otras características.

Cuando se utiliza más de un criterio para clasificar los registros en grupos, entonces se debe dar niveles de prioridad a estos criterios, para que puedan ser aplicados simultáneamente. Cada uno de los criterios (columnas de la base de datos), corresponde a una dimensión del espacio y a un predictor. Es por eso que en este tipo de problemas se habla de espacios n-dimensionales.

En resumen:

Vecino más cercano	Clustering
Usado para predicción y consolidación	Usado mayormente para consolidación de datos en agrupaciones
Generalmente solo usa métricas de distancia para medir cercanía	Puede usar otras métricas

Tabla 1 - Vecino más cercano vs. Clustering

1.6.2. Técnicas de Nueva Generación:

- **Árboles de decisión**

Es un modelo predictivo que, como lo dice su nombre, puede ser visto como una estructura en forma de árbol. Es uno de los algoritmos más populares.

Este algoritmo puede ser usado para identificar individuos que tengan mayor probabilidad de hacer clic sobre un "banner" de publicidad o de comprar un producto específico de un sitio de e-commerce.

Esta es una de las técnicas elegidas para el desarrollo de la investigación por lo tanto se dará una explicación más profunda en el siguiente apartado del marco teórico.

- **Redes Neuronales Artificiales**

Son programas de computadores que implementan sofisticados algoritmos para la detección de patrones y de aprendizaje para construir modelos predictivos a partir de una gran base de datos histórica. Los resultados de una red neuronal se dan en forma numérica y en ocasiones es muy difícil de entender. Se han usado para la detección de fraudes en tarjetas de crédito, en asuntos militares y en pilotos automáticos, entre otros.

Las redes neuronales se han diseñado para simular en cierto modo el funcionamiento del cerebro humano y su capacidad para aprender. Para esto se usaron dos estructuras básicas.

El nodo = Simula a las neuronas humanas.
El vinculo = Simula la conexión entre las neuronas.

Las redes neuronales para predecir se basan en los nodos de entrada, los cuales envían información que luego es multiplicada por lo vínculos y luego se une esta información en un nodo final que se le llama el nodo de salida. Una función podría ser aplicada a esta información que nos dará la predicción (0 significa que hay riesgo de fraude en la tarjeta de crédito y 1 significa que todo esta bien).

- **Reglas**

Las reglas son en lo que más se piensa cuando se habla de Minería de Datos. Se trata de buscar en una base de datos reacciones que me generen una regla que quizá nadie se había dado cuenta con anterioridad. ("Buscar oro"). Esto se hace dando significado a los datos y que tan repetitivos se producen.

Las reglas son un proceso automatizado y probablemente es la mejor técnica para la Minería de Datos para encontrar posibles patrones predictivos en un base de datos. Cuando se aplican las reglas se pueden llegar a conclusiones que no se habían pensado.

Esta es una de las técnicas elegidas para el desarrollo de la investigación por lo tanto se dará una explicación más profunda en la siguiente sección del marco teórico.

1.7. REGLAS DE ASOCIACIÓN

Las reglas de asociación son parecidas a las reglas de clasificación. Tienen un lado izquierdo con condicionales que debe cumplir, y un lado derecho con las consecuencias de cumplir estas condiciones.

Se encuentran también usando un procedimiento de *covering* (o cobertura). Sin embargo, en el lado derecho de las reglas, puede aparecer cualquier par o pares atributo-valor. Para encontrar ese tipo de reglas se debe de considerar cada posible combinación de pares atributo-valor del lado derecho. Para posteriormente podarlas usando cobertura (número de instancias predichas correctamente) y precisión (proporción de número de instancias a las cuales aplica la regla). En reglas de asociación, la cobertura también es llamada soporte (*support*) y la precisión también es llamada confianza (*confidence*).

En términos de probabilidades, el soporte y la confianza están dados por las siguientes fórmulas:

$$\text{soporte}(A \Rightarrow B) = P(A \cup B)$$

$$\text{confianza}(A \Rightarrow B) = P(B|A) = \frac{\text{soporte}(A \cup B)}{\text{soporte}(A)}$$

La fórmula de soporte esta dada por la unión de dos probabilidades, es decir, el soporte de la regla (A → B) es equivalente a probabilidad que se cumplan simultáneamente A y B.

Por otro lado, la fórmula de confianza anterior esta dada en términos de probabilidades condicionales. Esta fórmula podría interpretarse así: probabilidad de que ocurra B, dado que ocurre A, y su equivalente en términos de soporte se muestran también en la fórmula.

En realidad se está interesado únicamente en reglas que tienen mucho soporte, por lo que se busca (independientemente de que lado aparezcan), pares atributo-valor que cubran una gran cantidad de instancias. A estos, se les llama *item-sets* y a cada par atributo-valor ítem. Un ejemplo típico de reglas de asociación es el análisis de la canasta de mercado. Básicamente, encontrar asociaciones entre los productos de los clientes, las cuales pueden impactar a las estrategias mercadotécnicas.

Cuando se tienen todos los conjuntos, se transforman en reglas con la confianza mínima requerida.

Algunos ítems producen más de una regla y otros no producen ninguna. Se tiene el siguiente ejemplo:

Ambiente	Temp.	Humedad	Viento	Clase
soleado	alta	alta	no	N
soleado	alta	alta	si	N
nublado	alta	alta	no	P
lluvia	media	alta	no	P
lluvia	baja	normal	no	P
lluvia	baja	normal	si	N
nublado	baja	normal	si	P
soleado	media	alta	no	N
soleado	baja	normal	no	P
luvia	media	normal	no	P
soleado	media	normal	si	P
nublado	media	alta	si	P
nublado	alta	normal	no	P
lluvia	media	alta	si	N

Tabla 2 - Datos ejemplo – reglas de asociación

Con los datos de la tabla, el itemset:

humedad=normal, viento=no, clase=P

Puede producir las siguientes posibles reglas:

If humedad=normal and viento=no Then clase=P 4/4
If humedad=normal and clase=P Then viento=no 4/6
If viento=no and clase=P Then humedad=normal 4/6
If humedad=normal Then viento=no and clase=P 4/7
If viento=no Then clase=P and humedad=normal 4/8
If clase=P Then viento=no and humedad=normal 4/9
If true Then humedad=normal and viento=no and clase=P 4/12

Si se piensa en 100% de éxito, entonces sólo la primera regla cumple. De hecho existen 58 reglas considerando la tabla completa que cubren al menos dos ejemplos con un 100% de exactitud (*accuracy*). El proceso es el siguiente:

1. Genera todas los *items sets* con un elemento. Usa estos para generar los de dos elementos, y así sucesivamente.

   Se toman todos los posibles pares que cumplen con las medidas mínimas de soporte. Esto permite ir eliminando posibles combinaciones ya que no todas se tienen que considerar.

2. Genera las reglas revisando que cumplan con el criterio mínimo de confianza.

Una observación interesante, es que si una conjunción de consecuentes de una regla cumple con los niveles mínimos de soporte y confianza, sus subconjuntos (consecuentes) también los cumplen. Por el contrario, si algún ítem no los cumple, no tiene caso considerar sus súper conjuntos. Esto da una forma de ir construyendo reglas, con un solo consecuente, y a partir de ellas construir de dos consecuentes y así sucesivamente. Este método hace una pasada por la base de datos cada para cada conjunto de ítems de diferente tamaño.

El esfuerzo computacional depende principalmente de la cobertura mínima requerida, y se lleva prácticamente todo en el primer paso. El proceso de iteración del primer paso se llama *level-wise* y va considerando los súper conjuntos nivel por nivel. Lo que se tiene es una propiedad anti-monótona: si un conjunto no pasa una prueba, ninguno de sus súper conjuntos la pasan. Si un conjunto de ítems no pasa la prueba de soporte, ninguno de sus súper conjuntos la pasan. Esto se aprovecha en la construcción de candidatos para no considerar todos. Por ejemplo, se tiene la siguiente tabla con listas de compras de productos.

id1	P1,p2,p5
id2	P2,p4
id3	P2,p3
id4	p1,p2,p4
id5	p1,p3
id6	p2,p3
id7	p1,p3
id8	p1,p2,p3,p5
id9	p1,p2,p3

Tabla 3 - Productos por compra

La figura muestra este proceso con los datos de la tabla anterior.

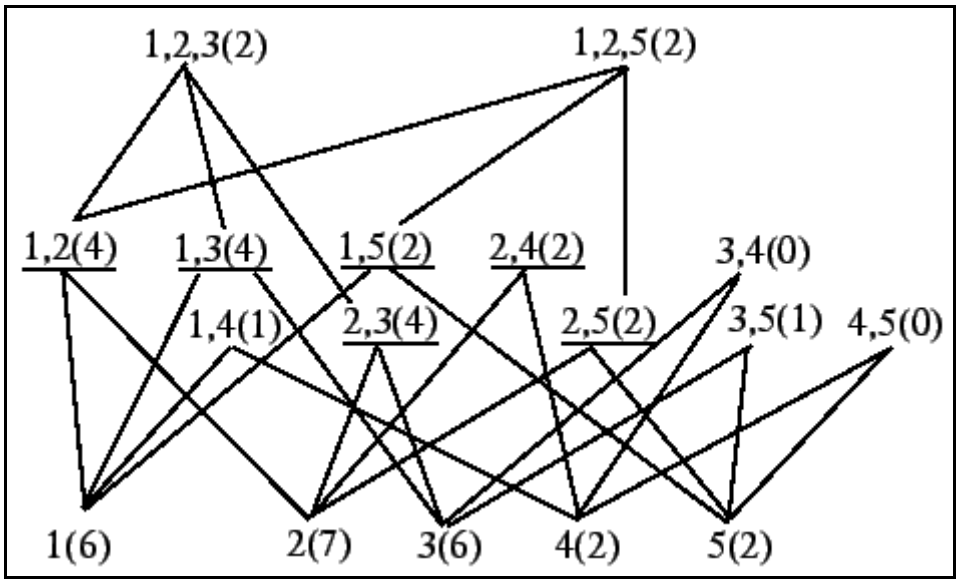


Ilustración 4 – Generación de item-sets

Se muestra la generación de candidatos por niveles. El primer número indica el producto y el número entre paréntesis las veces que ocurre.

Una vez que se tienen los conjuntos de ítems, generar las reglas es relativamente sencillo.

- Para cada conjunto l de ítems, genera todos sus subconjuntos.

- Para cada subconjunto $s \subset l$, genera una regla:

$s \Rightarrow (l - s)$ si:

$$\frac{\text{soporte}(l)}{\text{soporte}(s)} \geq \text{nivel\_confianza}$$

Todas las reglas satisfacen los niveles mínimos de soporte.

Se han hecho algunas mejoras al algoritmo básico de reglas de asociación (apriori) para hacerlo más eficiente:

- Usar tablas *hash* para reducir el tamaño de los candidatos de los *itemsets*
- Eliminar transacciones (elementos en la base de datos) que no contribuyan en súper conjuntos a considerar.
- Dividir las transacciones en particiones disjuntas, evaluar itemsets locales y luego, en base a sus resultados, estimar los globales.
- Hacer aproximaciones con muestreos en la lista de productos, para no tener que leer todos los datos.

Dentro de las extensiones principales, se pueden citar:

1. Encontrar reglas de asociación a diferentes niveles de abstracción.

   Normalmente se empieza con las clases superiores, y los resultados pueden servir para filtrar clases inferiores.

   Por ejemplo, considerar reglas de asociación sobre computadoras e impresoras, y luego sobre laptops y estaciones de trabajo, por un lado, y sobre impresoras láser y de punto por otro, etc.

   Al proceder a las subclases se puede considerar:

- Un criterio de soporte uniforme.
- Reduciendo el criterio para las subclases.
- Considerar todas las subclases independientemente del criterio de soporte.
- Tomando en cuenta el criterio de soporte de una de las superclases de un ítem o $\boldsymbol{k}$ superclases de $\boldsymbol{k}$ ítems.
- Considerar ítems aunque el nivel de soporte de sus padres no cumplan con el criterio de soporte, pero que sea mayor que un cierto umbral.

   Al encontrar reglas de asociación a diferentes niveles de abstracción es común generar reglas redundantes o reglas que no dicen nada nuevo (la regla más general, ya decía lo mismo), por lo que es necesario incorporar mecanismos de filtrado.

2. Encontrar reglas de asociación combinando información de múltiples tablas o reglas de asociación multidimensionales.

3. Las reglas de asociación, al igual que los árboles de decisión y las reglas de clasificación que se han visto, funcionan, en su forma original, con atributos discretos.

   Al igual que en las otras técnicas se han propuesto mecanismos para manejar atributos continuos.

   Los enfoques más comunes son:

- Discretizar antes de minar en rangos usando posiblemente jerarquías predefinidas.
- Discretizar dinámicamente durante el proceso tratando de maximizar algún criterio de confianza o reducción de longitud de reglas.
- Discretizar utilizando información semántica, formar grupos con elementos cercanos (posiblemente haciendo *clustering* sobre los atributos). Una vez establecidos los clusters, encontrar las reglas de asociación con esos *clusters* basados en similitudes.

1.8. ÁRBOLES DE DECISIÓN

El proceso de creación de un árbol de decisión se debe ver como un proceso recursivo: primero se selecciona un atributo para ubicar en el nodo raíz y hacer una rama para cada posible valor del atributo. Esto divide el conjunto en dos subconjuntos y el proceso se repite recursivamente para cada una de las ramas.

La pregunta es: ¿cómo decidir qué atributos poner en qué nodo del árbol?. Para poder responderla, se debe conocer primero el concepto de pureza de un nodo. Si un atributo dado clasifica los datos de prueba de forma que en cada rama resultante queden la mayor cantidad de valores iguales para el atributo clase (es decir el atributo que se desea predecir por ejemplo), entonces se dice que ese atributo es puro.

Para visualizar mejor los conceptos, de ahora en adelante, los ejemplos que se utilicen se referirán a la siguiente tabla con datos de futbolistas que registran información de su estado y si jugó o no.

Estado de Animo	Salud	Rendimiento en entrenamientos	Juega
triste	buena	positivo	si
triste	mala	positivo	no
triste	mala	mediocre	no
triste	mala	positivo	no
serio	mala	mediocre	no
serio	buena	positivo	si
serio	mala	positivo	si
serio	buena	positivo	si
alegre	buena	mediocre	si
alegre	buena	positivo	si
alegre	buena	positivo	si
alegre	mala	negativo	no

Ilustración 5 – Datos ejemplo – árboles de decisión.

A partir de estos datos, se pueden generar las siguientes clasificaciones:

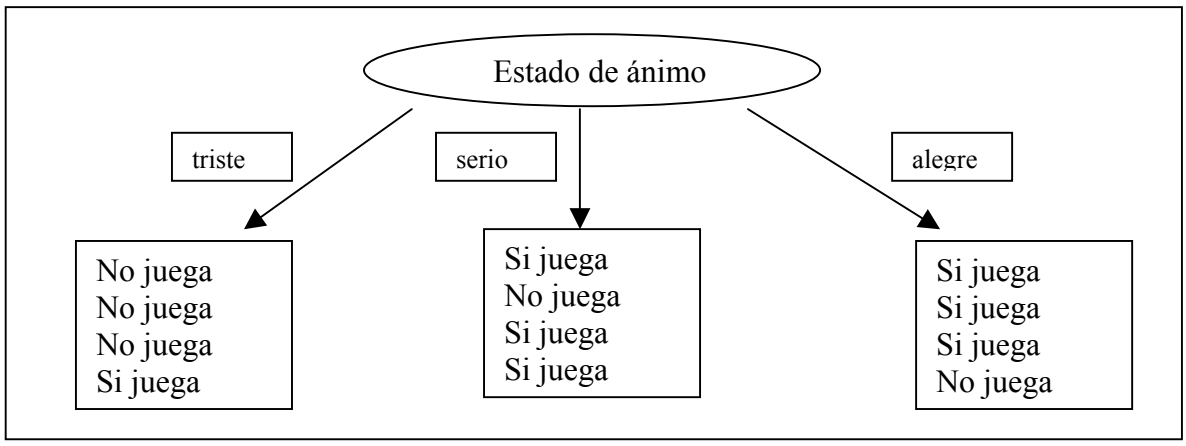


Ilustración 6 – Clasificación datos según atributo "Estado de ánimo"

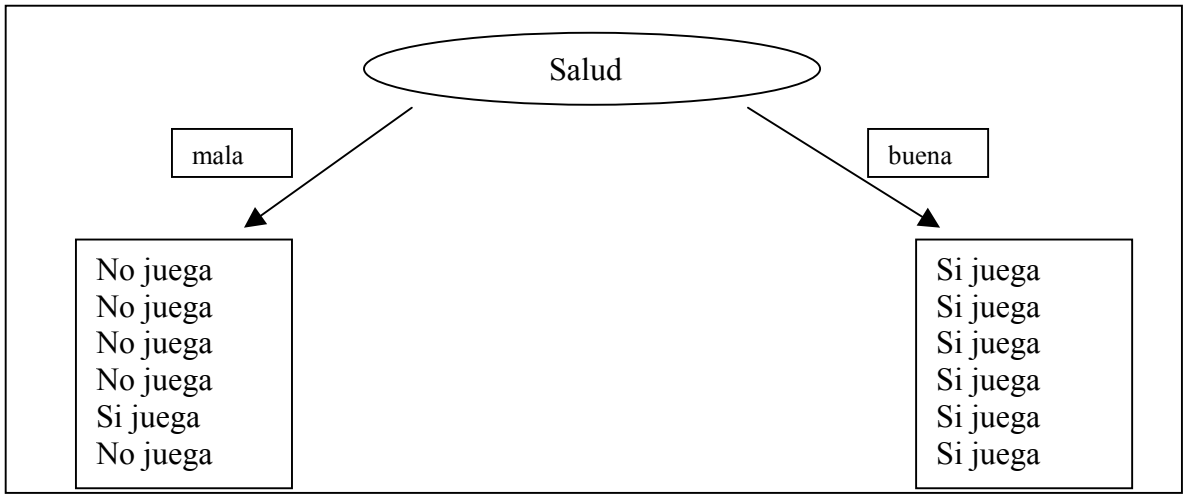


Ilustración 7 – Clasificación datos según atributo "Salud"

Se puede ver en las figuras 6 y 7., que se diferencian en cuanto a la clasificación que hacen de los datos originales, respecto al atributo clase, aunque los dos atributos que se toman para hacer la comparación (estado de ánimo y salud) tienen distinto número de valores posibles (3 y 2).

Se puede ver a simple vista que la clasificación hecha a partir del atributo "salud" de un jugador, separa mejor las instancias de acuerdo al valor que tienen en el atributo clase por lo tanto es un nodo más puro. La medida usada para la pureza es llamada información y es medida en unidades llamadas bits. Esta medida representa la cantidad de información que se necesitaría para determinar la clasificación de una instancia, dado que se llegó a ese nodo.

Para decidir qué atributo tomar como el que irá en la raíz del árbol (o en el nodo en cuestión cuando se trate de una fase recursiva del algoritmo), se debe calcular cuál de estas opciones representa mayor ganancia en términos de información como fue descrita anteriormente.

Esta ganancia es la diferencia entre la información que se tenía antes de esta iteración y la información que se obtendría si se seleccionara alguna de las opciones en cuestión. Para entender este concepto, es mejor verlo con el ejemplo.

Antes de aplicar cualquiera de los dos árboles en las figuras, se tiene la siguiente información:

Info([7,5])

pues se tienen 7 instancias con el valor sí en el atributo clase, y 5 instancias con el valor no, para el mismo atributo.

Para cada uno de los árboles de tiene la siguiente medida de información:

Estado de ánimo	Info([1,3],[3,1],[3,1])
Salud	Info([1,5],[6,0])

Para saber con cuál clasificación quedarse, se debe hacer la resta de las dos cantidades para hallar la ganancia que se obtiene con cada opción y comparar los resultados, así:

ganancia(Estado de ánimo) = Info([7,5]) - Info([1,3],[3,1],[3,1])

ganancia(Salud) = Info([7,5]) - Info([1,5],[6,0])

El cálculo que de mayor ganancia, se tomará como la opción de clasificación y se seguirá el proceso recursivamente con los demás atributos (excluyendo el que ya se tomo como atributo clasificador).

Se deben tomar precauciones con atributos identificadores como un ID, pues debido a que identifican únicamente a cada instancia de los datos, pues quedaría un árbol con la siguiente forma:

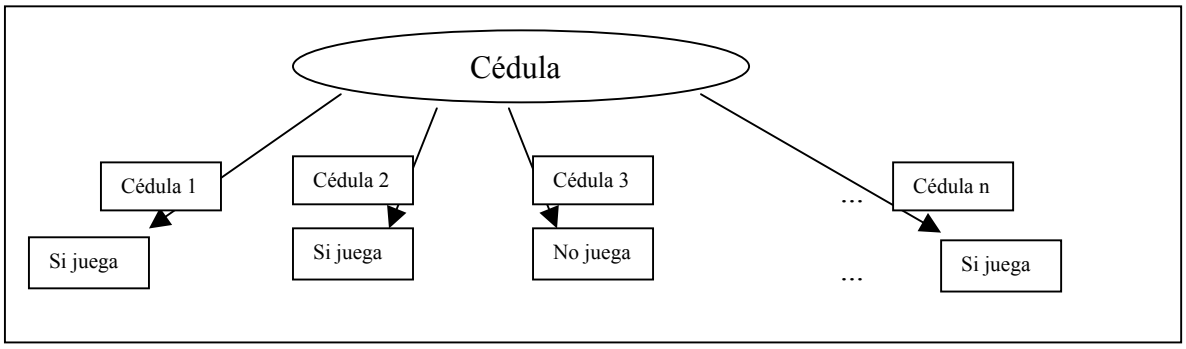


Ilustración 8 – Clasificación datos según atributo "Cédula"

Por este efecto, el algoritmo creería que se trata de un atributo que separa las instancias de manera perfecta, pues en cada nodo resultante siempre hay una sola instancia (y consiguientemente una sola clasificación para el atributo clase). Esta situación es compensada por el algoritmo mediante una medida denominada gain ratio (que puede traducir "ganancia"), que toma en cuenta el número y tamaño de los nodos en los que un atributo divide el dataset, sin tener en cuenta la información del atributo clase.

La herramienta de Minería de Datos WEKA utilizada como herramienta base de este proyecto de investigación[1], implementa el algoritmo J4.8 el cual es una implementación del ya mencionado C4.5 (también conocida como *C4.5 Revision 8*) la cual fue la última versión pública de esta familia de algoritmos antes de la salida de una implementación comercial llamada C5.0.

1.9. CRITERIOS DE COMPARACIÓN DE ALGORITMOS

Para poder realizar una comparación adecuada de los algoritmos, primero se deben tener claramente establecidos los criterios por los cuales se decidirá que

[1] Para conocer más acerca de la herramienta de Minería de Datos WEKA, dirigirse al anexo 5.1. - Herramienta WEKA

un algoritmo dado es más o menos apto para un dominio de aplicación específico. Dentro de estos criterios, se han establecido los siguientes:

1. Velocidad de Ejecución:

Claramente se debe tener en cuenta el tiempo que se demora cada uno de los algoritmos en obtener el modelo.

- Costo computacional involucrado en generar el modelo.

- Mostrará con qué rapidez el algoritmo es capaz de extraer un modelo exitosamente a partir de los datos de entrenamiento suministrados.

2. Precisión para clasificación de datos de origen

Esta medida mostrará en qué porcentaje, los datos de entrenamiento suministrados al algoritmo, son clasificados correctamente por la o las reglas creadas. Se presentará la exactitud con que el modelo logró clasificar los registros suministrados.

3. Precisión para predicción de datos futuros

Después de entrenarse con los datos iniciales y generar el modelo, deberán hacerse pruebas que permitan determinar con qué precisión ese conjunto de reglas generado clasifica registros para los cuales no se suministra el campo "clase", es decir, aquel cuyo valor se desea predecir.

Se mostrará la exactitud que tiene el modelo generado para predecir datos futuros.

4. Escalabilidad

Habilidad de construir un modelo eficiente dadas grandes cantidades de datos. El término "grandes cantidades de datos" se refiere al dominio sobre el que se trabaja. En el caso de los datos de esta investigación se tendría que comparar con el número total de estudiantes (ingeniería).

5. Robustez

Habilidad del modelo para hacer predicciones correctas en base a datos con ruido o datos con valores faltantes.

La manera como cada uno de estos factores será tenido en cuenta para la evaluación de cada uno de los algoritmos en términos del dominio de aplicación, será explicada en el capitulo de metodología de esta investigación.

1.10. PSEUDOCÓDIGOS

1.10.1. Pseudocódigo de árboles de asociación

A continuación se describe el pseudocódigo del algoritmo de árboles de decisión J.48. En la primera parte se hace una descripción de cada uno de los algoritmos que componen el algoritmo general, así como de los parámetros de entrada que requieren y sus salidas. En la segunda parte se muestra el algoritmo en forma de pseudocódigo.

DEFINICIÓN DE LOS ALGORITMOS:

hallarArbol(A,R,C): devuelve el subárbol que mejor divide los datos en el conjunto de registros R, con el conjunto de atributos A para clasificar los registros de acuerdo al atributo clase C.

hallarAtributo(A,R,Res): devuelve el atributo del conjunto de atributos A que mejor divide los registros del conjunto R, en el conjunto de conjuntos de registros Res.

hallarInfo(Atributo, R): encuentra la ganancia de información que se obtendría de dividir los registros R por el atributo "Atributo".

agregarInfos(VectorInfos, info, i): agrega el valor "info" al vector VectorInfos en la posición i.

agregarHijo(subárbol1, arbolFinal): agrega una rama que contiene el subárbol "Subárbol1" al árbol que contiene el resultado: arbolFinal.

Valores(A): devuelve un vector con los posibles valores de un atributo determinado.

Tamaño(A): devuelve el número de atributos que contiene el conjunto de atributos A.

Longitud(V): devuelve la longitud (número de elementos) del vector V.

PSEUDOCODIGO:

```
hallarArbol(A, R, C)
|       atributo ← hallarAtributo(A,R,Res, Valores)                    a)
|       numValores ← Longitud( Valores(atributo) )
|       para i ← 0 hasta  numValores hacer
        b)
|       |       conjuntoValoresClase ← Res[i]
|       |       agregarHijo(hallarArbol(A-atributo,  R,  conjuntoValoresClase),
arbolFinal)
|       fin_para
|       devolver arbolFinal
Fin
```

a) En atributo queda el que mejor divide a los datos y debe estar en la raíz
   // Res se envía vacío
   // Valores se envía vacío

b) Para cada valor del atributo seleccionado, se crea una rama del árbol, con un subárbol creado por el mismo algoritmo

```
hallarAtributo(A, R, Res, Valores)
|       tam ← Tamaño(A)
|       para i ← 0 hasta  tam  hacer                                   a)
|       |       atributo ← A[i]
|       |       info ← hallarInfo(Atributo, R, Valores)
|       fin_para
|       posMax ← Maximo (VectorInfos)                                  b)
|       devolver  A[posMax]
fin
```

a) Para cada uno de los atributos en la lista, hallar su valor de "info"
b) Encontrar cual de todos los atributos fue el que dio mayor "info"

```
hallarInfo (Atributo, R, Clase, Valores)
|       vectorValores ← [vacío]
|
```

```
|       numValores ← Longitud( Valores (Atributo) )                a)
|       para i ← 0 hasta numValores hacer
|       |       crearVector(vectorValores, i)
|       fin_para
|
|       numRegistros ← Tamaño(R)
|
|       para j ←0 hasta numRegistros hacer                                b)
|       |       registro ← R[j]
|       |       valor ← valorAtributo(Atributo, registro)
|       |       clase ← valorClase(Clase, registro)
|       |       agregarElemento(valor, clase)
|       fin_para
|
|       tamVector ← Tamaño(vectorValores)
|
|       para k ← 0 hasta tamVector hacer                    c)
|       |       calcularInfo(k)                             // Info
|       fin_para
|
|       calcularInfoTotal()                          // Entropía
|       devolver Info
fin
```

a) Este ciclo créale vector "horizontal" (vectorValores) donde se almacenará los valores de "info". Tiene una posición por cada posible valor del atributo
b) Este ciclo recorre todos los registros en R, y suma 1 a la cuenta correspondiente.
c) Este ciclo recorre todo el vector para sacar la cuenta de "info" para cada valor del atributo y después sumar todos los valores (Entropía)

Para el cálculo de la información suministrada al dividir los datos por un atributo determinado, se utiliza la función hallarInfo, que a su vez utiliza la siguiente estructura de datos para almacenar los números de info necesarios para realizar este cálculo:

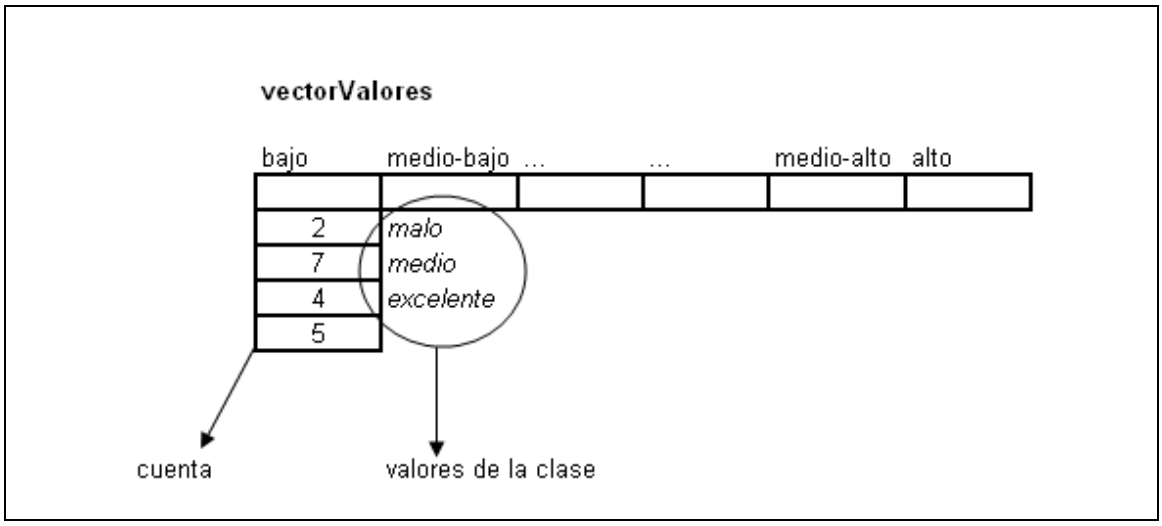


Ilustración 9 - Proceso árboles de asociación

1.10.2. Pseudocódigo de reglas de asociación

A continuación se muestra el pseudocódigo del algoritmo de reglas de asociación "a-priori".

DEFINICIÓN DE LOS ALGORITMOS

Atributos: arreglo que contiene la lista de todos los atributos en el conjunto de datos.

ValoresAtributo: arreglo que contiene la lista de todos los posibles valores para un atributo i.

TamañoItemSet(IS): devuelve el número de item sets que contiene el conjunto de item sets IS

Tamaño(A): devuelve el número de atributos que contiene el conjunto de atributos A

Valores(A): devuelve el número de posibles valores de un atributo determinado

nuevoItemSet(A,V): función que devuelve un nuevo item set, con la pareja atributo(A)–valor(V) suministrada. También puede recibir como parámetro un conjunto de parejas de atributos y valores.

hallarCubrimiento(IS): devuelve un valor con el cubrimiento del itemSet IS sobre el conjunto de registros R.

item-sets-uno(A, R, cub): devuelve el conjunto de item-sets de tamaño 1, que tienen un cubrimiento mayor a "cub" en el conjunto de registros R, con atributos A.

a-priori (A, R, cub, pres): algoritmo a-priori, que devuelve las reglas que se generan a partir de los registros en R, con los atributos A, de acuerdo a un cubrimiento "cubrimiento" y una precisión "precisión" mínimos preestablecidos por el usuario.

PSEUDOCODIGO:

```
a-priori (A, R, cubrimiento, precisión)
|       is1 ← item-sets-uno (A, R, cubrimiento)
|       isTotal ← generar-item-sets (is1, is1, A, R, cubrimiento
        a)
|       devolver reglas(isTotal, precisión)
Fin
```

a) se llama a "generar-item-sets" con el conjunto de itemSets de 1 elemento en los dos argumentos, para que empiece generando el conjunto de itemSets de 2 elementos y continúe en adelante.

Encontrar los 1-Item-Sets:

```
item-sets-uno(A, R, cub)
|       numAtributos ← Tamaño(A)
|       para i ← 0 hasta numAtributos                                   a)
|       |       atributo ← Atributos[i]
|       |       numValores ← Valores(atributo)
|       |       para j ← 0 hasta numValores                             b)
|       |       |       v ← ValoresAtributo[i]
|       |       |       itemSet1 ← nuevoItemSet (a,v)
|       |       |       cubrimiento ← hallarCubrimiento (itemSet1, R)
|       |       |       si cubrimiento > cub  →
|       |       |       |       agregarItemSet (itemSet1, itemSets)
|       |       |       fin_si
|       |       fin_para
|       fin_para
|       devolver itemSets
Fin
```

a) Equivalente a: "Para cada atributo a del conjunto de datos, hacer".
b) Equivalente a: "Para cada valor del atributo a: v, hacer".

Encontrar los n-Item-Sets:

```
generar-item-sets (itemSetsNuevos, itemSetsUno, A, R, cub)            a)
|       tamañoNuevos ← tamañoItemSet(itemSetsNuevos)
|       tamañoUno ← tamañoItemSet(itemSetsUno)
|       si (    (tamañoNuevos < tamaño(A))    y
                b)
```

```
|       |       (tamañoNuevos > 0)        )  →
|       |       para i ← 0 hasta tamañoNuevos
|       |       |       para j ← 0 hasta tamañoUno
|       |       |       |       i1 ← nuevoItemSet ( itemSetsUno(j),
itemSetsNuevos(i) )
|       |       |       |       cubrimiento ← hallarCubrimiento (i1, R)
|       |       |       |       si cubrimiento > cub  →
|       |       |       |       |       agregarItemSet (i1, itemSetsNuevosMasUno)
|       |       |       |       fin_si
|       |       |       fin_para
|       |       fin_para
|       |       itemSetFinal ← generar-item-sets (itemSetsUno,
|       |                                               itemSetsNuevosMasUno, A, R,
|       |                                               cub)
|       fin_si
|       devolver itemSetFinal
Fin
```

a) -itemSetsNuevos: conjunto de itemSets generado en la ejecución anterior. El que se cree en esta ejecución tener 1 itemSet más.
- itemSetsUno: conjunto de itemSets de tamaño 1, que sirven para ampliar el conjunto itemSetsNuevos en 1.

b) Para saber si el conjunto de itemSets no contiene aun todos los atributos y para saber si la anterior ejecución encontró algún itemSet válido.

---

reglas(itemSets, precisionMinima): obtiene reglas a partir de los item-sets hallados, que se encuentren en el conjunto itemSets y que cumplan con la precisión establecida por la variable precisionMinima (establecida por el usuario para aceptar una regla como válida).

numItemSets(itemSets): número total de item-sets en el conjunto de item-sets itemSets.

Buscar(Y,X): funcion que devuelve la posición de un item set dado X, dentro de un conjunto de item sets Y

obtenerCubrimiento(Y,X): devuelve el cubrimiento que otorga un data set que se encuentra en la posición X de un conjunto de data sets Y.

agregarRegla(Y,X): adiciona una regla Y, a un conjunto de reglas X.

```
reglas(itemSets, precisionMinima)
|       para i ← 0 hasta S, hacer
|       |       cubrimiento ← ObtenerCubrimiento (itemSets, i)
|       |
|       |       hacer todas las reglas posibles a partir del itemSet i
|       |
|       |       para cada regla R creada, hacer:
|       |       |       itemSetTemp ← nuevoItemSet (parejas a-v del antecedente
|       |       |                                       de la regla R)
|       |       |       posición ← Buscar (itemSets, itemSetTemp)
|       |       |       cubTemp ← ObtenerCubrimiento (itemSets, posición)
|       |       |       precision ← cubrimiento / cubTemp
|       |       |       si precision > PrecisionMinima →
|       |       |       |       agregarRegla (R, ReglasFinales)
|       |       |       fin_si
|       |       fin_para
|       fin_para
Fin
```

2. METODOLOGIA

En este capítulo se mostrarán las pruebas que fueron diseñadas, con base en los criterios de comparación de algoritmos mostrados en el capítulo 1, para efectuar dicha comparación en los algoritmos objeto del estudio. Primero se dará una breve descripción de cada una de las pruebas, para después describirlas en detalle. Posteriormente, se hará un recuento de los elementos que conforman la salida de la herramienta de Minería de Datos utilizada en la investigación (WEKA), tanto para árboles de decisión como para reglas de asociación.

También se hace una descripción de los datos de origen de la investigación, es decir, la base de datos sobre la cual se aplicaron los dos algoritmos de tal manera que se entienda que datos se tenían a disposición y que posibles salidas de podrían tener de la aplicación de los algoritmos.

Se muestra también en este capítulo, cual fue el proceso de limpieza y discretización de datos que se aplicó a los datos de origen, que como se describió en el capítulo 1, es parte integral del proceso de Minería de Datos.

Finalmente, se muestra cuál es el proceso de interacción con la herramienta WEKA, es decir, cuál es el proceso para alimentar la información a la misma y qué objetivos se perseguían con el diseño de las consultas[2] seleccionadas.

2.1. PRUEBAS PARA LA COMPARACIÓN DE ALGORITMOS

Según los criterios de comparación de algoritmos del capítulo 1, fue diseñada una prueba para cada uno de ellos, con el fin de poder evaluar con precisión que algoritmo es mejor en cada aspecto. Se diseñó una prueba para evaluar los algoritmos en términos de velocidad de ejecución, precisión para clasificación de datos de origen, precisión para predicción de datos futuros, escalabilidad y robustez. Las pruebas diseñadas son las siguientes:

[2] En lo que resta de este proyecto de investigación, se entenderá por *consultas*, los criterios de selección de registros de la base de datos que permiten la selección y extracción de información para la aplicación de los algoritmos de Minería de Datos.

2.1.1. Pruebas a ejecutar para cada criterio

- **VELOCIDAD DE EJECUCIÓN**

Para el caso del algoritmo de árboles de decisión, la misma herramienta WEKA brinda una medición del tiempo que necesitó para crear el modelo.

Para tener una velocidad precisa del tiempo de construcción se ejecutará WEKA desde línea de comandos con un parámetro que indicará el tiempo inicial y el tiempo final de la operación. Este método se aplicará para las dos técnicas.

- **PRECISIÓN PARA CLASIFICACIÓN DE DATOS DE ORIGEN**

Esta medida mostrará en que porcentaje, los datos de entrenamiento suministrados al algoritmo, son clasificados correctamente por la o las reglas creadas.

- **PRECISIÓN PARA PREDICCIÓN DE DATOS FUTUROS**

La idea en este caso es realizar pruebas conocidas como backtest, las cuales consisten en hacer algo parecido a lo que hace el algoritmo de WEKA al extraer modelos a partir de la información inicial; es decir, dividir los datos en dos porciones. En este caso puede tratarse de dos divisiones en base a tiempo, una parte de registros (porción 1) desde el año A hasta el año B y la segunda (porción 2) desde el año B hasta el año C. Posteriormente se realiza el entrenamiento de la herramienta a partir de los datos de la porción 1 y se genera un modelo determinado. Este modelo es puesto a prueba (de acuerdo al criterio expuesto en este punto), con los datos de la porción 2.

De esta manera se puede determinar con qué precisión el modelo generado inicialmente se puede aplicar, no solo a los datos que se tienen en la porción 2, sino a cualquier dato futuro que se pueda tener, lo cual favorece al dueño de la información, en este caso a la Universidad.

Se tomarán un tamaño determinado de registros de forma aleatoria para determinar el modelo correspondiente a esta porción. Luego se validara este modelo con los datos sobrantes para comprobar que tan preciso es el modelo generado midiendo el porcentaje de registros que son clasificados correctamente con el modelo.

- **ESCALABILIDAD**

Se generará un modelo con una porción de los datos con el fin de comparar otro modelo generado con una porción más grande de los datos. Se compararán los dos modelos en base a la precisión de predicción de cada uno de ellos.

- **ROBUSTEZ**

Se realizarán predicciones de registros de valores ruidosos o con valores faltantes y se comparará con las predicciones de registros correctos. Se podrán comparar los porcentajes de registros validos con los porcentajes de registros validos con los registros ruidosos.

2.1.2. Metodología de pruebas

Primero se hará un recuento de ciertos valores importantes para la evaluación de los resultados de los algoritmos. Posteriormente se mostrará cuál será la metodología a seguir en la aplicación de las pruebas.

Soporte:

El parámetro del algoritmo a priori, lowerBoundMinSupport es un decimal que representa el porcentaje mínimo de instancias que cubre una regla (para la que se cumple las condiciones) con respecto al total de instancias.

Ejemplo:
Número total de instancias: 24
lowerBoundMinSupport: 0.2 (las reglas deben cubrir el 20% de 24 = 4.8)

1 Si el soporte de las reglas es superior a 4, es tenida en cuenta. De lo contrario es eliminada.

Para la investigación, se definirá un soporte mínimo del 5%. Se escogió este valor debido al tamaño de población que se tiene, ya que para una regla que aplique a un 5%, se considera de importancia para los interesados en los resultados. Esto implica que la variable lowerBoundMinSupport tendrá un valor de 0.05.

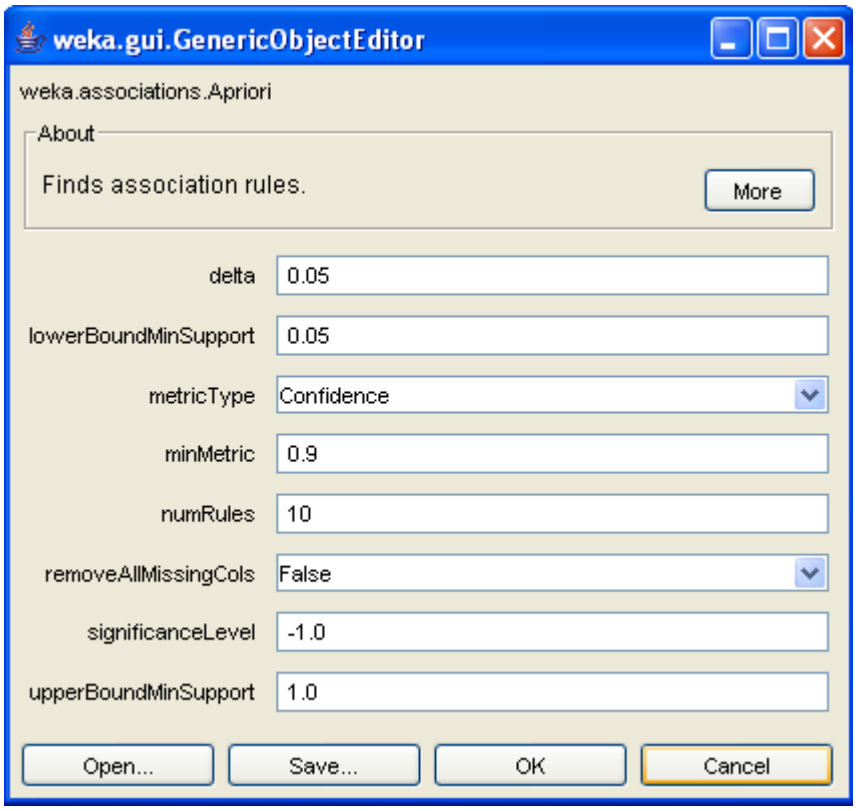


Ilustración 10 – Parámetros para el algoritmo a-priori de reglas de asociación

Este es el valor clave para realizar la poda del árbol resultante del algoritmo J4.8. Las reglas que se extraigan de cada una de las ramas del árbol resultante, serán evaluadas con este parámetro de soporte y eliminadas las que no lo cumplan.

Confianza:

El parámetro de confianza es el otro valor a tener en cuenta, ya que mostrará con qué grado de precisión una regla clasifica los registros para los cuales es aplicable.

En la herramienta WEKA, para el caso de las reglas de asociación, el valor de confianza no se puede establecer previamente, sino que es mostrado para cada regla como parte de los resultados de la ejecución. Dependiendo de este valor, se seleccionarán las reglas que cumplan con un cierto valor de confianza mínimo.

```
Best rules found:

 1. clasificacion=ALTO 892 ==> tperdidos=0 876    conf:(0.98)
 2. sexo=M clasificacion=ALTO 578 ==> tperdidos=0 567    conf:(0.98)
 3. clasificacion=ALTO estadomatricula=S 720 ==> tperdidos=0 705    conf:(0.98)
 4. sexo=M clasificacion=ALTO estadomatricula=S 451 ==> tperdidos=0 441    conf:(0.98)
 5. clasificacion=ALTO tcursados=16-20 417 ==> tperdidos=0 405    conf:(0.97)
 6. clasificacion=ALTO tcursados=16-20 estadomatricula=S 385 ==> tperdidos=0 373    conf:(0.97)
 7. clasificacion=ALTO tcursados=16-20 417 ==> estadomatricula=S 385    conf:(0.92)
 8. clasificacion=ALTO tperdidos=0 tcursados=16-20 405 ==> estadomatricula=S 373    conf:(0.92)
 9. tcursados=1-5 estadomatricula=S 435 ==> tperdidos=0 397    conf:(0.91)
```

Ilustración 11 – Confianza en reglas resultantes

Para el caso de árboles de asociación, se tiene como resultado de la ejecución, el número de instancias para las cuales aplica una regla (o rama del árbol) y el número de instancias mal clasificadas por esta regla. Con el dato del soporte (número de instancias bien clasificadas) y el número total de instancias a las que aplica la regla se puede conocer el valor de la confianza para una regla determinada.

Nuevamente se realizará el proceso de eliminación de las reglas que no cumplan con el valor mínimo preestablecido.

Para el caso de esta investigación se tomará el valor de la confianza de un 93%. El valor de la confianza dará la precisión con la que se quiere que estén los resultados. Este valor se estableció de acuerdo al número de reglas que resultaron después de aplicar los algoritmos de tal manera que se pudo determinar el valor mínimo de confianza con el objetivo de tener las reglas necesarias para el desarrollo del Proyecto de Investigación.

- **Velocidad de ejecución**

Se tomarán 20 datos de hora inicial de ejecución del algoritmo y hora de finalización del mismo, para después hacer la resta y saber el tiempo que tomó su ejecución. Se extraerá el promedio de estos datos para concluir cual es el tiempo promedio de ejecución con los datos seleccionados.

El archivo .bat que permite conocer el tiempo de ejecución será el siguiente para el caso de árboles de clasificación:

```
start time
java weka.classifiers.trees.J48 -t query2.arff
start time
```

Para el caso de reglas de asociación, el archivo .bat contendrá los siguientes comandos:

```
start time
start java weka.associations.Apriori -t query2.arff
start time
```

De estas pruebas resultarán los tiempos antes y despues de la aplicación de los algritmos, que servirá para conocer la diferencia de los tiempos y asi poder comprar tiempos entre tecnicas:

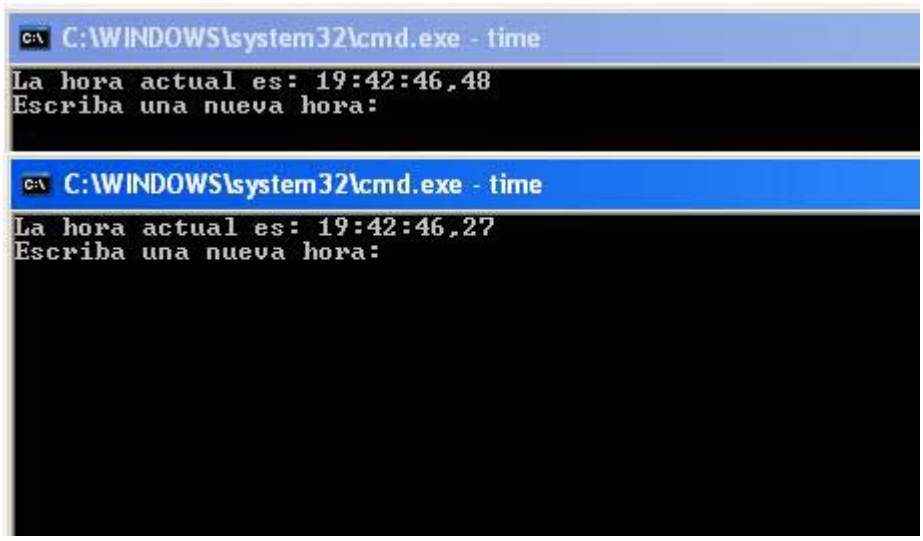


Ilustración 12 - Prueba de velocidad

Número identificador de prueba	Tiempo Inicial	Tiempo Final	Diferencia
A1			
A2			
A3			
A4			
A5			

Tabla 4 - Resultados prueba velocidad

	Número de Pruebas	Suma total de Diferencias	Promedio
Arboles de decisión	20		
Reglas de asociación	20		

Tabla 5 - Comparación resultados

- **Precisión para clasificación de datos de origen**

Al realizar el procedimiento de árboles de decisión y de reglas la herramienta muestra en sus resultados el número de instancias clasificadas correctamente y el número de instancias clasificadas incorrectamente. Con estos datos se podría hacer una comparación con estos datos del porcentaje de cada una de las técnicas y compararlos.

- **Precisión para predicción de datos futuros**

En esta prueba se partirán los datos en el tiempo. Se tomará como base para la generación del modelo un conjunto de datos perteneciente a años anteriores a partir de un año base seleccionado y se dejarán registros pertenecientes a años posteriores al año base seleccionado.

El siguiente paso es generar el modelo con los datos del año base hacia atrás.

Finalmente se tomarán los datos del año base en adelante y se empleará el modelo para conocer cual es el numero de datos clasificados correctamente, con esto se podrá saber con datos reales a futuro cual de las dos técnicas es la más conveniente.

Ejemplo:

"SELECT PROGRAMAS.NOM_PROGRAMA,
NOTASGENERAL.TCURSADOS, NOTASGENERAL.TPERDIDOS,
NOTASGENERAL.CLASIFICACIÓN, ESTUDIANTE.SEXO
FROM ESTUDIANTE, NOTASGENERAL, PROGRAMAS, ESTADO
WHERE NOTASGENERAL.ID=ESTUDIANTE.ID And
ESTADO.SEQ=ESTUDIANTE.NEXTVAL AND
ESTADO.PRG=ID_PROGRAMA AND PER_CIVIL<'231';"

En este caso se añadió una nueva condición con el fin de que se modele desde el primer semestre del 2003 hacia atrás. El objetivo es generar un modelo con datos anteriores en el tiempo para poder así comparar con los datos del 2003 en adelante la precisión en la predicción.

El objetivo es observar cuál es el número de estudiantes que estuvo bien clasificado por el modelo para los años 2003 y 2004. Así se conocerá que tan preciso es el modelo para poder predecir la clasificación de un estudiante con los parámetros de entrada.

Para conocer el número de intancias bien clasficadas por cada una de las técnicas de estudio se desarrollará una aplicación que recorrerá los registros a predecir y mostrará el número de registros que clasifica correctamente según el modelo generado.

Este proceso es ilustrado en la siguiente figura:

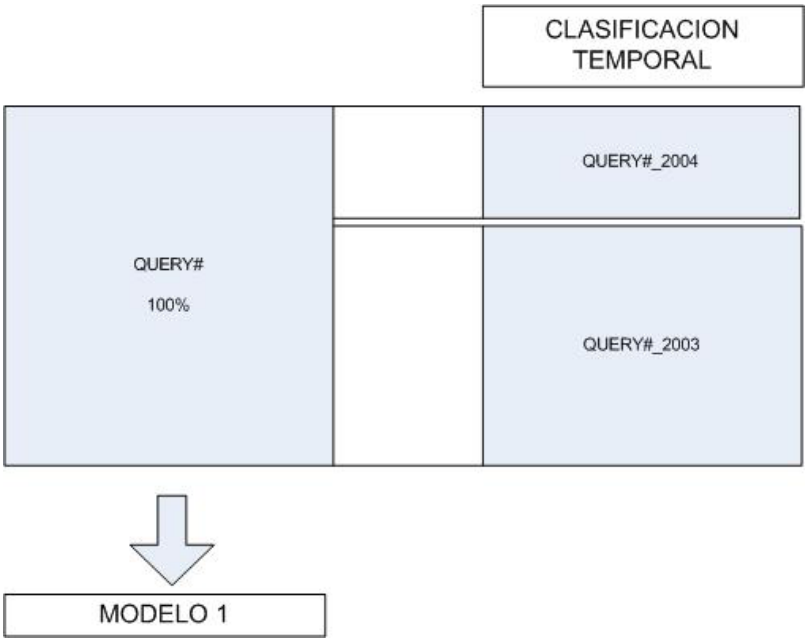


Ilustración 13 - Precisión en la clasificación de datos futuros

4. Escalabilidad.

Para poder efectuar este tipo de pruebas, se debe seleccionar un subconjunto de datos a partir del conjunto original de datos, de tal forma que se pueda comparar si existe o no relación entre los modelos generados por ambos conjuntos de datos de origen.

Para ello, se escogerá un subconjunto de un tamaño inferior al original. Esta muestra se eligirá de manera aleatoria para que la prueba tenga validez. Se generará el modelo para el 100% de los datos de origen (ModeloA), después se repetirá el procedimiento para una porción del 10% de los datos (ModeloB).

A partir de estos dos modelos se realizará la prueba de precisión en la predicción que dará como resultado el número de instancias correctamente clasificadas según cada modelo.

ModeloA = Ra (número de instancias correctamente clasificadas)
ModeloB = Rb (número de instancias correctamente clasificadas)

La diferencia entre los valores de cada uno de los modelos será lo que definirá la mayor o menor escalabilidad de cada técnica.

Rtécnica = |Rb – Ra|

Rtécnica se refiere a las técnicas seleccionadas para aplicar la prueba, en el caso de este proyecto se aplicará para árboles de decisión y para reglas de asociación.

Nuevamente se puede ver ilustrado el proceso en la siguiente figura:

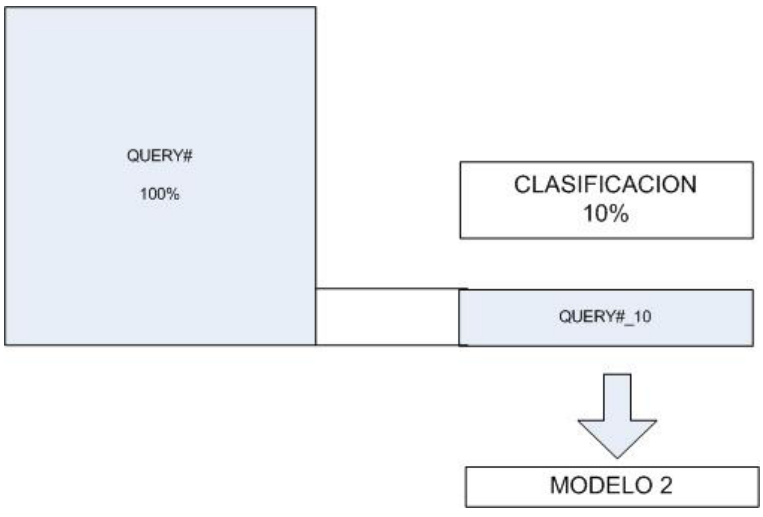


Ilustración 14 – Escalabilidad

5. Robustez.

Este tipo de pruebas buscan evaluar de qué manera el mecanismo para la generación de los modelos, soporta la existencia de datos ruidosos.

Para la prueba de robustez se generarán dos modelos por cada una de las técnicas de estudio. El primero se basará en los datos de origen sin modificación alguna y el segundo se presentará a partir de los mismos datos con modificaciones de registros, afectando sus campos de tal manera que contengan el valor nulo o vacío.

Esta prueba deberá ser realizada tanto para la técnica de árboles de decisión como para la de reglas de asociación.

Cada una de las pruebas dará como resultado un valor que indicará la diferencia de registros correctamente clasificados según los modelos resultantes. Con estos valores se podrán comparar las dos técnicas de estudio seleccionadas para el proyecto aplicando la prueba de predicción de datos futuros.

Este proceso es ilustrado en la siguiente figura:

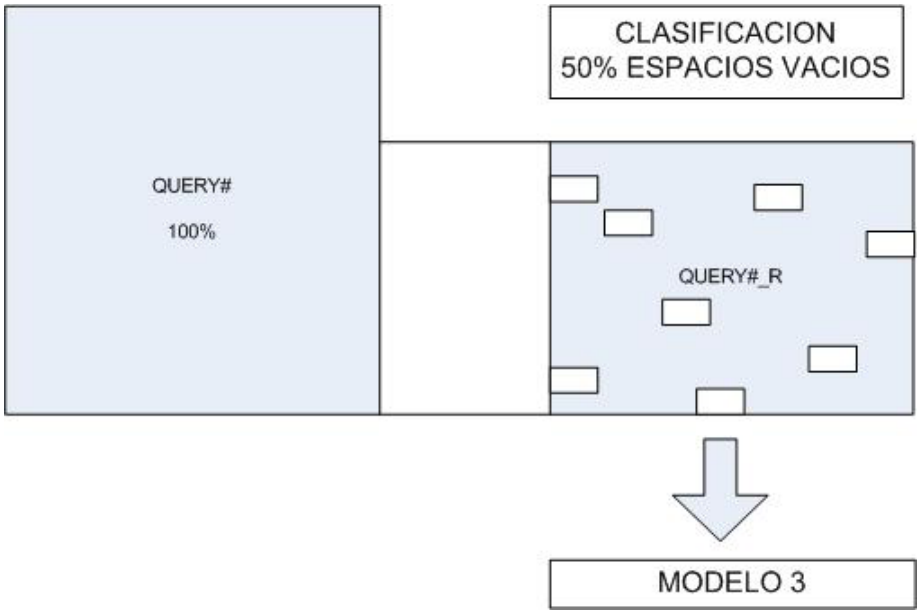


Ilustración 15 - Robustez

Para el caso de árboles de decisión se debe realizar el proceso adicional del paso de árboles a reglas. Para esto se diseñó en Excel una tabla donde se calculan y se muestran las reglas que corresponden a las ramas del árbol,

con sus respectivos valores de soporte y confianza. Con esto se puede seleccionar las que superen el mínimo establecido para estos parámetros.

Un ejemplo de esta tabla se puede observar en la siguiente figura:

Total	*# Clasificadas*	*# Mal Clasificadas*	*Soporte*	*Confianza*	*Entra*
20847	178	93	0,407732527	0,47752809	0
20847	83	48	0,167889864	0,421686747	0
20847	3	0	0,01439056	1	0

Tabla 6 - Ejemplo resultado

2.2. USO DE LA HERRAMIENTA WEKA

En este apartado se muestra, de qué manera fue utilizada la herramienta WEKA y los distintos elementos de salida que brinda después de la ejecución de un algoritmo de Minería de Datos, para la realización de las pruebas de comparación anteriormente descritas.

2.2.1. Elementos de resultados de WEKA para árboles de clasificación.

La ejecución del algoritmo de árboles de clasificación en la herramienta WEKA produce, además del árbol mismo, una serie de resultados numéricos que se muestran a continuación. Se utilizará el árbol (generado por WEKA) de la ilustración 14, para ilustrar algunos conceptos.

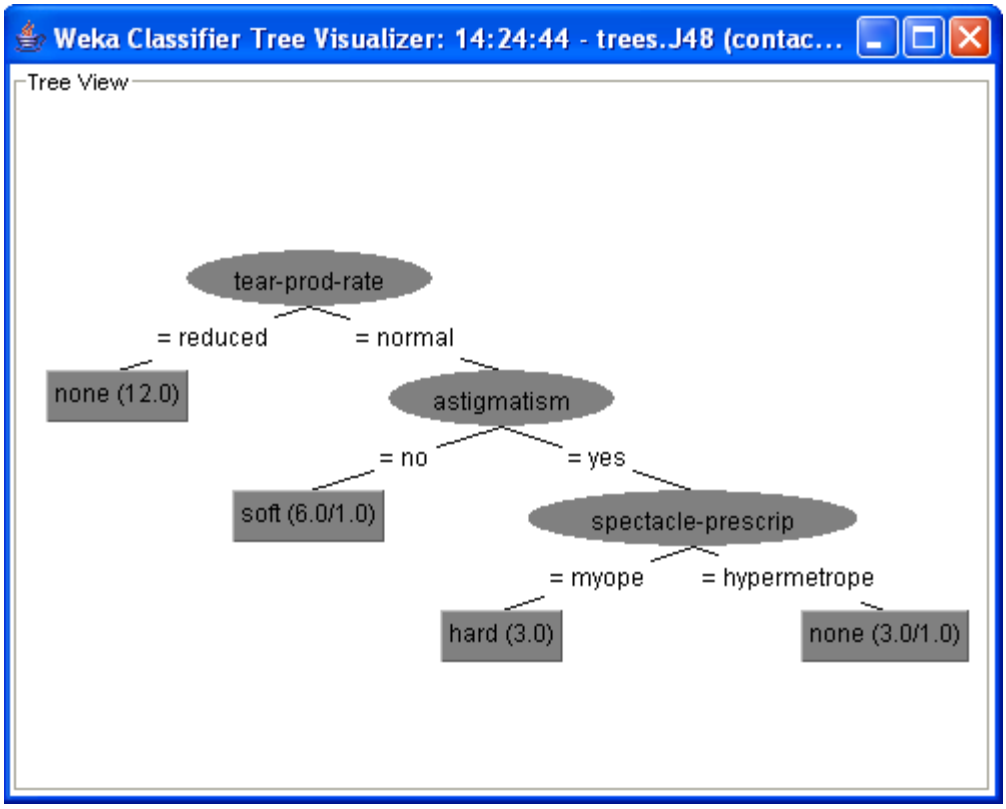


Ilustración 16 – Árbol resultante de WEKA

```
Number of Leaves  :    4
```
Numero de hojas del árbol resultante de la aplicación del algoritmo. También indica el número de posibles clasificaciones para un nuevo registro.

```
Size of the tree :    7
```
Número total de elementos que componen el árbol, entre hojas y nodos intermedios

```
Time taken to build model: 0.03 seconds
```
Tiempo tomado por el algoritmo en su ejecución. Este tiempo será de utilidad en las pruebas de velocidad de ejecución.

```
Correctly Classified Instances          20    83.3333%
```
Número de instancias correctamente clasificadas. El primer número corresponde al número de registros bien clasificados. El segundo es un porcentaje relativo al total de registros del conjunto de datos.

```
Incorrectly Classified Instances        4     16.6667%
```
Número de instancias mal clasificadas. Nuevamente aparece el valor absoluto y como porcentaje del total de registros.

La herramienta brinda las siguientes medidas del error cometido por el algoritmo de clasificación. Para todas ellas se cumple que p es el valor predicho para una instancia y a es su valor real. El número n corresponde al número total de instancias.

```
Mean absolute error                     0.15
```
Mide la magnitud individual de cada uno de los errores cometidos en la predicción. Su fórmula es:

$$= (|p_1 - a_1| + \ldots + |p_n - a_n|) / n$$

```
Root mean squared error                 0.3249
```
Es la medida más comúnmente utilizada, pues se comporta bien matemáticamente. Su fórmula es:

$$= \sqrt{((p_1 - a_1)^2 + \ldots + (p_n - a_n)^2 / n)}$$

Tiende a exagerar el efecto que pueden tener instancias cuyo error de predicción sea mucho mayor al promedio de los errores.

```
Relative absolute error                  39.7059 %
```
Su fórmula es:

$$= (|p_1 - a_1| + \ldots + |p_n - a_n|) / (|a_1 - \bar{a}| + \ldots + |a_n - \bar{a}|)$$

Corresponde al error absoluto total, normalizado dividiéndolo por las diferencias de los valores reales.

```
Root relative squared error              74.3898 %
```
Su fórmula es:

$$= \sqrt{(p_1 - a_1)^2 + \ldots + (p_n - a_n)^2 / (a_1 - \bar{a})^2 + \ldots + (a_n - \bar{a})^2}$$

donde $\bar{a}$ corresponde a la media de los valores reales.

```
Total Number of Instances            39185
```
Número total de instancias del conjunto de datos de origen.

```
Ignored Class Unknown Instances                 4079
```
Número de instancias que no contienen valor para el atributo clase y que por tanto son **ignoradas** para realizar el modelo.

```
=== Detailed Accuracy By Class ===
TP Rate   FP Rate   Precision   Recall  F-Measure
Class
  1          0.053      0.833     1          0.909    soft
  0.75       0.1        0.6       0.75       0.667    hard
  0.8        0.111      0.923     0.8        0.857    none
```

TP Rate: significa "True-Positive Rate" e indica el porcentaje de instancias de una cierta clase, clasificadas correctamente dentro de esa clase.

FP Rate: significa "True-Positive Rate" e indica el porcentaje de instancias que, siendo de una cierta clase, son clasificadas incorrectamente dentro de esa clase. Por ejemplo, el primer caso (con valor 0.053) es el resultado de tomar el número de instancias clasificadas como "soft" cuando no lo eran (1 instancia) y dividirlo por el total de instancias con valores distintos a "soft" (19 instancias). Entonces, 1/19 = 0.05263 ≈ 0.053.

Matriz de confusión:

```
=== Confusion Matrix ===

  a  b  c   <-- classified as
  5  0  0 |  a = soft
  0  3  1 |  b = hard
  1  2 12 |  c = none
```

Un ejemplo más complejo de una matriz de confusión, para uno de los árboles utilizados en esta investigación es el siguiente:

```
=== Confusion Matrix ===

     a     b     c     d     e     f     g   <-- classified as
 13755  1539   935    11     6     0     0 |     a = 3.5 - 4.0
  4368  4069   304   264     4     0     3 |     b = 3.0 - 3.5
  7556    98  1854     0     9     0     0 |     c = 4.0 - 4.5
   384   778    22   780     2     2    22 |     d = 2.5 - 3.0
  1133    38   627     0     7     0     0 |     e = 4.5 - 5.0
     1    56     0   278     0     0    29 |     f = 2.0 - 2.5
     0    12     0   171     0     3    65 |     g = 0.0 - 2.0
```

Para todos los registros con un cierto valor de atributo clase, indica cuántos de estos registros fueron clasificados dentro de cada uno de los valores del atributo clase. Para el caso de la matriz anterior, para el valor "a" del atributo clase (correspondiente al valor "3.5 - 4.0"), 13755 fueron correctamente clasificadas, mientras que 1539 fueron puestas en la clasificación "b".

2.2.2. Elementos de resultados de WEKA para reglas de asociación

En la herramienta se muestran los resultados de las reglas con el soporte y la confianza anteriormente definidos en la sección de Metodología de Pruebas. Como ejemplo se tiene:

Apriori
=======

Minimum support: 0.05
Minimum metric <confidence>: 0.9
Number of cycles performed: 19

Generated sets of large itemsets:

Size of set of large itemsets L(1): 16

Size of set of large itemsets L(2): 57

Size of set of large itemsets L(3): 43

Size of set of large itemsets L(4): 12

Best rules found:

1. NOM_PROGRAMA=Ingeniería Electronica CLASIFICACIÓN=4.0 – 4.5 2455 ==> TPERDIDOS=0 2418 conf:(0.98)

El soporte mínimo(minimum support) es el que indica el valor establecido como soporte mínimo para la aceptación de la regla; toda regla que se encuentre por debajo de ese soporte no será tenida en cuenta.

La métrica de confianza es el rango de error en el que se quiere que estén las reglas a ser tomadas en el modelo. Este valor debe ser previamente establecido, en el caso del ejemplo se dio un 90% de precisión.

2.2.3. Cómo comparar árboles con reglas (resultantes de WEKA)

1. Reglas: están definidas por:
    a. Support (o coverage) = #instancias clasificadas correctamente (para las cuales se cumplen todas las condiciones de la regla)
    b. Confidence o accuracy = #total de instancias a las que aplica la regla (para las cuales se cumplen las condiciones del antecedente)

2. Árboles: cada camino desde la raíz a una hoja, puede ser convertido en una regla. Weka da ciertos valores al lado de cada hoja del árbol. A continuación se muestra una regla de asociación extraída a partir de los resultados que dio WEKA para una consulta determinada:

    Si (1<= TPerdidos <= 5) y (11<= TCursados <= 15) y (Prg = 'C') → Promedio = 'MedioAlto' (12.0/4.0)

Al costado derecho se puede ver los números que da WEKA para cada hoja del árbol. Estos significan:

a) 12.0: Indica el total de instancias para las cuales aplica el camino desde la raíz hasta la hoja
b) 4.0: Indica el número de instancias que son mal clasificadas con el camino generado.

Cómo se traducen estos valores en términos de support y confidence?

Support = Total de instancias aplicables – Instancias mal clasificadas =

Instancias bien clasificadas

→ support = 12 – 4 = 8

Confidence = Instancias bien clasificadas / Total de instancias aplicables =

→ confidence = 8 / 12 = 2 / 3

El proceso final consiste en traducir todas las ramas resultantes del árbol en términos de support y confidence, para después establecer un umbral mínimo para ambos valores, que se usarán en el momento de aplicar reglas al mismo conjunto de datos. Por último, se seleccionarán solo las ramas del árbol que cumplan con los umbrales establecidos.

Las reglas resultantes serán comparables con las otorgadas por el algoritmo a-priori.

2.3. DESCRIPCIÓN DE DATOS DE ORIGEN

A continuación se hará una descripción del dominio de aplicación que se refiere a información histórica académica de estudiantes de la Pontificia Universidad Javeriana. Para poder analizar los datos se debe tener una coherencia entre ellos formando una estructura donde se muestren las relaciones entre los mismos.

Relaciones de las tablas existentes con descripción de contenido

- **ACTUALIZACION_ESTUDIANTE:**

	Column Name	Data Type	Length	Allow Nulls
	NEXTVAL	float	8	✓
🔑	ID	nvarchar	36	
	IDANTERIOR	nvarchar	36	✓
	APELLIDO1	nvarchar	50	✓
	NOMBRE	nvarchar	50	✓
	APELLIDO2	nvarchar	50	✓
	TELEFONO	numeric	9	✓
	DIRECCION	nvarchar	150	✓
	E_MAIL1	nvarchar	50	✓
	E_MAIL2	nvarchar	50	✓
	EPS	nvarchar	10	✓
	TELACUDIENTE	numeric	9	✓
	PLANCOMP	nvarchar	50	✓
	NOMBRECOTIZANTE	nvarchar	50	✓
	IDCOTIZANTE	numeric	9	✓
	NUMBENEFICIARIOS	numeric	5	✓
	NUMEPS	nvarchar	30	✓
	FECHAVENCEEPS	smalldatetime	4	✓
	COLEGIO	nvarchar	80	✓
	FECHAUPDATE	smalldatetime	4	✓
	OTROMEDICO	nvarchar	70	✓
	TIPOID	nvarchar	2	✓
	TIPOIDANTERIOR	nvarchar	2	✓
	TIPOIDCOTIZANTE	nvarchar	2	✓
	SEXO	nvarchar	1	✓
	FECHANACIM	smalldatetime	4	✓
	ANOGRADO	numeric	5	✓
	DIRACUDIENTE	nvarchar	150	✓
	CIUDADACUDIENTE	nvarchar	20	✓
	APELLIDO1COTIZ	nvarchar	50	✓
	APELLIDO2COTIZ	nvarchar	50	✓
	SIAFILIADO	nvarchar	1	✓
	NOMBREACUD	nvarchar	50	✓
	APELLIDO1ACUD	nvarchar	50	✓
	APELLIDO2ACUD	nvarchar	50	✓
	SEMINGRESO	nvarchar	1	✓
	ANOINGRESO	nvarchar	4	✓
	SECUENCIAL	numeric	13	✓
	CELULAR	numeric	9	✓
	BEEPER	numeric	9	✓
	CODIGO_BEEPER	numeric	5	✓
	TELEFONO_OFICINA	numeric	13	✓
	EXT_OFICINA	numeric	5	✓
	FECHAUPDATEID	smalldatetime	4	✓
	LDAP_USER	nvarchar	30	✓

Tabla 7 -ACTUALIZACION_ESTUDIANTE

- **Descripción:**

  En esta tabla se tiene toda la información personal del estudiante. La columna NEXTVAL indica un número secuencial único que identifica al estudiante, igualmente la columna ID es la cédula del estudiante que nos indica un número único por estudiante (Llave primaria). Esta tabla se relacionará con la mayoría de tablas ya que son los datos de cada estudiante de la universidad. Se tienen actualmente 6052 registros en esta tabla.

Ejemplo de registro:

NEXTVAL	ID	IDANTERIOR	APELLIDO1	NOMBRE	APELLIDO2	TELEFONO	DIRECCION	E_MAIL1
2252	3146523	80080900248	GALVIS	MARIO	GUERRERO	6335776	calle 140 # 16 - 94	m.galvis@javer

Tabla 8 - Ejemplo 1

- **ACTUALIZACION_GEN_DEPARTAMENTO**

	Column Name	Data Type	Length	Allow Nulls
	ID_DEPTO	nvarchar	5	
	NOM_DEPTO	nvarchar	60	
	ID_FACULTAD	nvarchar	4	✓

Tabla 9 - ACTUALIZACION_GEN_DEPARTAMENTO

- **Descripción:**

  En esta tabla se encuentra la identificación del departamento, el nombre del departamento y a que facultad pertenece. Se puede notar en el ejemplo del registro que el ID_DEPTO y el ID_FACULTAD son llaves conjuntas.

Ejemplo de registro:

ID_DEPTO	NOM_DEPTO	ID_FACULTAD
S	INGENIERIA DE SISTEMAS	I
I	INGENIERIA INDUSTRIAL	I
E	INGENIERIA ELECTRONICA	I
C	INGENIERIA CIVIL	I

Tabla 10 - Ejemplo 2

- **ACTUALIZACION_GEN_FACULTAD**

	Column Name	Data Type	Length	Allow Nulls
🔑	ID_FACULTAD	nvarchar	4	
	NOM_FACULTAD	nvarchar	50	✓

Tabla 11 - ACTUALIZACION_GEN_FACULTAD

- **Descripción:**

  Esta tabla contiene la identificación de cada una de las facultades de la universidad con su respectivo nombre. Se puede ver que el ID_FACULTAD es la Llave primaria de esta tabla.

Ejemplo de Registros:

	ID_FACULTAD	NOM_FACULTAD
	F	FILOSOFIA
	G	ARTES
	H	COMUNICACION Y LENGUAJE
	I	INGENIERIA
	J	CIENCIAS JURIDICAS
	K	VICERRECTORIA DEL MEDIO
	L	LENGUAS
	M	MEDICINA

Tabla 12 - Ejemplo 3

- **ACTUALIZACION_GEN_PROGRAMA**

	Column Name	Data Type	Length	Allow Nulls
🔑	ID_PROGRAMA	nvarchar	4	
	NOM_PROGRAMA	nvarchar	100	✓
	ID_FACULTAD	nvarchar	4	✓

Tabla 13 - ACTUALIZACION_GEN_PROGRAMA

- **Descripción:**

  En esta tabla se encuentra la información de los programas de cada facultad. En este caso se tiene únicamente los programas de la facultad de ingeniería. Se tiene el ID_PROGRAMA como Llave primaria de la tabla.

Ejemplo de registros:

	ID_PROGRAMA	NOM_PROGRAMA	ID_FACULTAD
	A	Tecnología en construcción de edif	I
	C	Ingeniería Civil	I
	E	Ingeniería Electronica	I
	G	Esp. Gerencia de construcciones	I
	I	Ingeniería Industrial	I
	K	Esp. Geotecnia Vial y Pavimentos	I
	L	Maestría en Ingeniería Electrónica	I
	M	Esp. Sistemas Gerenciales	I
	R	Instituto geofísico	I
	S	Ingeniería de Sistemas	I

Tabla 14 - Ejemplo 4

- **ACTUALIZACION_MV_EST_ESTADOS**

	Column Name	Data Type	Length	Allow Nulls
	SEQ	float	8	✓
	FAC	nvarchar	4	✓
	PRG	nvarchar	4	✓
	ESTADO	numeric	5	✓
	FEC_IN	numeric	5	✓
	FECHA_GRADO	numeric	5	✓

Tabla 15 - ACTUALIZACION_MV_EST_ESTADOS

- **Descripción:**

  En esta tabla se tiene la información de los estudiantes que se encuentran en determinado programa de determinada facultad. Se puede notar claramente que se tiene una relación con la tabla ACTUALIZACION_GEN_PROGRAMA y ACTUALIZACION_GEN_FACULTAD.

Ejemplo de Registro:

SEQ	FAC	PRG	ESTADO	FEC_IN	FECHA_GRADO
2252	I	S	1	991	<NULL>

Tabla 16 - Ejemplo 5

- **ACTUALIZACION_MV_EST_NOTASMATERIAH**

	Column Name	Data Type	Length	Allow Nulls
	ID	float	8	✓
	PRG	nvarchar	4	✓
	FAC_M	nvarchar	4	✓
	PRG_M	nvarchar	1	✓
	MAT	numeric	5	✓
	NOTA	numeric	5	✓
	PER_CIVIL	numeric	5	✓

Tabla 17 - ACTUALIZACION_MV_EST_NOTASMATERIAH

- **Descripción:**

  En esta tabla se tiene el historial de las notas de cada uno de los alumnos desde el momento que ingresaron a la universidad. El ID es el número que identifica al estudiante viene de la relación con la tabla ACTUALIZACION_ESTUDIANTE explicada anteriormente con la columna NEXTVAL. Se puede notar también que se tiene una relación con la tabla ACTUALIZACION_GEN_PROGRAMA y con la tabla ACTUALIZACION_GEN_FACULTAD. Se tiene la calificación obtenida de cada una de las materias vistas y el periodo civil en que el estudiante inscribió la materia. (ej: 223 = el tercer periodo del 2003).

Ejemplo de Registros:

	ID	PRG	FAC_M	PRG_M	MAT	NOTA	PER_CIVIL
	2252	S	R	C	20	44	233
	2252	S	I	I	164	45	233
	2252	S	I	S	54	36	233
	2252	S	I	Y	18	44	233
	2252	S	I	P	20	31	233
	2252	S	I	S	47	39	233
	2252	S	I	S	56	45	233
	2252	S	I	I	128	43	223

Tabla 18 - Ejemplo 6

- **MV_EST_SEC_ACADEM:**

	Column Name	Data Type	Length	Allow Nulls
	ID	nvarchar	36	✓
	PER_CIVIL	int	4	✓
	PRG	nvarchar	4	✓
	CCURSADOS	int	4	✓
	CPERDIDOS	int	4	✓
	PUNTOS	float	8	✓
	PROMEDIO	float	8	✓

Tabla 19 - MV_EST_SEC_ACADEM

- **Descripción:**

  En esta tabla se tiene la información de cada uno de los estudiantes donde se puede observar el periodo civil el programa en el que se encuentra, los créditos cursados y los créditos perdidos. Se tiene también el promedio acumulado de los estudiantes. El ID es el identificador del estudiante.

Ejemplo de registros:

	ID	PER_CIVIL	PRG	CCURSADOS	CPERDIDOS	PUNTOS	PROMEDIO
	10005028	233	A	13	2	51	3,92
	17652320	233	A	13	2	46,8	3,6
	52260167	233	A	13	5	42,9	3,3
	52425316	233	A	13	2	53	4,08
	52527736	233	A	13	0	53,2	4,09
	75070794	233	A	13	2	52,9	4,07
	7695311	233	A	13	2	47,5	3,65
	79793078	233	A	13	7	39,8	3,06
	79968825	233	A	13	8	39,4	3,03
	80422740	233	A	13	5	48,5	3,73

Tabla 20 - Ejemplo 7

- **GEN_ACTIVIDADES_EXTRA**

	Column Name	Data Type	Length	Allow Nulls
	TIPO	nvarchar	10	
	ACTIVIDAD	nvarchar	15	
🔑	ID_ACTIVIDAD	numeric	5	

Tabla 21 - GEN_ACTIVIDADES_EXTRA

- **Descripción:**

  Esta tabla contiene una lista de Actividades que los estudiantes deben elegir durante el proceso de inscripción de datos. Se puede notar que ID_ACTIVIDAD es la llave primaria de esa tabla.

Ejemplo de registros:

	TIPO	ACTIVIDAD	ID_ACTIVIDAD
	DEPORTE	Tenis	1
	DEPORTE	Baloncesto	26
	DEPORTE	Tenis de mesa	2
	DEPORTE	Fútbol	3
	DEPORTE	Taekwondo	4
	DEPORTE	Ajedrez	5
	DEPORTE	Atletismo	6
	DEPORTE	Voleibol	7
	DEPORTE	Squash	8
	DEPORTE	Natación	9
	DEPORTE	Esgrima	10
	DEPORTE	Patinaje	11
	DEPORTE	Ciclismo	12
	ARTE	Música	13
	ARTE	Literatura	14
	ARTE	Fotografía	15
	ARTE	Pintura	16
	ARTE	Cine	17
	OTROS	Montañismo	18
	OTROS	Astronomía	19
	OTROS	Filatelia	20
	OTROS	Numismática	21
	OTROS	Escalada	22
	OTROS	Pastoral	23
	OTROS	Sector social	24
	OTROS	Periodismo	25
	OTROS	Otros	-3
	DEPORTE	Otros	-2
	ARTE	Otros	-1

Tabla 22 - Ejemplo 8

- **EST_ACTIVIDADES_EXTRA**

	Column Name	Data Type	Length	Allow Nulls
	EST_SEC	numeric	13	
	ID_ACTIVIDAD	numeric	5	
	OTROS	nvarchar	500	✓

Tabla 23 - EST_ACTIVIDADES_EXTRA

- **Descripción:**

  En esta tabla se muestran las actividades extras que realizan los estudiantes. Esta información se ingresa en el momento de actualizar o inscribir datos. Existe una relación directa entre esta tabla y GEN_ACTIVIDADES_EXTRA.

Ejemplo de registros:

EST_SEC	ID_ACTIVIDAD	OTROS
1999	-3	servicio social
3279	-3	<NULL>
3279	-1	<NULL>
3279	-2	<NULL>
3559	-1	<NULL>
3559	-2	<NULL>
323	-3	<NULL>
323	-1	<NULL>
323	-2	<NULL>
1999	-1	<NULL>
1999	-2	golf

Tabla 24 - Ejemplo 9

- **EST_EXP_PROFESIONAL**

	Column Name	Data Type	Length	Allow Nulls
🔑	IDEXP	numeric	5	
	INSTITUCION	nvarchar	100	✓
	CARGO	nvarchar	50	✓
	FECHAINGRESO	smalldatetime	4	✓
	FECHARETIRO	smalldatetime	4	✓
	LABORES	nvarchar	300	✓
🔑	ID_EST	numeric	13	

Tabla 25 - EST_EXP_PROFESIONAL

- **Descripción:**

  En esta tabla se encuentran los datos de la experiencia laboral de cada estudiante. Se puede ver que el ID del estudiante (NEXVAL de la tabla de estudiante) actúa en conjunto con el IDEXP.

Ejemplo de registros:

IDEXP	INSTITUCION	CARGO	FECHAINGRESO	FECHARETIRO	LABORES	ID_EST
0	NCR Colombia	Consultor de tecnologia	01/01/2003	01/06/2003	Actividades de Tecnologia y soporte interno a la empresa	2252

Tabla 26 - Ejemplo 10

- **EST_IDIOMAS**

	Column Name	Data Type	Length	Allow Nulls
🔑	EST_SEC	numeric	13	
	IDIOMA	numeric	5	
	HABLAR	numeric	5	
	LEER	numeric	5	
	ESCRIBIR	numeric	5	
	NOMBREEXAMEN	nvarchar	50	✓
	PUNTAJEOBTENIDO	nvarchar	20	✓
	MAXIMOPUNTAJEPRU	nvarchar	20	✓
🔑	IDIDIOMA	numeric	5	
	ESCUCHAR	numeric	5	✓

Tabla 27 - EST_IDIOMAS

- **Descripción:**

  En esta tabla se muestra el nivel de idiomas de cada estudiante en porcentajes de hablar, escuchar y escribir. El EST_SEC es el identificador del estudiante que se relaciona con el NEXVAL de la tabla estudiante.

Ejemplo de registros:

EST_SEC	IDIOMA	HABLAR	LEER	ESCRIBIR
2247	1	100	100	100

Tabla 28 - Ejemplo 11

Durante la especificación de esta estructura se encontraron los siguientes problemas:

- Diferentes claves para representar un mismo elemento. El estudiante se puede identificar por su ID o el NEXTVAL.
- Carencia de tablas que den explicación a datos de tablas existentes.
- En algunas tablas faltan los registros.
- Existen registros que tienen datos erróneos.
- No se pueden relacionar algunas tablas por desorden en el diseño.
- Se repiten tablas.

Se debe proceder a la siguiente actividad que consiste en realizar una limpieza de los datos para hallar una coherencia completa de las tablas y destrucción de datos inútiles.

2.4. LIMPIEZA DE DATOS Y DISCRETIZACION

A continuación se mostrará el proceso que se ejecutó previo a la ejecución de los algoritmos de Minería de Datos, de tal manera que los datos de origen fueran adecuados para el desarrollo del proceso.

2.4.1. Limpieza de los datos

Como se mencionó en el capítulo 1, para una investigación de este tipo, el proceso de limpieza y discretización de los datos de origen es de suma importancia. En este caso específico, nos interesarán los siguientes problemas en los datos.

APELLIDO2	EPS	PLANCOMP	COLEGIO
	-1		
DELGADO	-1		
VILLAMIL	1		COLEGIO CALASANZ
MORA	13	13	GENERAL RAFAEL REYES
CADAVID	null	null	COLEGIO CALASANZ PEREIRA
BEDOYA	3	COLSANITAS 1	LICEO FRANCES DE PEREIRA
ECHEVERRY	0	0	CALASANZ
ARANGO	-1		
MEJIA	4	4	COLEGIO SALESIANO
CHUJFI	5	5	FUNDACION LICEO INGLES
CARDONA	13		POLICIA NACIONAL
PULGARIN	-1		
GRAJALES	-1		
WITTORF	-1		
POLANIA	-1		

Ilustración 17 – Limpieza de los datos

No Normalización:

Para un mismo valor de variable se puede presentar varias representaciones. En la ilustración 15 se puede ver que para un solo valor real de un atributo: "Colegio Calasanz", existen al menos 3 representaciones distintas tan solo en esta pequeña extracción de datos. El investigador debe unificar estos valores. Si las tablas de origen se encontraran en 2NF (Segunda Forma Normal)3, este problema no se presentaría. Es decir, para este caso, tendría que tenerse una tabla "Colegios" donde se tengan todos los posibles colegios y en la tabla de personas que se ve en la figura, solo una referencia a dicha tabla.

[3] 2NF: "Una relación está en segunda forma normal (2NF) si y solo si está en 1NF y todos los atributos no clave dependen por completo de la clave primaria."

Sin embargo, en este formato se encontraban los registros originales. Adicionalmente, para poder utilizar una base de datos relacional con la herramienta WEKA, se deben importar los datos a través de una consulta de la base de datos. Dicha consulta debe incluir todos los valores (de una o más tablas) necesarios para lo que se quiere descubrir. Si los datos provienen de más de una tabla, deberán utilizarse *joins*, que permitan la unión de las tablas por algún criterio, por lo que finalmente, los datos que se utilicen en WEKA tendrán este formato, aunque no se presentará el problema expuesto en el párrafo anterior.

Valores nulos o ausencia de valores:

También se puede observar en la figura que se tienen valores *null* o valores vacíos para ciertas columnas. Se debe previamente tomar una determinación con respecto a que valor se le otorgará a una columna con alguno de estos valores. Se puede ignorar la ausencia de valor o reemplazar por un valor determinado (el más frecuente en el conjunto de datos por ejemplo).

2.4.2. Discretización

Adicionalmente a la limpieza de los datos, se debe realizar un proceso de discretización para poder aplicar algoritmos de Minería de Datos. Este proceso consiste en limitar los posibles valores de un atributo, para convertir una variable continua en una variable discreta y de esta manera poder aplicar los algoritmos de Minería de Datos para que produzcan resultados de utilidad.

Es claro que se sacrifica precisión de la clasificación en este proceso, pero se debe recordar también que excesiva precisión no permite su fácil aplicación en el dominio de los datos.

Hay dos razones principales que motivan la discretización:

1. El algoritmo a-priori de reglas de asociación no acepta atributos numéricos para su aplicación, por lo que todos los atributos de este tipo, deben ser convertidos a texto.

2. Sin embargo, no es suficiente con convertir estos atributos a texto, pues generar por ejemplo un árbol

a partir de estos atributos, generaría tantas clasificaciones como valores existan y por lo tanto, habría una cantidad enorme de clasificaciones, lo cual no brinda utilidad a la hora de su aplicación.

Por tanto, se utilizó discretización en los siguientes casos:

- Atributo "Clasificación": este atributo corresponde al promedio ponderado acumulado de un estudiante. Este atributo era numérico, con una cifra decimal, entre 0.0 y 5.0. Se pasó a caracteres y se crearon las siguientes posibilidades de valor:

```
0.0 - 2.0
2.0 - 2.5
2.5 - 3.0
3.0 - 3.5
3.5 - 4.0
4.0 - 4.5
4.5 - 5.0
```

  Esta discretización permitió la clasificación de los registros en 7 grupos distintos, lo cual redujo el número de clasificaciones resultantes y permitió producir árboles y reglas de mayor utilidad.

2.5. INGRESO DE DATOS A LA HERRAMIENTA

Las herramientas de Minería de Datos se basan en grandes volúmenes de datos ya sean en bases de datos o en bodegas de datos. En este caso se usa una base de datos con registros históricos de hasta el primer semestre del 2004. Para que WEKA interprete los datos se debe llevar a cabo un proceso previo.

Se debe crear un archivo llamado DatabaseUtils.props en la carpeta del directorio de WEKA. En este archivo se tiene que definir el driver de la base de datos con el que se va a trabajar, en este caso se deben escribir las siguientes lineas:

```
jdbcDriver=sun.jdbc.odbc.JdbcOdbcDriver
dbcURL=jdbc:odbc:dbname
```

bdname es el nombre del origen de datos que tiene que crearse antes de la ejecución de la herramienta (DNS).

Luego de realizar este proceso se puede dar inicio a la herramienta. En el caso de WEKA se carga la BD y estos datos una vez cargados se pueden exportar a un tipo de archivo reconocido por WEKA como arff (Attribute-Relation File Format). Es un archivo ASCII de texto que describe una lista de instancias que comparten un conjunto de atributos- Este tipo de archivo fue desarrollado por el departamento de ciencia de computo de la Universidad de Waikato. Como ejemplo se tiene:

```
@relacion ambiente

@attribute ambiente(soleado, medio, lluvioso)
@attribute temperatura real
@attribute viento (true, false)
@attribute juega (si, no)

@data
Soleado,35,false,no
Medio,29,false,no
Medio,32,true,si
Lluvioso,3,true,no
Lluvioso,24,false,si
Soleado,15,true,si
```

Se puede notar que en la parte inicial del ejemplo se describen los atributos que tiene el conjunto de datos con sus posibles valores. Luego se encuentran los datos relacionados en cada una de las instancias.

Soleado,35,false,no

Si esta soleado, la temperatura es de 35 grados y no hay viento, entonces se decide no jugar, probablemente por que seria perjudicial este tipo de clima para los jugadores.

2.6. FORMULACIÓN DE PREGUNTAS

- ¿Qué hay en el conjunto de datos?

En el caso de esta investigación en colaboración con el grupo ORBIS, que proporcionó los datos de origen, se plantearon ciertas preguntas que podrían de ser interés para la Universidad. La formulación de las preguntas serán necesarias para la preparación de las consultas que van a ser usadas para el desarrollo de la investigación. Las consultas se usarán para la aplicaciòn de las pruebas y traerán como resultado información de utilidad para la universidad de acuerdo a los datos brindados. Las preguntas a resolver son:

- ¿Qué características de un estudiante pueden dar indicios claros de cómo será su desempeño en su vida académica (en cuanto a su promedio ponderado acumulado)?

- ¿Qué características de un estudiante pueden dar indicios claros de cuál será el estado de su matrícula académica en el futuro?

- ¿Se pueden responder las preguntas planteadas?

Una vez estén planteadas estas preguntas, es necesario saber si se pueden responder estas preguntas con base en los recursos que se tienen disponibles. En este caso el recurso necesario para dar respuesta a estas preguntas son los datos origen y las herramientas de Minerá de Datos. Se deben analizar las propiedades de los estudiantes y aplicar las técnicas de minería para conocer las relaciones que existen entre los diferentes campos y asi obtener el modelo que nos servirá para conocer las características de los estudiantes y asi obtener respuesta a las preguntas formuladas.

3. RESULADOS Y ANALISIS DE RESULTADOS

Este capítulo se encuentra dividido en dos secciones. En la primera se muestran los criterios de selección de registros de la base de datos de origen o consultas que fueron diseñadas y seleccionadas para la investigación. Para cada uno de ellos se muestra el comando SQL que permite la extracción de sus datos, así como el objetivo que se persigue en relación a los resultados prácticos que brinde.

En la segunda sección se muestran los resultados de las pruebas descritas en el capítulo 2, aplicadas a cada una de las consultas seleccionadas. Así mismo, se presenta un análisis de los resultados de cada una de las pruebas y un análisis global de las mismas.

3.1. DISEÑO DE CONSULTAS

3.1.1. Consulta #1

Comando SQL:

```
SELECT programas.nom_programa, notasgeneral.tcursados,
notasgeneral.tperdidos, notasgeneral.clasificacion, estudiante.sexo
FROM estudiante, notasgeneral, programas, estado WHERE
notasgeneral.id=estudiante.id AND estado.seq=estudiante.nextval
AND estado.prg=id_programa
```

Objetivo:

Con esta consulta se pretende conocer si el programa de un estudiante así como el número de créditos perdidos, créditos cursados y su sexo influyen de una u otra manera en la clasificación o el promedio del mismo.

En este caso se hizo una discretización de los datos en rangos de 0.5 para la clasificación de promedios.

Esta consulta podría ser útil para identificar el promedio de las personas así como también para poder predecirlo en un momento dado, de acuerdo a los parámetros de entrada.

3.1.2. Consulta #2

Comando SQL:

SELECT sexo, tperdidos, tcursados, prom_acum, estadomatricula FROM matriculas_recortadas WHERE tcursados='1-5' OR tcursados='6-10' OR tcursados='11-15' OR tcursados='16-20' OR tcursados='21-25'

Objetivo:

A través de esta consulta se pretende producir reglas en torno al atributo clase **estadomatricula**, que se extrae de la tabla *matriculas_recortadas* y que es de tipo alfanumérico, con los siguientes posibles valores e interpretaciones:

Valor	*Interpretación*
'C'	En matrícula condicional
'S'	Sin matrícula condicional

Tabla 29 - Atributo estado matrícula

La tabla *matriculas_recortadas* no hace parte del conjunto original de datos, pero es resultado del trabajo que se hizo sobre los mismos para poder usarla para la investigación. En ella se almacenaron datos sobre la matrícula condicional de los estudiantes, eliminando atributos no necesarios para esta consulta.

Se restringió la consulta a personas que hayan cursado hasta 25 créditos en el semestre, pues más allá de este límite, se trata de casos excepcionales.

3.1.3. Consulta #3

Comando SQL:

SELECT sexo, id_depto, nom_facultad, display_name from retiromateria, retiromateriacausa, genrazonretmat, estudiante, facultades where (retiromateria.id_retiro = retiromateriacausa.id_retiro) AND (retiromateriacausa.id_razon = genrazonretmat.id_razon) AND (estudiante.nextval = retiromateria.id_estudiante) AND (Facultades.id_facultad=id_fac)

Objetivo:

A través de esta consulta se pretende realizar un análisis en torno a los retiros de materias, para poder encontrar la correlación de distintos factores y características de un estudiante (como su género, departamento y facultad) con la razón por la cual efectuaron dicho retiro.

Esta consulta podría ser útil para la clasificación de materias y para la determinación de que razones de retiro son más frecuentes para cada una de ellas. De esta manera se podría encontrar que, por ejemplo, para una materia dada se repite con mucha frecuencia que la razón de su retiro es "problemas con el profesor" y así poder plantear una solución para esta situación.

3.2. PRUEBA DE VELOCIDAD DE EJECUCIÓN

Como se explicó en la sección de metodología, se aplicó cada uno de los algoritmos en 20 ocasiones, realizando mediciones respecto al tiempo de ejecución.

Se obtuvieron dos valores de media, uno para cada algoritmo. Se debe establecer si una de las medias es mayor que la otra por medio de algún procedimiento confiable. Sin embargo, para poder hacer esta afirmación, se debe realizar una prueba de hipótesis que demuestre que la diferencia entre las dos medias es mayor a cero. Esta prueba de hipótesis esta dada por la siguiente probabilidad:

Se plantean las hipótesis:

- Hipótesis Nula: $H_0 = \mu_1 - \mu_2 = 0$

- Hipótesis Alternativa: $H_1 = \mu_1 - \mu_2 > 0$

Como no se conocen las varianzas reales, se utiliza el estadístico *t de student*. El estadístico a utilizar, con una significancia del 5%, es:

$$t_{n_1 + n_2 - 2, 0.995} = t_{38, 0.995}$$

Este valor corresponde aproximadamente a **2 .58**. Este valor corresponde al valor crítico para la prueba.

El valor de prueba (que se debe calcular para cada una de las consultas), se obtiene de la siguiente fórmula:

$$=((\overline{x_1} - \overline{x_2}) - (\mu_1 - \mu_2)) / S_p \cdot \sqrt{(1/n_1 + 1/n_2)}$$

, donde:

$$S_p^{\,2} = ((n_1 - 1)S_1^{\,2} + (n_1 - 1)S_2^{\,2}) / n_1 + n_2 - 2$$

Los siguientes fueron los resultados obtenidos para cada una de las consultas:

- ***Consulta #1***

ÁRBOLES DE DECISION:

Prueba #	*Tiempo Inicial*	*Tiempo Final*	*Tiempo de Ejecución*	
1	*19:42:46,27*	*19:42:46,48*	*0:00:00,21*	*21*
2	*19:43:14,59*	*19:43:14,79*	*0:00:00,20*	*20*
3	*19:43:56,84*	*19:43:57,09*	*0:00:00,25*	*25*
4	*19:44:28,60*	*19:44:28,81*	*0:00:00,21*	*21*
5	*19:45:02,27*	*19:45:02,48*	*0:00:00,21*	*21*
6	*19:45:26,43*	*19:45:26,63*	*0:00:00,20*	*20*
7	*19:46:09,84*	*19:46:10,02*	*0:00:00,18*	*28*
8	*19:46:45,18*	*19:46:45,38*	*0:00:00,20*	*20*
9	*19:47:14,51*	*19:47:14,71*	*0:00:00,20*	*20*
10	*19:47:47,21*	*19:47:47,43*	*0:00:00,22*	*22*
11	*19:48:18,87*	*19:48:19,10*	*0:00:00,23*	*23*
12	*19:49:21,70*	*19:49:21,90*	*0:00:00,20*	*20*
13	*19:49:59,51*	*19:49:59,71*	*0:00:00,20*	*20*
14	*19:50:47,87*	*19:50:48,09*	*0:00:00,22*	*22*
15	*19:51:31,35*	*19:51:31,56*	*0:00:00,21*	*21*
16	*19:52:15,57*	*19:52:15,77*	*0:00:00,20*	*20*
17	*19:54:30,15*	*19:54:30,35*	*0:00:00,20*	*20*
18	*19:55:07,37*	*19:55:07,57*	*0:00:00,20*	*20*
19	*19:56:00,35*	*19:56:00,57*	*0:00:00,22*	*22*
20	*19:56:36,18*	*19:56:36,38*	*0:00:00,20*	*20*
		Promedio =	*0:00:00,21*	
		Varianza =	*4,2211*	

REGLAS DE ASOCIACIÓN:

Prueba #	*Tiempo Inicial*	*Tiempo Final*	*Tiempo de Ejecución*	
1	*19:22:12,21*	*19:22:12,27*	*0:00:00,06*	*6*
2	*19:25:39,56*	*19:25:39,63*	*0:00:00,07*	*7*
3	*19:26:24,45*	*19:26:24,62*	*0:00:00,17*	17
4	*19:27:03,42*	*19:27:03,59*	*0:00:00,17*	17
5	*19:27:41,65*	*19:27:41,74*	*0:00:00,09*	9
6	*19:28:21,62*	*19:28:21,71*	*0:00:00,09*	9
7	*19:29:39,52*	*19:29:39,60*	*0:00:00,08*	8
8	*19:30:12,21*	*19:30:12,27*	*0:00:00,06*	6
9	*19:34:33,93*	*19:34:34,06*	*0:00:00,13*	13
10	*19:34:59,88*	*19:34:59,96*	*0:00:00,08*	8
11	*19:35:54,01*	*19:35:54,06*	*0:00:00,05*	5
12	*19:36:43,62*	*19:36:43,68*	*0:00:00,06*	6
13	*19:37:35,63*	*19:37:35,71*	*0:00:00,08*	8
14	*19:38:12,54*	*19:38:12,63*	*0:00:00,09*	9
15	*19:38:43,49*	*19:38:43,57*	*0:00:00,08*	8
16	*19:39:29,92*	*19:39:29,99*	*0:00:00,07*	7
17	*19:40:04,81*	*19:40:04,87*	*0:00:00,06*	6
18	*19:40:42,51*	*19:40:42,60*	*0:00:00,09*	9
19	*19:41:11,95*	*19:41:12,02*	*0:00:00,07*	7
20	*19:42:11,68*	*19:42:11,74*	*0:00:00,06*	6
		Promedio =	*0:00:00,09*	
		Varianza =	*11,4184*	

Se puede ver que para el caso del algoritmo de árboles de decisión, se tiene un promedio de 21 centésimas de segundo, mientras que para el caso de reglas de asociación, se tiene un valor para este promedio de 9 centésimas de segundo.

Aunque esta consulta tiene una población objetivo relativamente pequeña (alrededor de 6000 registros), se puede observar una diferencia significativa entre los dos valores. Sin embargo, esta afirmación solo se puede hacer después de realizar el proceso de comparación de la diferencia de las medias.

Para estos datos, se obtuvo:

$$S_1 = 4{,}2211 \qquad S_2 = 11{,}4184 \qquad \mu_1 = 21 \qquad \mu_2 = 9$$

Luego,

$$S_p^2 = ((20-1)(4{,}2211)^2 + (20-1)(11{,}4184)^2)/20+20-2$$

$$S_p^2 = (338{,}5360 + 2477{,}2173)/38 = 74{,}0987$$

$$S_p = 8{,}6080$$

Entonces,

$$t_p = ((21-9)-(0))/S_p.\sqrt{(1/20+1/20)} = 4{,}408..$$

Como el valor t de prueba cae en la zona de rechazo de la hipótesis nula (por ser mayor a 2,58), se concluye que EXISTE evidencia de que se cumple la hipótesis alternativa y que la diferencia entre las medias es mayor que cero, es decir:

Las dos medidas son diferentes con una significancia del 0,5% y por tanto se puede decir con este nivel de confianza, que el algoritmo de árboles de decisión es más lento que el algoritmo de reglas de asociación para esta consulta.

- ***Consulta #2***

ÁRBOLES DE DECISION:

Prueba #	Tiempo Inicial	Tiempo Final	Tiempo de Ejecución	
1	13:59:09,60	13:59:09,82	0:00:00,22	22
2	14:01:24,26	14:01:24,48	0:00:00,22	22
3	14:04:57,89	14:04:58,07	0:00:00,18	18
4	14:05:26,70	14:05:26,85	0:00:00,15	15
5	14:05:59,09	14:05:59,26	0:00:00,17	17
6	14:06:38,82	14:06:38,98	0:00:00,16	16
7	14:07:07,40	14:07:07,60	0:00:00,20	20
8	14:07:33,00	14:07:33,09	0:00:00,09	9
9	14:08:02,89	14:08:03,04	0:00:00,15	15
10	14:08:29,45	14:08:29,60	0:00:00,15	15
11	14:09:55,09	14:09:55,28	0:00:00,19	19
12	14:10:20,84	14:10:21,01	0:00:00,17	17
13	14:10:46,42	14:10:46,59	0:00:00,17	17
14	14:11:12,31	14:11:12,51	0:00:00,20	20
15	14:11:33,92	14:11:34,09	0:00:00,17	17
16	14:12:04,01	14:12:04,18	0:00:00,17	17
17	14:12:25,73	14:12:25,89	0:00:00,16	16
18	14:12:52,48	14:12:52,64	0:00:00,16	16
19	14:13:19,07	14:13:19,25	0:00:00,18	18
20	14:13:41,39	14:13:41,56	0:00:00,17	17
		Promedio =	0:00:00,17	
		Varianza =	8,028947368	

REGLAS DE ASOCIACIÓN:

Prueba #	Tiempo Inicial	Tiempo Final	Tiempo de Ejecución	
1	14:24:50,10	14:24:50,17	0:00:00,07	7
2	14:25:36,10	14:25:36,15	0:00:00,05	5
3	14:26:00,31	14:26:00,34	0:00:00,03	3
4	14:26:24,60	14:26:24,71	0:00:00,11	11
5	14:26:48,53	14:26:48,57	0:00:00,04	4
6	14:27:11,32	14:27:11,35	0:00:00,03	3
7	14:27:36,48	14:27:36,53	0:00:00,05	5
8	14:28:00,46	14:28:00,56	0:00:00,10	10
9	14:28:24,95	14:28:25,00	0:00:00,05	5
10	14:28:51,48	14:28:51,51	0:00:00,03	3
11	14:29:51,70	14:29:51,75	0:00:00,05	5
12	14:30:13,67	14:30:13,70	0:00:00,03	3
13	14:30:33,92	14:30:33,93	0:00:00,01	1
14	14:30:52,56	14:30:52,60	0:00:00,04	4
15	14:31:18,37	14:31:18,42	0:00:00,05	5
16	14:31:41,12	14:31:41,20	0:00:00,08	8
17	14:32:02,14	14:32:02,15	0:00:00,01	1
18	14:32:24,73	14:32:24,82	0:00:00,09	9
19	14:32:45,10	14:32:45,15	0:00:00,05	5
20	14:33:05,21	14:33:05,28	0:00:00,07	7
		Promedio =	0:00:00,05	
		Varianza =	7,536842105	

Se puede ver que para el caso del algoritmo de árboles de decisión, se tiene un promedio de 17 centésimas de segundo, mientras que para el caso de reglas de asociación, se tiene un valor para este promedio de 5 centésimas de segundo.

Aunque esta consulta tiene una población objetivo relativamente pequeña (alrededor de 6000 registros), se puede observar una diferencia significativa entre los dos valores. Sin embargo, esta afirmación solo se puede hacer después de realizar el proceso de comparación de la diferencia de las medias.

Para estos datos, se obtuvo:

$$S_1 = 8{,}0289 \qquad S_2 = 7{,}5368 \qquad \mu_1 = 17 \qquad \mu_2 = 5$$

Luego,

$$S_p^{\,2} = ((20-1)(8{,}0289)^2 + (20-1)(7{,}5368^2)/20+20-2$$

$$S_p^{\,2} = (1224{,}8014 + 1079{,}2637)/38 = 60{,}6332$$

$$S_p = 7{,}786$$

Entonces,

$$t_p = ((17-5)-(0))/S_p.\sqrt{(1/20+1/20)} = 4{,}873..$$

Como el valor t de prueba cae en la zona de rechazo de la hipótesis nula (por ser mayor a 2,58), se concluye que EXISTE evidencia de que se cumple la hipótesis alternativa y que la diferencia entre las medias es mayor que cero, es decir:

Las dos medidas son diferentes con una significancia del 0,5% y por tanto se puede decir con este nivel de confianza, que el algoritmo de árboles de decisión es más lento que el algoritmo de reglas de asociación para esta consulta.

- ***Consulta #3***

ÁRBOLES DE DECISION:

Prueba #	Tiempo Inicial	Tiempo Final	Tiempo de Ejecución	
1	17:04:17,03	17:04:17,20	0:00:00,17	17
2	17:06:08,82	17:06:09,00	0:00:00,18	18
3	17:06:37,48	17:06:37,67	0:00:00,19	19
4	17:07:12,51	17:07:12,68	0:00:00,17	17
5	17:10:39,48	17:10:39,67	0:00:00,19	19
6	17:11:05,06	17:11:05,23	0:00:00,17	17
7	17:11:30,48	17:11:30,64	0:00:00,16	16
8	17:11:58,28	17:11:58,45	0:00:00,17	17
9	17:12:30,21	17:12:30,39	0:00:00,18	18
10	17:12:52,78	17:12:53,00	0:00:00,22	22
11	17:13:26,53	17:13:26,70	0:00:00,17	17
12	17:13:49,46	17:13:49,64	0:00:00,18	18
13	17:14:13,35	17:14:13,53	0:00:00,18	18
14	17:14:36,57	17:14:36,75	0:00:00,18	18
15	17:14:58,03	17:14:58,21	0:00:00,18	18
16	17:15:18,92	17:15:19,07	0:00:00,15	15
17	17:15:40,85	17:15:41,03	0:00:00,18	18
18	17:16:02,32	17:16:02,50	0:00:00,18	18
19	17:16:23,95	17:16:24,12	0:00:00,17	17
20	17:16:45,35	17:16:45,53	0:00:00,18	18
		Promedio =	0:00:00,18	
		Varianza =	1,881578947	

REGLAS DE ASOCIACIÓN:

Prueba #	Tiempo Inicial	Tiempo Final	Tiempo de Ejecución	
1	17:26:18,92	17:26:18,95	0:00:00,03	3
2	17:27:56,76	17:27:56,85	0:00:00,09	9
3	17:28:57,96	17:28:58,00	0:00:00,04	4
4	17:29:56,98	17:29:57,07	0:00:00,09	9
5	17:36:09,20	17:36:09,23	0:00:00,03	3
6	17:36:47,96	17:36:48,00	0:00:00,04	4
7	17:37:07,28	17:37:07,32	0:00:00,04	4
8	17:37:25,70	17:37:25,75	0:00:00,05	5
9	17:38:05,92	17:38:05,98	0:00:00,06	6
10	17:38:25,53	17:38:25,56	0:00:00,03	3
11	17:38:43,01	17:38:43,09	0:00:00,08	8
12	17:39:03,14	17:39:03,20	0:00:00,06	6
13	17:40:51,85	17:40:51,89	0:00:00,04	4
14	17:41:23,90	17:41:23,95	0:00:00,05	5
15	17:43:02,87	17:43:02,90	0:00:00,03	3
16	17:44:34,96	17:44:35,07	0:00:00,11	11
17	17:46:15,07	17:46:15,10	0:00:00,03	3
18	17:46:35,01	17:46:35,12	0:00:00,11	11
19	17:48:02,62	17:48:02,68	0:00:00,06	6
20	17:48:22,18	17:48:22,21	0:00:00,03	3
		Promedio =	0:00:00,06	
		Varianza =	7,315789474	

Se puede ver que para el caso del algoritmo de árboles de decisión, se tiene un promedio de 18 centésimas de segundo, mientras que para el caso de reglas de asociación, se tiene un valor para este promedio de 6 centésimas de segundo.

Aunque esta consulta tiene una población objetivo relativamente pequeña (alrededor de 700 registros), se puede observar una diferencia significativa entre los dos valores. Sin embargo, esta afirmación solo se puede hacer después de realizar el proceso de comparación de la diferencia de las medias.

Para estos datos, se obtuvo:

$$S_1 = 1{,}8815 \qquad S_2 = 7{,}3157 \qquad \mu_1 = 18 \qquad \mu_2 = 6$$

Luego,

$$S_p^{\,2} = ((20-1)(1{,}8815)^2 + (20-1)(7{,}3157)^2)/20+20-2$$

$$S_p^{\,2} = (67{,}2608 + 1016{,}8698)/38 = 28{,}5297$$

$$S_p = 5{,}3413$$

Entonces,

$$t_p = ((18-6)-(0))/S_p.\sqrt{(1/20+1/20)} = 7{,}1044...$$

Como el valor t de prueba cae en la zona de rechazo de la hipótesis nula (por ser mayor a 2,58), se concluye que EXISTE evidencia de que se cumple la hipótesis alternativa y que la diferencia entre las medias es mayor que cero, es decir:

Las dos medidas son diferentes con una significancia del 0,5% y por tanto se puede decir con este nivel de confianza, que el algoritmo de árboles de decisión es más lento que el algoritmo de reglas de asociación para esta consulta.

3.3. PRUEBA DE PRECISIÓN EN LA CLASIFICACIÓN DE DATOS DE ORIGEN

Para el caso de árboles de decisión, el número de instancias correctamente clasificadas, es uno de los resultados que otorga la herramienta WEKA.
Para el caso de reglas de asocación, cada regla tiene un número correspondiente a la cantidad de instancias que pueden ser clasificadas por la misma, así como el número de instancias que cumplen tanto con los antecendentes, como con las consecuencias de la regla. Por eso, para hallar el número de instancias correctamente clasificadas, se debe usar el programa desarrollado para la realización de las pruebas[4], donde se cuentan los registros que son correctamente clasificados por las reglas arrojadas por la aplicación del algoritmo de Minería de Datos.

- ***Consulta #1***

Árboles de decisión:
Se obtuvieron los siguientes resultados de la aplicación del algoritmo de árboles de decisión en el conjunto de datos de origen:

Correctly Classified Instances	10816	51.8828 %
Incorrectly Classified Instances	10031	48.1172 %
Kappa statistic	0.3024	
Mean absolute error	0.1676	
Root mean squared error	0.2903	
Relative absolute error	80.9727 %	
Root relative squared error	90.2342 %	
Total Number of Instances	20847	

Para este caso, se tiene cerca de un 52% de instancias correctamente clasificadas, así como un 48% de instancias incorrectamente clasificadas.

Reglas de asociación:
Se obtuvieron los siguientes resultados de la aplicación del algoritmo de reglas de asociación en el conjunto de datos de origen:

[4] "Herramienta para la validación de reglas", referirse al anexo 5.2

```
1. CLASIFICACIÓN=4.5 - 5.0 1210 ==> TPERDIDOS=0 1210    conf:(1)
2. NOM_PROGRAMA=Ingeniería Industrial CLASIFICACIÓN=4.0 - 4.5 SEXO=M
   1188 ==> TPERDIDOS=0 1167    conf:(0.98)
3. NOM_PROGRAMA=Ingeniería Electronica CLASIFICACIÓN=4.0 - 4.5 1329 ==>
   TPERDIDOS=0 1304    conf:(0.98)
4. NOM_PROGRAMA=Ingeniería Industrial CLASIFICACIÓN=4.0 - 4.5 2475 ==>
   TPERDIDOS=0 2427    conf:(0.98)
5. CLASIFICACIÓN=4.0 - 4.5 SEXO=F 2126 ==> TPERDIDOS=0 2083
conf:(0.98)
6. CLASIFICACIÓN=4.0 - 4.5 5733 ==> TPERDIDOS=0 5613    conf:(0.98)
7. NOM_PROGRAMA=Ingeniería Industrial CLASIFICACIÓN=4.0 - 4.5 SEXO=F
   1287 ==> TPERDIDOS=0 1260    conf:(0.98)
8. CLASIFICACIÓN=4.0 - 4.5 SEXO=M 3607 ==> TPERDIDOS=0 3530
conf:(0.98)
9. NOM_PROGRAMA=Ingeniería Industrial TCURSADOS=16-20 CLASIFICACIÓN=4.0
   - 4.5 1372 ==> TPERDIDOS=0 1330    conf:(0.97)
10. TCURSADOS=16-20 CLASIFICACIÓN=4.0 - 4.5 SEXO=F 1223 ==> TPERDIDOS=0
    1184    conf:(0.97)
```

El número de registros clasificados correctamente es 5613
El número total de registros es 20847.

Este valor corresponde a un 26,924% del total de los registros[5].

- ***Consulta #2***

Árboles de decisión:
Se obtuvieron los siguientes resultados de la aplicación del algoritmo de árboles de decisión en el conjunto de datos de origen:

Correctly Classified Instances	4858	82.3111 %
Incorrectly Classified Instances	1044	17.6889 %
Kappa statistic	0.6462	
Mean absolute error	0.277	
Root mean squared error	0.3742	
Relative absolute error	55.4068 %	
Root relative squared error	74.8479 %	
Total Number of Instances	5902	

Para este caso, se tiene cerca de un 82% de instancias correctamente clasificadas, así como un 17% de instancias incorrectamente clasificadas.

Reglas de asociación:
Se obtuvieron los siguientes resultados de la aplicación del algoritmo de reglas de asociación en el conjunto de datos de origen:

[5] Resultados obtenidos con la herramienta "Herramienta para la validación de reglas"

```
1. CLASIFICACIÓN=4.0 - 4.5 410 ==> estadomatricula=S 410    conf:(1)
2. tperdidos=0 CLASIFICACIÓN=4.0 - 4.5 383 ==> estadomatricula=S 383
   conf:(1)
3. CLASIFICACIÓN=4.0 - 4.5 410 ==> tperdidos=0 estadomatricula=S 383
   conf:(0.93)
4. CLASIFICACIÓN=4.0 - 4.5 estadomatricula=S 410 ==> tperdidos=0 383
   conf:(0.93)
5. CLASIFICACIÓN=4.0 - 4.5 410 ==> tperdidos=0 383    conf:(0.93)
6. tcursados=1-5 CLASIFICACIÓN=3.5 - 4.0 425 ==> tperdidos=0 390
   conf:(0.92)
7. tcursados=1-5 estadomatricula=S 435 ==> tperdidos=0 397
   conf:(0.91)
```

El número de registros clasificados correctamente es 734
El número total de registros es 5902.

Este valor corresponde a un 12,436% del total de los registros[6].

- ***Consulta #3***

Árboles de decisión:
Se obtuvieron los siguientes resultados de la aplicación del algoritmo de árboles de decisión en el conjunto de datos de origen:

Correctly Classified Instances	224	32.0458 %
Incorrectly Classified Instances	475	67.9542 %
Kappa statistic	0.0784	
Mean absolute error	0.2219	
Root mean squared error	0.34	
Relative absolute error	95.9985 %	
Root relative squared error	100.0292 %	
Total Number of Instances	699	

Para este caso, se tiene cerca de un 32% de instancias correctamente clasificadas, así como un 67% de instancias incorrectamente clasificadas.

Reglas de asociación:
Se obtuvieron los siguientes resultados de la aplicación del algoritmo de reglas de asociación en el conjunto de datos de origen:

[6] Resultados obtenidos con la herramienta "Herramienta para la validación de reglas"

```
1. id_depto=P 67 ==> nom_facultad=INGENIERIA 67    conf:(1)
2. id_depto=F 47 ==> nom_facultad=CIENCIAS 47    conf:(1)
3. sexo=M id_depto=P 44 ==> nom_facultad=INGENIERIA 44
conf:(1)
4. id_depto=I display_name=Alta carga académica 38 ==>
nom_facultad=INGENIERIA 38    conf:(1)
5. id_depto=I display_name=Bajo rendimiento 35 ==>
nom_facultad=INGENIERIA 35    conf:(1)
6. sexo=F id_depto=I 89 ==> nom_facultad=INGENIERIA 86
conf:(0.97)
7. id_depto=E 42 ==> nom_facultad=INGENIERIA 40    conf:(0.95)
8. id_depto=I 185 ==> nom_facultad=INGENIERIA 174
conf:(0.94)
9. sexo=M id_depto=I 96 ==> nom_facultad=INGENIERIA 88
conf:(0.92)
```

El número de registros clasificados correctamente es 117
El número total de registros es 699.

Este valor corresponde a un 16,738% del total de los registros[7].

[7] Resultados obtenidos con la herramienta "Herramienta para la validación de reglas"

3.4. PRUEBA DE PRECISIÓN EN LA CLASIFICACIÓN DE DATOS FUTUROS

- ***Consulta #1***

Árboles de decisión:

Los resultados del algoritmo aplicado a los datos del 2003 en adelante que superan los valores para los parámetros de confianza y soporte establecidos son[8]:

Total	# Clasificadas	# Mal Clasificadas	Soporte	Confianza	Entra
20847	1695	111	7,598215571	0,934513274	1
20847	2532	165	11,35415168	0,934834123	1
20847	2614	132	11,9057898	0,949502678	1

Las reglas que corresponden a estos resultados son:

```
1. TPERDIDOS= 0 TCURSADOS=16-20 NOM_PROGRAMA=Ingeniería Electrónica ==>
   3.5 - 4.0 (1695.0/111.0)
2. TPERDIDOS= 6-10 TCURSADOS=16-20 ==> 3.0 - 3.5 (2532.0/165.0)
3. TPERDIDOS= 1-5 NOM_PROGRAMA = Ingeniería Industrial ==> 3.5 - 4.0
(2614.0/132.0)
```

Reglas eliminadas por # de antecedentes insuficientes

Reglas eliminadas por confianza menor a la requerida

Se aplicaron las reglas no eliminadas a los datos posteriores al 2003. Los resultados fueron:

El número de registros clasificados correctamente es 2468
El número total de registros es 18318.

Este valor corresponde a un 13,473% del total de los registros.

[8] Para observar la tabla con todas las reglas ver anexos.

Reglas de asociación:

Se obtuvieron los siguientes resultados de la aplicación del algoritmo de reglas de asocación en el conjunto de datos de origen:

```
1. CLASIFICACIÓN=4.5 - 5.0 1210 ==> TPERDIDOS=0 1210    conf:(1)
2. NOM_PROGRAMA=Ingeniería Industrial CLASIFICACIÓN=4.0 - 4.5 SEXO=M
   1188 ==> TPERDIDOS=0 1167    conf:(0.98)
3. NOM_PROGRAMA=Ingeniería Electronica CLASIFICACIÓN=4.0 - 4.5 1329
   ==> TPERDIDOS=0 1304    conf:(0.98)
4. NOM_PROGRAMA=Ingeniería Industrial CLASIFICACIÓN=4.0 - 4.5 2475 ==>
   TPERDIDOS=0 2427    conf:(0.98)
5. CLASIFICACIÓN=4.0 - 4.5 SEXO=F 2126 ==> TPERDIDOS=0 2083
conf:(0.98)
6. CLASIFICACIÓN=4.0 - 4.5 5733 ==> TPERDIDOS=0 5613    conf:(0.98)
7. NOM_PROGRAMA=Ingeniería Industrial CLASIFICACIÓN=4.0 - 4.5 SEXO=F
   1287 ==> TPERDIDOS=0 1260    conf:(0.98)
8. CLASIFICACIÓN=4.0 - 4.5 SEXO=M 3607 ==> TPERDIDOS=0 3530
conf:(0.98)
9. NOM_PROGRAMA=Ingeniería Industrial TCURSADOS=16-20
   CLASIFICACIÓN=4.0 - 4.5 1372 ==> TPERDIDOS=0 1330    conf:(0.97)
10. TCURSADOS=16-20 CLASIFICACIÓN=4.0 - 4.5 SEXO=F 1223 ==>
   TPERDIDOS=0
    1184    conf:(0.97)
```

Reglas eliminadas por # de antecedentes insuficientes

Reglas eliminadas por confianza menor a la requerida

Se aplicaron las reglas no eliminadas a los datos posteriores al 2003. Los resultados fueron:

El número de registros clasificados correctamente es 3728
El número total de registros es 18318.

Este valor corresponde a un 20,351% del total de los registros.

- ***Consulta #2***

Árboles de decisión:

Los resultados del algoritmo aplicado a los datos del 2003 en adelante que superan los valores para los parámetros de confianza y soporte establecidos son:

Total	*# Clasificadas*	*# Mal Clasificadas*	*Soporte*	*Confianza*	*Entra*
5902	748	51	11,80955608	0,931818182	1
5902	698	47	11,03015927	0,932664756	1
5902	115	6	1,846831583	0,947826087	0
5902	410	0	6,946797696	1	1

Las reglas que corresponden a estos resultados son:

```
1. CLASIFICACIÓN=3.0 - 3.5 SEXO=F ==> C (748.0/51.0)
2. CLASIFICACIÓN=3.0 - 3.5 SEXO=M TPERDIDOS=1-5 ==> C (698.0/47.0)
3. CLASIFICACIÓN=3.0 - 3.5 SEXO=M TPERDIDOS=11-15 ==> C (115.0/6.0)
4. CLASIFICACIÓN=4.0 - 4.5 ==> S (410.0)
```

Reglas eliminadas por # de antecedentes insuficientes

Reglas eliminadas por confianza menor a la requerida

Se aplicaron las reglas no eliminadas a los datos posteriores al 2003. Los resultados fueron:

El número de registros clasificados correctamente es 257
El número total de registros es 2574.

Este valor corresponde a un 9,984% del total de los registros.

Reglas de asociación:

Se obtuvieron los siguientes resultados de la aplicación del algoritmo de reglas de asocación en el conjunto de datos de origen:

```
1. CLASIFICACIÓN=4.0 - 4.5 410 ==> estadomatricula=S 410    conf:(1)
2. tperdidos=0 CLASIFICACIÓN=4.0 - 4.5 383 ==> estadomatricula=S 383
   conf:(1)
3. CLASIFICACIÓN=4.0 - 4.5 410 ==> tperdidos=0 estadomatricula=S 383
   conf:(0.93)
4. CLASIFICACIÓN=4.0 - 4.5 estadomatricula=S 410 ==> tperdidos=0 383
   conf:(0.93)
5. CLASIFICACIÓN=4.0 - 4.5 410 ==> tperdidos=0 383    conf:(0.93)
6. tcursados=1-5 CLASIFICACIÓN=3.5 - 4.0 425 ==> tperdidos=0 390
   conf:(0.92)
7. tcursados=1-5 estadomatricula=S 435 ==> tperdidos=0 397
   conf:(0.91)
```

Reglas eliminadas por # de antecedentes insuficientes

Reglas eliminadas por confianza menor a la requerida

Se aplicaron las reglas no eliminadas a los datos posteriores al 2003. Los resultados fueron:

El número de registros clasificados correctamente es 322
El número total de registros es 2574.

Este valor corresponde a un 12,509% del total de los registros.

- ***Consulta #3***

Árboles de decisión:

Los resultados del algoritmo aplicado a los datos aleatorios tomados para poder ser predecidos, no superaron el mínimo de confianza y soporte establecidos para la investigación, por lo tanto esta técnica con estos datos no retornó ninguna regla.

Reglas de asociación:

Se obtuvieron los siguientes resultados de la aplicación del algoritmo de reglas de asociación en el conjunto de datos de origen:

```
1. id_depto=P 67 ==> nom_facultad=INGENIERIA 67    conf:(1)
 2. id_depto=F 47 ==> nom_facultad=CIENCIAS 47    conf:(1)
 3. sexo=M id_depto=P 44 ==> nom_facultad=INGENIERIA 44
conf:(1)
 4. id_depto=I display_name=Alta carga académica 38 ==>
nom_facultad=INGENIERIA 38    conf:(1)
 5. id_depto=I display_name=Bajo rendimiento 35 ==>
nom_facultad=INGENIERIA 35    conf:(1)
 6. sexo=F id_depto=I 89 ==> nom_facultad=INGENIERIA 86
conf:(0.97)
 7. id_depto=E 42 ==> nom_facultad=INGENIERIA 40
conf:(0.95)
 8. id_depto=I 185 ==> nom_facultad=INGENIERIA 174
conf:(0.94)
 9. sexo=M id_depto=I 96 ==> nom_facultad=INGENIERIA 88
conf:(0.92)
```

Reglas eliminadas por # de antecedentes insuficientes

Reglas eliminadas por confianza menor a la requerida

Se aplicaron las reglas no eliminadas a los datos posteriores al 2003. Los resultados fueron:

El número de registros clasificados correctamente es 55
El número total de registros es 339.

Este valor corresponde a un 16,224% del total de los registros.

3.5. PRUEBA DE ESCALABILIDAD

- ***Consulta #1***

Árboles de decisión:

Al realizar la prueba de escalabilidad se encontró que ninguna de las reglas que arrojó la ejecución del algoritmo superan el porcentaje de confianza (93%) y soporte (5%) establecidos para la investigación como mínimo para considerar válida una regla. Por la tanto no existen reglas de árboles para escalabilidad.

Reglas de asociación:

Se realizó el proceso de Minería de Datos para el 10% de los datos de origen, que resultó en el siguiente conjunto de reglas:

```
1. CLASIFICACIÓN=4.5 - 5.0 123 ==> TPERDIDOS=0 123    conf:(1)
2. NOM_PROGRAMA=Ingeniería Electronica CLASIFICACIÓN=4.0 - 4.5 111
   ==>       TPERDIDOS=0 111    conf:(1)
3. NOM_PROGRAMA=Ingeniería Industrial CLASIFICACIÓN=4.0 - 4.5
   SEXO=F 130 ==>    TPERDIDOS=0 127    conf:(0.98)
4. CLASIFICACIÓN=4.0 - 4.5 535 ==> TPERDIDOS=0 520    conf:(0.97)
5. NOM_PROGRAMA=Ingeniería Industrial CLASIFICACIÓN=4.0 - 4.5 247
   ==>    TPERDIDOS=0 240    conf:(0.97)
6. NOM_PROGRAMA=Ingeniería Industrial CLASIFICACIÓN=4.0 - 4.5
   SEXO=M 117 ==> TPERDIDOS=0 113    conf:(0.97)
7. TCURSADOS=16-20 CLASIFICACIÓN=4.0 - 4.5 SEXO=F 113 ==>
   TPERDIDOS=0 107    conf:(0.95)
8. NOM_PROGRAMA=Ingeniería Industrial TCURSADOS=16-20
   CLASIFICACIÓN=4.0 - 4.5 128 ==> TPERDIDOS=0 121    conf:(0.95)
```

Reglas eliminadas por # de antecedentes insuficientes

Reglas eliminadas por confianza menor a la requerida

Se aplicaron las reglas no eliminadas a los datos posteriores al 2003. Los resultados fueron:

El número de registros clasificados correctamente es 2877
El número total de registros es 18318.

Este valor corresponde a un 15,705% del total de los registros.
Ra = 2877

Para la prueba del 100% de los datos se tomarán los resultados de la prueba de precisión en la clasificación.

El número de registros clasificados correctamente es 3728
El número total de registros es 18318.

Este valor corresponde a un 20,351% del total de los registros.
Rb = 3728

Como se planteó en la metodología de pruebas se aplicará la fórmula que dará como resultado el índice de escalabilidad de la técnica:

$$Rtécnica = | Rb - Ra |$$

$$Rtécnica = | 3728 - 2877 | = 851$$

Rtécnica = 851
4,645%

- ***Consulta #2***

Árboles de decisión:

Al realizar la prueba de escalabilidad se encontró que ninguna de las reglas que arrojó la ejecución del algoritmo superan el porcentaje de confianza (93%) y soporte (5%) establecidos para la investigación como mínimo para considerar válida una regla. Por la tanto no existen reglas de árboles para escalabilidad.

Reglas de asociación:

Se realizó el proceso de Minería de Datos para el 10% de los datos de origen, que resultó en el siguiente conjunto de reglas:

```
1. CLASIFICACIÓN=4.0 - 4.5 56 ==> estadomatricula=S 56    conf:(1)
2. tperdidos=0 CLASIFICACIÓN=4.0 - 4.5 51 ==> estadomatricula=S 51
conf:(1)
3. tcursados=16-20 CLASIFICACIÓN=4.0 - 4.5 34 ==> estadomatricula=S
   34    conf:(1)
4. tcursados=16-20 tperdidos=0 CLASIFICACIÓN=4.0 - 4.5 32 ==>
   estadomatricula=S 32    conf:(1)
5. sexo=M CLASIFICACIÓN=4.0 - 4.5 30 ==> estadomatricula=S 30
conf:(1)
6. tcursados=1-5 CLASIFICACIÓN=3.5 - 4.0 41 ==> tperdidos=0 39
conf:(0.95)
```

```
7. tcursados=1-5 estadomatricula=S 55 ==> tperdidos=0 52
conf:(0.95)
8. tcursados=1-5 CLASIFICACIÓN=3.5 - 4.0 estadomatricula=S 36 ==>
   tperdidos=0 34    conf:(0.94)
9. tcursados=16-20 CLASIFICACIÓN=4.0 - 4.5 34 ==> tperdidos=0
   estadomatricula=S 32    conf:(0.94)
10. tcursados=16-20 CLASIFICACIÓN=4.0 - 4.5 estadomatricula=S 34 ==>
   tperdidos=0 32    conf:(0.94)
```

Reglas eliminadas por # de antecedentes insuficientes
Reglas eliminadas por confianza menor a la requerida

Se aplicaron las reglas no eliminadas a los datos posteriores al 2003. Los resultados fueron:

El número de registros clasificados correctamente es 337
El número total de registros es 2574.

Este valor corresponde a un 13,092% del total de los registros.
Ra = 337

Para la prueba del 100% de los datos se tomarán los resultados de la prueba de precisión en la clasificación.

El número de registros clasificados correctamente es 174
El número total de registros es 2574.

Este valor corresponde a un 6,759% del total de los registros.
Rb = 174

Como se planteó en la metodología de pruebas se aplicará la fórmula que dará como resultado el índice de escalabilidad de la técnica:

$$Rtécnica = | Rb - Ra |$$

$$Rtécnica = | 337 - 174 | = 163$$

Rtécnica = 163
6,332%

- ***Consulta #3***

Árboles de decisión:

Al realizar la prueba de escalabilidad se encontró que ninguna de las reglas que arrojó la ejecución del algoritmo superan el porcentaje de confianza (93%) y soporte (5%) establecidos para la investigación como mínimo para considerar válida una regla. Por la tanto no existen reglas de árboles para escalabilidad.

Reglas de asociación:

Se realizó el proceso de Minería de Datos para el 10% de los datos de origen, que resultó en el siguiente conjunto de reglas:

```
1. id_depto=I 22 ==> nom_facultad=INGENIERIA 22    conf:(1)
 2. sexo=F id_depto=I 14 ==> nom_facultad=INGENIERIA 14
conf:(1)
 3. sexo=M id_depto=I 8 ==> nom_facultad=INGENIERIA 8
conf:(1)
 4. id_depto=I display_name=Alta carga académica 7 ==>
nom_facultad=INGENIERIA 7    conf:(1)
 5. id_depto=F 7 ==> nom_facultad=CIENCIAS 7    conf:(1)
 6. sexo=F id_depto=I display_name=Alta carga académica 6 ==>
nom_facultad=INGENIERIA 6    conf:(1)
 7. sexo=F nom_facultad=INGENIERIA display_name=Alta carga
académica 6 ==> id_depto=I 6    conf:(1)
 8. sexo=M id_depto=F 6 ==> nom_facultad=CIENCIAS 6
conf:(1)
 9. sexo=M display_name=Problemas personales 5 ==>
nom_facultad=INGENIERIA 5    conf:(1)
10. sexo=F display_name=Bajo rendimiento 5 ==>
nom_facultad=INGENIERIA 5    conf:(1)
```

Reglas eliminadas por # de antecedentes insuficientes

Reglas eliminadas por confianza menor a la requerida

En los resultados se puede observar que la regla 2 y la regla 3 pueden ser vistas como una sola donde el sexo no es tenido en cuenta. Esta regla tendría un solo antecedente por lo cual se eliminarían estas dos reglas.

Se aplicaron las reglas no eliminadas a los datos posteriores al 2003. Los resultados fueron:

El número de registros clasificados correctamente es 52
El número total de registros es 339.

Este valor corresponde a un 15,339% del total de los registros.
Ra = 52

Para la prueba del 100% de los datos se tomarán los resultados de la prueba de precisión en la clasificación.

El número de registros clasificados correctamente es 55
El número total de registros es 339.

Este valor corresponde a un 16,224% del total de los registros.

Como se planteó en la metodología de pruebas se aplicará la fórmula que dará como resultado el índice de escalabilidad de la técnica:

$$Rtécnica = | Rb - Ra |$$

$$Rtécnica = |52 - 55| = 2$$

$$Rtécnica = 2$$

0,589%

3.6. PRUEBA DE ROBUSTEZ

- ***Consulta #1***

Árboles de decisión:

Los resultados del algoritmo aplicado a los datos ruidosos:

Total	*# Clasificadas*	*# Mal Clasificadas*	*Soporte*	*Confianza*	*Entra*
20847	1898	110	8,576773636	0,942044257	1
20847	1832	123	8,197822229	0,932860262	1

Las reglas que corresponden a estos resultados son:

```
1. TPERDIDOS= 0 TCURSADOS=16-20 NOM_PROGRAMA=Ingeniería Electrónica
   ==> 3.5 - 4.0 (1695.0/111.0)
2. TPERDIDOS= 6-10 TCURSADOS=16-20 ==> 3.0 - 3.5 (2532.0/165.0)
```

Reglas eliminadas por # de antecedentes insuficientes
Reglas eliminadas por confianza menor a la requerida

Se aplicaron las reglas no eliminadas a los datos posteriores al 2003. Los resultados fueron:

El número de registros clasificados correctamente es 1681
El número total de registros es 18318.

Este valor corresponde a un 9.176% del total de los registros.

Reglas de asociación:

Se realizó el proceso de Minería de Datos para el conjunto de datos de origen con valores nulos, que resultó en el siguiente conjunto de reglas:

```
 1. NOM_PROGRAMA=Ingeniería Industrial CLASIFICACIÓN=4.0 - 4.5
    2085 ==> TPERDIDOS=0 1943    conf:(0.93)
```

Reglas eliminadas por # de antecedentes insuficientes
Reglas eliminadas por confianza menor a la requerida

Se aplicaron las reglas no eliminadas a los datos posteriores al 2003. Los resultados fueron:

El número de registros clasificados correctamente es 1623
El número total de registros es 18318.

Este valor corresponde a un 8,860% del total de los registros.
Ra = 1623

Para la prueba del 100% de los datos se tomarán los resultados de la prueba de precisión en la clasificación.

El número de registros clasificados correctamente es 3728
El número total de registros es 18318.

Este valor corresponde a un 20,351% del total de los registros.
Rb = 3728

Rtécnica = | Rb – Ra |

Rtécnica = | 3728 - 1623 | = 2105

Rtécnica = 2105
11.491%

- ***Consulta #2***

Árboles de decisión:

Los resultados del algoritmo aplicado a los datos del 2003 en adelante que superan los valores para los parámetros de confianza y soporte establecidos son:

Total :	*# Clasificadas*	*# Mal Clasificadas*	*Soporte*	*Confianza*	*Entra*
3328	211	14	5,919471154	0,933649289	1

Las reglas que corresponden a estos resultados son:

```
1. CLASIFICACIÓN = 4.0 - 4.5: S (211.0/14.0)
```

Reglas eliminadas por # de antecedentes insuficientes
Reglas eliminadas por confianza menor a la requerida

Por tratarse de una regla con un solo antecedente, no se considera como válida para ser aplicada en la prueba, por tanto, el resultado de esta prueba es que se encontró que ninguna de las reglas que arrojó la ejecución del algoritmo superan el porcentaje de confianza (93%) y soporte (5%) establecidos para la investigación como mínimo para considerar válida una regla. Por la tanto no existen reglas de árboles para robustez.

Reglas de asociación:

Se realizó el proceso de Minería de Datos para el 10% de los datos de origen, que resultó en el siguiente conjunto de reglas:

```
1. CLASIFICACIÓN=4.0 - 4.5 211 ==> estadomatricula=S 197
conf:(0.93)
2. tperdidos=0 CLASIFICACIÓN=4.0 - 4.5 180 ==> estadomatricula=S
   167    conf:(0.93)
```

Reglas eliminadas por # de antecedentes insuficientes
Reglas eliminadas por confianza menor a la requerida

Se aplicaron las reglas no eliminadas a los datos posteriores al 2003. Los resultados fueron:

El número de registros clasificados correctamente es 174
El número total de registros es 2574.

Este valor corresponde a un 6,759% del total de los registros.
Ra = 174

Para la prueba del 100% de los datos se tomarán los resultados de la prueba de precisión en la clasificación.

El número de registros clasificados correctamente es 322
El número total de registros es 2574.

Este valor corresponde a un 12,509% del total de los registros.

Rtécnica = | Rb – Ra |

Rtécnica = | 322 - 174 | = 148

Rtécnica = 148
5,749%

- ***Consulta #3***

Arboles de decisión:

Los resultados del algoritmo aplicados a los registros ruidosos, no generaron ninguna regla. Por lo tanto no existen reglas con las cuales poder validar los datos futuros.

Reglas de asociación:

Se realizó el proceso de Minería de Datos para el conjunto de datos de origen con valores nulos, que resultó en el siguiente conjunto de reglas:

```
1. ID_DEPTO=P 67 ==> NOM_FACULTAD=INGENIERIA 67    conf:(1)
 2. ID_DEPTO=F 47 ==> NOM_FACULTAD=CIENCIAS 47    conf:(1)
 3. SEXO=M ID_DEPTO=P 44 ==> NOM_FACULTAD=INGENIERIA 44
conf:(1)
 4. ID_DEPTO=I DISPLAY_NAME=Bajo rendimiento 35 ==>
NOM_FACULTAD=INGENIERIA 35    conf:(1)
 5. SEXO=F ID_DEPTO=I 89 ==> NOM_FACULTAD=INGENIERIA 86
conf:(0.97)
 6. ID_DEPTO=E 42 ==> NOM_FACULTAD=INGENIERIA 40    conf:(0.95)
 7. ID_DEPTO=I 185 ==> NOM_FACULTAD=INGENIERIA 174
conf:(0.94)
 8. SEXO=M ID_DEPTO=I 96 ==> NOM_FACULTAD=INGENIERIA 88
conf:(0.92)
```

Reglas eliminadas por # de antecedentes insuficientes

Reglas eliminadas por confianza menor a la requerida

Se aplicaron las reglas no eliminadas a los datos posteriores al 2003. Los resultados fueron:

El número de registros clasificados correctamente es 50
El número total de registros es 339.

Este valor corresponde a un 14,749% del total de los registros.

Para la prueba del 100% de los datos se tomarán los resultados de la prueba de precisión en la clasificación.

El número de registros clasificados correctamente es 55
El número total de registros es 339.

Este valor corresponde a un 16,224% del total de los registros.

Rtécnica = | Rb – Ra |

Rtécnica = | 50 - 55| = 5

Rtécnica = 5
1,474%

La aplicación de las pruebas a cada uno de los algoritmos, permitió descubrir los aspectos positivos y negativos de cada uno de ellos con respecto al dominio de aplicación. Sin embargo, en este apartado, se pretende analizar estas diferencias desde un punto de vista técnico, sin tener en cuenta la aplicación que pueda tener en el dominio de aplicación estudiado.

En primera instancia, los resultados de la prueba de clasificación de datos de origen, mostraron que el algoritmo de árboles de decisión tiene mayor porcentaje de registros correctamente clasificados que el de reglas de asociación, por lo que se concluye que los modelos generados a partir de este algoritmo reflejan más fielmente la realidad a partir de la cual fueron generados para el dominio de aplicación estudiado. Esto no significa que sean más precisos a la hora de pronosticar o predecir valores para nuevos registros.

Para la clasificación de datos futuros, la prueba mostró que el algoritmo de reglas de asociación tiene una mayor precisión para realizar la clasificación de datos para los cuales se desea aplicar el modelo generado por el algoritmo en el caso del dominio de aplicación.

Para el caso de la prueba de escalabilidad, nuevamente se notó un mejor desempeño por parte del algoritmo de reglas de asociación, puesto que incluso se presentaron casos en los que le ejecución del algoritmo de árboles de decisión no arrojó ninguna regla que superara

los valores mínimos establecidos para los parámetros de soporte y confianza. Aún en los demás casos, también fue mejor el desempeño del algoritmo de reglas de asociación.

En cuanto a la prueba de robustez, se notó la existencia de un factor determinante en la diferenciación y comparación de los dos algoritmos en este aspecto y es el hecho que el algoritmo de árboles de decisión aún cuando no tiene en cuenta los registros con el valor nulo o vacío en el atributo clase (atributo a predecir), si lo hace si existe en algún otro atributo, considerándolo un valor válido para el mismo. Esto perjudica notablemente los resultados arrojados por la ejecución de dicho algoritmo, pues parte de las reglas obtenidas, incluyen la condición que uno de los atributos este vacío, lo cual no aportaría información en la conformación de esta regla y por tanto la hace inválida.

4. CONCLUSIONES

Se alcanzaron las siguientes conclusiones como resultado del proyecto de investigación. Dichas conclusiones son susceptibles de aplicación en el dominio de aplicación estudiado y no es posible ni recomendable su generalización.

- Después de realizar las pruebas de comparación de las técnicas de minería, se puede concluir que las dos tienen ventajas y desventajas con respecto al domino de aplicación. En este caso se pudo observar que el algoritmo de árboles de decisión tiene una mejor capacidad para la clasificación de los datos de origen en comparación al algoritmo de reglas de asociación. En el caso de estudio, una persona perteneciente al entorno académico puede valorar este resultado ya que podría clasificar un estudiante del cual no se conoce información histórica (estudiantes de primeros semestres) en una rama específica del árbol, con el fin de conocer el estado del estudiante.

  Para el caso de velocidad de ejecución, el algoritmo de reglas de asociación tiene una ventaja sobre el de árboles de decisión ya que se pudo observar que el tiempo es menor. Para este criterio, el usuario del domino no encontraría tan relevante este aspecto, ya que el proceso de Minería de Datos se ejecutaría con poca frecuencia, lo que demostraría que no es tan necesario para este dominio la velocidad de respuesta del algoritmo, aunque este criterio no se debe dejar de lado a la hora de comparar.

  En la prueba de predicción de datos futuros, se pudo observar que el algoritmo de reglas de asociación tiene una mejor precisión que el de árboles de decisión. Este criterio es uno de los más importantes a la hora de evaluar el desempeño de cada técnica, el algoritmo de reglas de asociación es más rápido y preciso que árboles de decisión para el dominio de aplicación estudiado. Para un usuario el poder predecir el estado académico de un estudiante después de que ha cursado algunos semestres o según como ha sido generado el modelo en base a años anteriores con otros estudiantes, es de gran relevancia, ya que se podrían tomar

decisiones para obtener cambios o descubrir las tendencias que tiene los estudiantes para que lleguen a tener un estado académico específico.

En la prueba de escalabilidad se pudo observar la diferencia que existe entre el modelo generado por el algoritmo de reglas de asociación y el de árboles de decisión. En este caso reglas fue más preciso a la hora de predecir los datos futuros que árboles de decisión con el modelo generado por el 10% de los datos, de lo que se puede concluir que reglas tiene una ventaja clara sobre árboles en este criterio. Para un usuario del ámbito académico, el criterio de escalabilidad puede ser considerado de vital importancia de acuerdo a la cantidad de datos que se manejen a la hora de ejecutar el proceso de Minería de datos. Se puede notar que reglas supera en el criterio de escalabilidad, lo cual es importante para la elección de la técnica que se va a usar para el entorno de aplicación.

Finalmente el criterio de robustez debe ser considerado, ya que en el caso del entorno académico se encontró que gran parte de los registros a los que se aplicó la Minería de Datos estaban ruidosos. En el momento de aplicar los algoritmos, se pudo observar que el de reglas de asociación obtuvo mejores resultados, ya que el algoritmo de árboles de decisión toma los valores nulos y los cuenta como atributo para relacionar en el momento de generar su modelo. Se puede concluir que para usuario final es mayor utilidad obtener reglas que no consideren los atributos nulos que se encuentran en los registros, por lo que reglas tiene ventaja a la hora de la elección.

- Dentro del entorno de aplicación utilizado para la realización de esta investigación, el uso de la Minería de Datos implica la obtención de varios beneficios y ventajas para los interesados. Las ventajas van más allá de los resultados o reglas obtenidas por la aplicación de los algoritmos. Tienen que ver con la aplicación que de este conjunto de reglas obtenidas se haga por parte del usuario final.

  En el caso de esta investigación, se puede considerar las consultas generadas como un resultado secundario de la misma y que pueden demostrar a las personas interesadas, de que manera pueden ser utilizados estos resultados para su beneficio.

La primera consulta se refiere a los diferentes factores o atributos de un estudiante que pueden determinar su nivel académico, reflejado en su promedio acumulado ponderado. Los beneficios de la aplicación de los resultados obtenidos son claros, pues estos permitirían predecir con un cierto grado de precisión, el estado académico de una persona de acuerdo a su información personal, como la carrera que cursa, su sexo y el rendimiento en el último semestre de acuerdo a créditos cursados y a créditos perdidos.

La segunda consulta se refiere al estado académico de una persona reflejado en si se encuentra en estado de matrícula académica o no. Nuevamente, los beneficios de la aplicación de los resultados de esta consulta pueden ser muchos. Se pueden realizar predicciones acerca de si una persona se encontrará en matrícula académica de acuerdo a su información personal y académica, como la descrita anteriormente.

La tercera consulta se refiere a los factores que influyen de mayor manera en la decisión de retiro de una materia por parte de un estudiante. Se tiene en cuenta el sexo, el departamento y facultad a los que pertenece el estudiante para predecir el atributo que identifica la causa dada por el estudiante para retirar una materia. De esta manera se podría conocer y determinar fácilmente si existe una causa frecuente para una materia y de esta manera poder plantear posibles soluciones.

- Las herramientas de Minería de Datos ofrecen información del dominio al cual son aplicadas, para lo cual se requiere que los resultados obtenidos del proceso de Minería de Datos sean útiles para el dominio de aplicación, en base a los datos con los que se cuenta. A partir de los modelos generados se procederá a la toma de decisiones, las cuales deben estar basadas en algo concreto y preciso. Los datos que se tienen son claves a la hora de realizar el proceso de minería, para que se obtenga información relevante se deben tener la mayor cantidad de atributos posibles y la mayor cantidad de registros posibles, con el fin de encontrar alguna relación entre los datos y así generar un modelo donde se puedan identificar las relaciones que existen entre ellos.

  Otro aspecto clave en el momento de la aplicación de la Minería de Datos, es definir los parámetros de soporte y confianza con los que se quieren los resultados. Esto lo debe definir el usuario de acuerdo

a las necesidades que encuentre para que se cumplan los objetivos propuestos antes de ser aplicado el proceso. Con estos valores se tendrá una mayor o menor precisión en la generación del modelo, que al final es en lo que se basará el usuario para la toma de decisiones.

También se debe tener en cuenta que el proceso de Minería de Datos es un proceso costoso en términos de hardware. Se necesitan equipos de alto desempeño para que el proceso de minería cumpla con las expectativas de velocidad y rendimiento. Para el caso de estudio se trabajó con los estudiantes de la Universidad Javeriana de la facultad de ingeniería desde el año 1998, se contó con un total de 20847 para el caso de la consulta 1, lo que no afectó de una manera notoria la realización del proceso en cuestión de velocidad ya que se contaba con un dominio relativamente pequeño.

- A partir de la investigación y de la realización de las pruebas, se concluyó que para que cualquiera de los dos algoritmos de Minería de Datos tenga resultados útiles, se requiere que las bases de datos que sirven como origen para la información a utilizar, cumplan ciertas condiciones. Entre ellas se encuentran las siguientes:

  La prueba de robustez, mostró claramente que el conjunto de datos de origen debe proveer información con un cierto grado de completitud. Es decir, en lo posible debe evitarse la aparición de registros con información faltante, pues este tipo de datos tiene una incidencia negativa en el desempeño de los algoritmos estudiados, especialmente para el caso del algoritmo de árboles de decisión.

  Por otra parte, el conjunto de datos de origen seleccionado debe tener un volumen importante para que los resultados obtenidos sean válidamente aplicables al dominio. Este volumen es determinado de acuerdo a la cantidad de información que conformaría el universo de datos del dominio de aplicación. Las personas interesadas en la aplicación de un proceso de Minería de Datos deberían posteriormente determinar un porcentaje de este universo como conjunto de información mínimo para la realización del proceso.

- Dentro de lo realizado en este proyecto de investigación, se pudo detectar que una ventaja clara del algoritmo de árboles de decisión

sobre el de reglas de asociación, es el hecho de poder seleccionar el atributo clase o atributo a predecir. De esta manera se evita la generación de reglas que pueden llegar a ser superfluas o que no resulten de interés por predecir otros atributos distintos al que interesa. Por otra parte, esta misma ventaja puede ser una desventaja en cuanto la Minería de Datos tiene por objeto el lograr encontrar reglas y asociaciones entre atributos que hasta ese momento eran desconocidas. Por esta misma razón de ser, el hecho de sesgar la búsqueda de reglas a un solo atributo puede así mismo sesgar los resultados de un proyecto de Minería de Datos. De cualquier forma, se considera que para el dominio de aplicación estudiado, esta generación de reglas prediciendo otros atributos distintos al atributo clase es una desventaja para el algoritmo de reglas de asociación.

- Para finalizar, en el desarrollo de las pruebas se pudo observar que el algoritmo de árboles de decisión es superior al de reglas de asociación si se trata de la clasificación de los datos para el dominio de aplicación de estudio. Después de realizar todas las pruebas de acuerdo a los criterios determinados, se pudo notar que el algoritmo de reglas de asociación cumple de una manera más completa con las expectativas que tiene un usuario en el ámbito académico. La velocidad, la precisión en la predicción, la escalabilidad y la robustez de las reglas de asociación tuvieron una ventaja clara en los resultados comparado con los resultados que se obtuvieron con árboles de decisión. Estos criterios de comparación son un punto clave en el momento de tomar la decisión de elegir una técnica o la otra, se mostraron las ventajas y desventajas de las dos técnicas seleccionadas, la elección dependería de lo que el usuario final desee lograr.

5. BIBLIOGRAFÍA

- Pyle Dorian. Data Preparation for Data Mining
  Morgan Kaufmann Publishers – 1999

- Date C.J., Introducción a los Sistemas de Bases de Datos
  Addison Wesley Longman Iberoamericana – 1998 (Quinta Edición)

- Ian H. Witten and Eibe Frank. "Data Mining: Practical machine learning tools with Java implementations,"
  Morgan Kaufmann, San Francisco – 2000

- Winston Patrick Henry. Inteligencia Artificial
  Addison Wesley Iberoamericana – Tercera Edición

- Han Jiawei, Kamber Micheline. Data Mining Concepts and Techniques.
  Morgan Kaufmann, San Francisco – 2001

- Mandenhall William, Scheaffer Richard. Estadística matemática con aplicaciones.
  Editorial Iberoamericana, S.A. – 1986.

- Barrera Garavito Andrés Alexander, Garzón Limberg Camilo, Ríos Ospina Mario Fabián. Proyecto de Investigación: "Herramienta de gestión y administración del ciclo de vida del estudiante de la Pontificia Universidad Javeriana, caso de estudio: Ingeniería de Sistemas"
  Noviembre 24 de 2003.

- Rakesh Agrawal. Mining Association Rules between Large Itemsets. Proceedings of the 1993 ACM Sigmod.
  Washington 1993.

- Usama Fayyad. Mining Database: Towards Algorithms for knowledge discovery.
  Microsoft Research – 1998.

- Harjinder S. Gill. Data Warehousing: La integración de información para la major toma de decisions.
  Prentice-Hall – México 1996.

- P. Yu. M. Chen, J. Han. Data Mining: An overview from a database perspective.
  Prentice-Hall – 1996.

- Rantzau Ralpf. Extended Concepts for Association Rule Discovery.
  Universit¨at Stuttgart - 1997

- R. Srikant & R. Agrawal, "Mining Generalized Association Rules," In Proceedings of the International Conference of Very Large Databases, September 1994.

- http://dns1.mor.itesm.mx/~emorales/Cursos/KDD/,
  Abril 13 de 2004.

- http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/MineriaDatosBressan.htm#Qué%20es%20Data%20Warehousing
  Junio 10 de 2004.

6. ANEXOS

6.1. HERRAMIENTA WEKA

WEKA es una colección de algoritmos de aprendizaje para tareas de Minería de Datos. Se trata de una herramienta totalmente desarrollada en el lenguaje de programación JAVA, que contiene una interfaz gráfica para la aplicación de los algoritmos directamente al conjunto de datos de origen, así como la posibilidad de aplicar los algoritmos desde un programa JAVA independiente.

Contiene herramientas para el pre-procesamiento de los datos, clasificación, regresión, clustering, reglas de asociación y visualización. Así mismo, facilita la creación de nuevos esquemas de aprendizaje mediante la utilización de los algoritmos ya implementados.

WEKA es software de código abierto y funciona bajo el esquema de GNU (General Public License). Fue creado en el Departamento de Ciencias de la Computación de la Universidad de Waikato, en Nueva Zelanda.

6.1.1. Características de la herramienta WEKA

- Interacción con datos de origen: como se pudo observar en la sección de ingreso de datos a la herramienta WEKA, la interacción con ella puede ser a través de diferentes medios. Puede recibir los datos de origen directamente desde una base de datos, utilizando protocolo JDBC, el cual fue el método utilizado para la elaboración de las pruebas de este proyecto.

  También puede interactuar con archivos con formato ARFF[9], lo cual es beneficioso para su desempeño, ya que una vez realizada una consulta a la base de datos, los resultados pueden ser almacenados en un archivo ARFF para después realizar el proceso de minería sobre este.

[9] Archivos planos con formato para identificar todos los atributos de un conjunto de datos, así como todos sus posibles valores.

- Interfaz gráfica: provee herramientas de visualización, preprocesamiento y métodos para la aplicación de los algoritmos de Minería de Datos.

- Preprocesamiento: son llamados "filtros" y permiten realizar actividades como discretización, normalización, selección de atributos, transformación y combinación de los mismos.

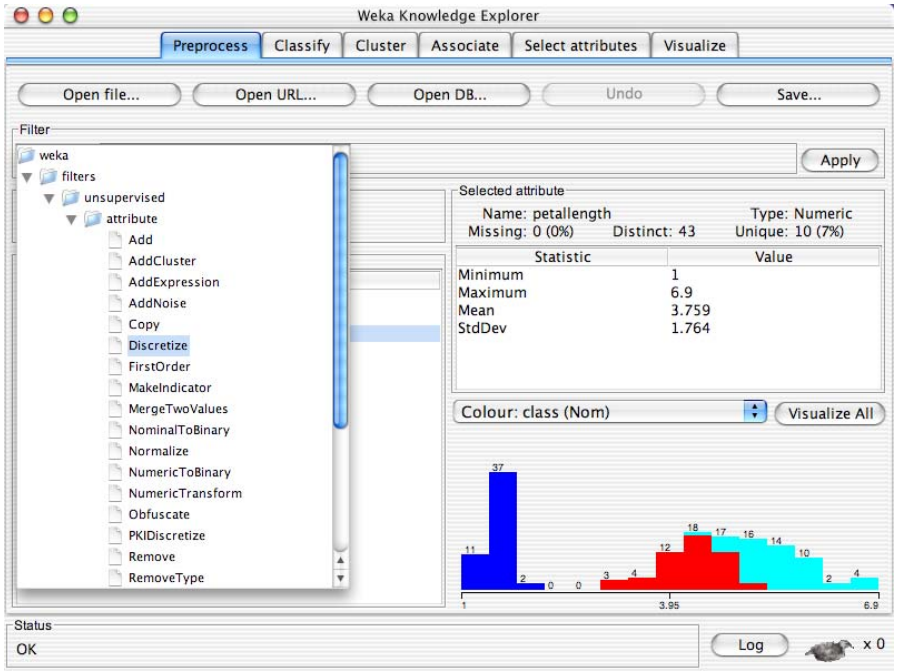


Ilustración 18 – Preprocesamiento de datos en WEKA

En la figura anterior se muestra la selección de uno de los filtros disponibles (en este caso de discretización) para aplicar en los datos cargados en memoria que se visualizan en la parte derecha.

- Algoritmos de clasificación: en WEKA, son los llamados "classifiers" que implementan una gran variedad de algoritmos de Minería de Datos. En el caso de esta investigación, se realiza el estudio sobre el algoritmo de árboles de decisión (j48) y de reglas de asociación.

  En la siguiente figura se muestra la selección del algoritmo de árboles de decisión j48 para su aplicación sobre los datos ya cargados en memoria.

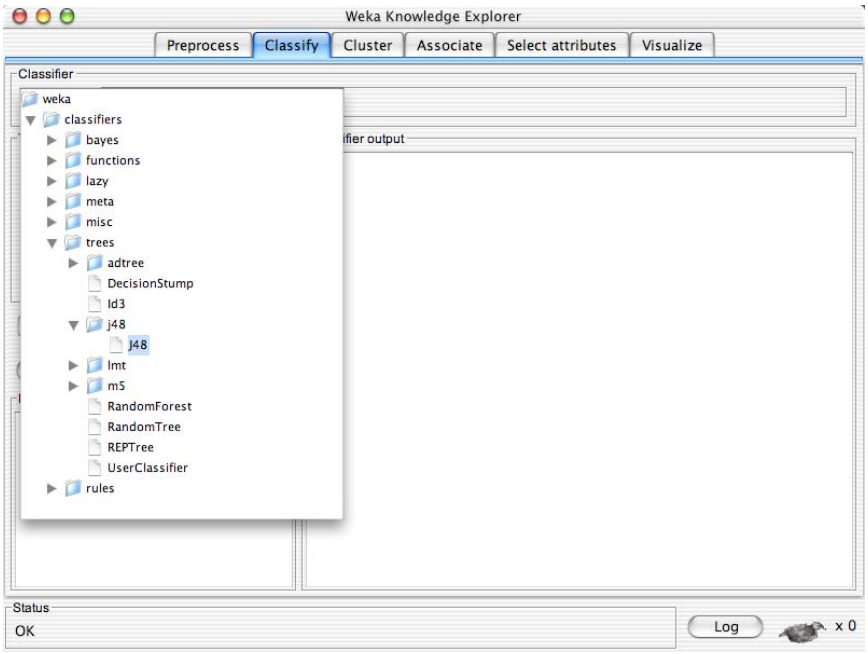


Ilustración 19 – Algoritmos de clasificación en WEKA

- Aplicación de los algoritmos de clasificación: en la siguiente figura se puede ver la manera en que se aplica el algoritmo seleccionado y como se visualizan los resultados de dicha aplicación en el panel derecho de la pantalla.

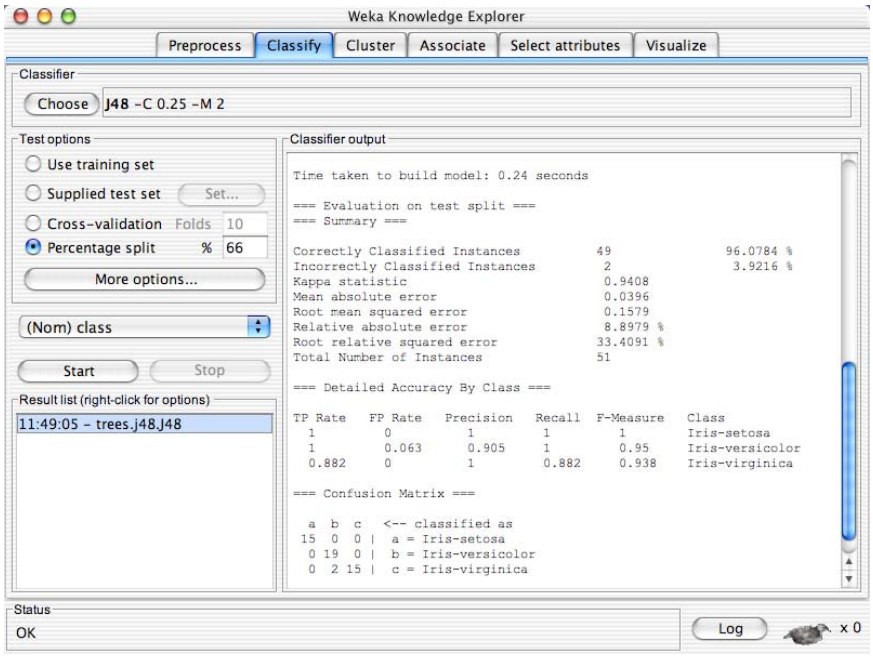


Ilustración 20 – Aplicación de algoritmos de clasificación en WEKA

- Aplicación de los algoritmos de asociación: WEKA provee el algoritmo a-priori de generación de reglas de asociación. Este algoritmo solo acepta datos discretos. Esta fue una de las razones para realizar el proceso de discretización de datos en la investigación. En la siguiente figura se puede ver la manera en que se aplica el algoritmo seleccionado y como se visualizan los resultados de dicha aplicación en el panel derecho de la pantalla.

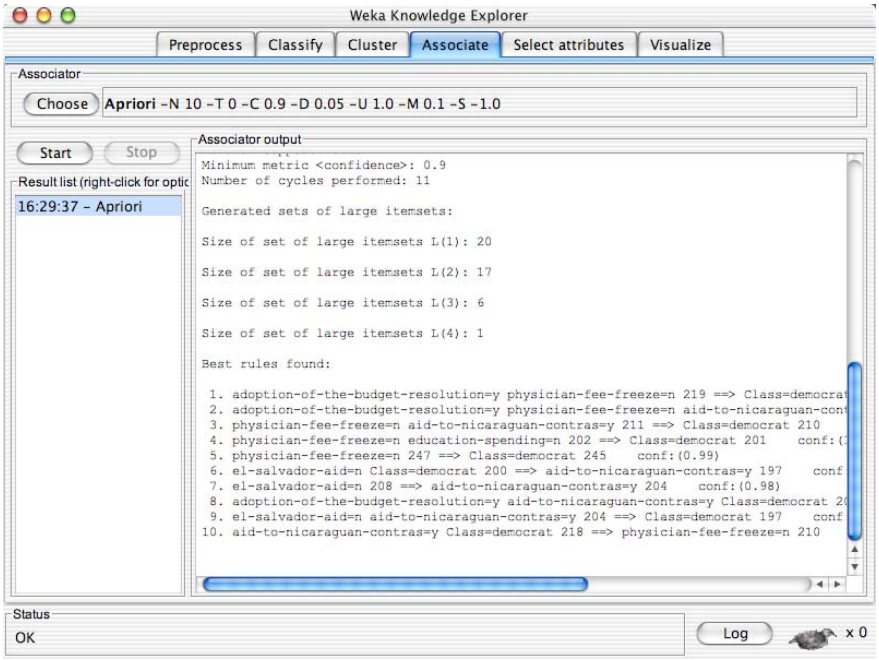


Ilustración 21 – Aplicación de algoritmos de asociación en WEKA

- WEKA proporciona una herramienta para realizar experimentos que permitan comparar el desempeño de varios algoritmos de Minería de Datos. Esta herramienta hubiera sido de gran utilidad para la investigación, sin embargo, esta herramienta llamada "Experimenter", solo puede ser aplicada a algoritmos de clasificación, pero no entre algoritmos de clasificación y de asociación, como pretende hacerlo este proyecto de investigación.

6.1.2. Razones para seleccionar WEKA

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

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

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

6.2. PRODUCTOS ADICIONALES

Con el fin de poder realizar las pruebas planteadas en la metodología del proyecto, fue necesario desarrollar herramientas de software auxiliares, para varias tareas. Estas herramientas, pueden ser aplicadas a cualquier proyecto de Minería de Datos, con el fin por ejemplo, de validar factores como el cubrimiento de las reglas generadas.

A continuación se describen dichas herramientas y se explica de qué manera fueron aplicadas en este proyecto de investigación.

6.2.1. Herramienta de validación de reglas

El objetivo de esta herramienta es obtener el número de instancias que son correctamente clasificadas por un conjunto de reglas obtenido a partir de un proceso de Minería de Datos. Estas reglas le son proporcionadas al programa en su código fuente, aunque para quienes deseen realizar una implementación más genérica de esta herramienta, se podría realizar una interfaz con el usuario que permita el ingreso de las reglas de forma dinámica.

Por esta razón, existe un programa para cada conjunto de reglas generado. El código fuente de esta herramienta, para el caso de la consulta número 1, es el siguiente:

```
1  public void predecirReglas() {
2    Connection con = getConnection();
3    Statement st = null;
4    ResultSet rs = null;
5    int registrosCorrectos = 0;
6    try {
7     String sql = "SELECT * FROM " + TABLE_NAME;
8     st = con.createStatement();
9     rs = st.executeQuery(sql);
10    boolean band;
11    while (rs.next()) {
12     band = false;
13     NOM_PROGRAMA = rs.getString("NOM_PROGRAMA");
14     CLASIFICACION = rs.getString("CLASIFICACION");
15     TPERDIDOS = rs.getString("TPERDIDOS");
16     SEXO = rs.getString("SEXO");
17     TCURSADOS = rs.getString("TCURSADOS");
       /*  1. NOM_PROGRAMA=Ingeniería Electronica CLASIFICACION=4.0 – 4.5 2455 ==> TPERDIDOS=0
18 2418   conf:(0.98)*/
19     if (NOM_PROGRAMA.equals("Ingeniería Electronica") &&
20        CLASIFICACION.equals("4.0 – 4.5") && TPERDIDOS.equals("0")) {
21      if (band == false)
22        registrosCorrectos++;
23      band = true;
24     }
25     /*  2. CLASIFICACION=4.0 – 4.5 SEXO=F 3504 ==> TPERDIDOS=0 3447   conf:(0.98)*/
26     if (CLASIFICACION.equals("4.0 – 4.5") && SEXO.equals("F") &&
27        TPERDIDOS.equals("0")) {
28      if (band == false)
29        registrosCorrectos++;
30      band = true;
31     }
       /*  4. NOM_PROGRAMA=Ingeniería Industrial CLASIFICACION=4.0 – 4.5 4122 ==> TPERDIDOS=0
32 4051   conf:(0.98)*/
33     if (NOM_PROGRAMA.equals("Ingeniería Industrial") &&
34        CLASIFICACION.equals("4.0 – 4.5") && TPERDIDOS.equals("0")) {
35      if (band == false)
36        registrosCorrectos++;
37      band = true;
38     }
39     /*  5. CLASIFICACION=4.0 – 4.5 SEXO=M 6000 ==> TPERDIDOS=0 5895   conf:(0.98)*/
40     if (CLASIFICACION.equals("4.0 – 4.5") && SEXO.equals("M") &&
```

```
41          TPERDIDOS.equals("0")) {
42        if (band == false)
43          registrosCorrectos++;
44        band = true;
45       }
         /*   6. NOM_PROGRAMA=Ingeniería Industrial TCURSADOS=16-20 CLASIFICACION=4.0 – 4.5 2336
46 ==> TPERDIDOS=0 2279    conf:(0.98)*/
47       if (NOM_PROGRAMA.equals("Ingeniería Industrial") &&
48          TCURSADOS.equals("16-20") && CLASIFICACION.equals("4.0 – 4.5") &&
49          TPERDIDOS.equals("0")) {
50        if (band == false)
51          registrosCorrectos++;
52        band = true;
53       }
54       /*   7. TCURSADOS=16-20 CLASIFICACION=4.0 – 4.5 5012 ==> TPERDIDOS=0 4883    conf:(0.97)*/
55       if (TCURSADOS.equals("16-20") && CLASIFICACION.equals("4.0 – 4.5") &&
56          TPERDIDOS.equals("0")) {
57        if (band == false)
58          registrosCorrectos++;
59        band = true;
60       }
         /*   8. TCURSADOS=16-20 CLASIFICACION=4.0 – 4.5 SEXO=M 2989 ==> TPERDIDOS=0 2911
61 conf:(0.97)*/
62       if (TCURSADOS.equals("16-20") && CLASIFICACION.equals("4.0 – 4.5") &&
63          SEXO.equals("M") && TPERDIDOS.equals("0")) {
64        if (band == false)
65          registrosCorrectos++;
66        band = true;
67       }
68     }
69     System.out.println(
70        "El numero de registros clasificados correctamente son = " +
71        registrosCorrectos);
72   }
73   catch (SQLException e) {
74     e.printStackTrace();
75   }
76   finally {
77     try {
78       if (rs != null) rs.close();
79       if (con != null) con.close();
80     }
81     catch (SQLException e) {}
82   }
83
84 }
```

La constante TABLE_NAME en la línea 7, tiene como valor el nombre de la tabla a partir de la cual se extraerán los datos para la validación de las reglas.

Cada uno de los condicionales (sentencias *if*), en las líneas 19, 26, 33, 47, 55, 62, representa una de las reglas del conjunto proporcionado por el algoritmo de Minería de Datos aplicado con anterioridad. La variable "band" es una bandera que permite que se cuente cada registro una sola vez (si es clasificado por alguna de las reglas).

El resultado final se obtendrá en la variable "registrosCorrectos".

INDICE

ILUSTRACIONES

TABLAS