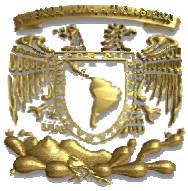


UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DIVISIÓN DE INGENIERÍA ELÉCTRICA

TESIS PARA OBTENER EL TÍTULO DE INGENIERO EN COMPUTACIÓN

ESTUDIO DE MINERÍA DE DATOS EN LA FACULTAD DE INGENIERÍA.

ALUMNOS
IBARRA GARCÍA ERNESTO PATHROS
MÁRQUEZ TELLEZ CLAUDIA EYZALLADET ROXANA

DIRECTORA DE TESIS
LIZÁRRAGA RAMÍREZ GABRIELA BETZABÉ

Ciudad Universitaria, D.F. Junio del 2009

~ a mi mamá por todo su apoyo y cariño. TQM

~ a mi papá por todo su apoyo y cariño. TQM

~ a mi hermana por todo su apoyo y cariño. TQM

~ a Bernarda.

~ a mis parientes que ya se me adelantaron en el camino … q.e.d.

~ A todas mis tías, tíos, primas, primos, …

~ Моей девушке Юлия

…и моим друзьям Анна, Ксения, Ольга.

~ À tous mes amis français.

~ Meinen Deustchen Freunden.

~ Till mina svenska och alla andra europeiska vänner.

~ A nuestra querida Facultad de Ingeniería, UNAM.

~ A Betzabé por todo su valioso apoyo.

~ Al Dr. Coria por asesorarnos sobre dónde y cómo empezar…

~ A Enrique Larios por sus comentarios y sus sugerencias valiosas.

~ A nuestros sinodales por sus comentarios, sugerencias y críticas.

~ A todos mis compañeros de trabajo de la División de Ciencias Básicas de la Facultad de Ingeniería …
(afortunadamente, es una lista enorme) …

Atte: Ernesto Pathros Ibarra García. patrotsky@yahoo.com

Dedico el presente trabajo de tesis
a las personas que me lo han dado
todo a mi madre y a mi abuelita,
quienes con su cariño y apoyo,
han hecho realidad uno de mis
mas grandes anhelos.

A mis tíos, a mi hermana y a mis primos
que de alguna u otra manera participaron
en alentarme para llevar a cabo este trabajo.

"GRACIAS"

A mi directora de tesis,
quien a puesto gran dedicación
y esfuerzo y ha compartido sus
conocimientos para que se pudiera
llevar a cabo el presente trabajo.

"GRACIAS"

A los sinodales y a todos
mis demás profesores
que colaboraron en mi
educación profesional.

"GRACIAS"

ÍNDICE

INTRODUCCIÓN

En la Facultad de Ingeniería de la UNAM se tiene la situación de que existe un gran número de alumnos que desertan o abandonan sus estudios. Para atender esta situación se han realizado diversos estudios estadísticos y se han puesto en marcha diversas medidas y acciones, algunas exitosas.

Lo que se propone en este trabajo escrito es aportar un nuevo enfoque a dichos estudios estadísticos utilizando la *minería de datos* con el fin de encontrar patrones de comportamiento de los alumnos que desertan y de aquellos que terminan todas sus materias.

En principio se buscan las materias con más alto índice de reprobación de cada de una de las carreras que se imparten en la Facultad, teniendo estos resultados, se aplican el o los algoritmos de minería de datos para que arroje las posibles causas de dicho problema

Para llevar a cabo el estudio, se contó con copias de las tablas de los historiales académicos de la base de datos de la Facultad de Ingeniería.

Se sabe que existen muchos factores externos tales como los económicos, sociales, y geográficos, que intervienen en el desempeño académico de los estudiantes los cuales no se abordan en este trabajo el cual se centra principalmente en poner de relieve el gran potencial que tiene la *minería de datos*.

La *minería de datos* es un proceso de extracción o descubrimiento de conocimiento útil y novedoso de una Base de Datos (o varias) que normalmente es enorme. Es como sumergirse en una montaña gigantesca de información en la cual existe un conocimiento que no se ha descubierto (oculto a simple vista). Conocimiento que contiene información valiosa para la toma de decisiones.

El estudio se centrará en los siguientes puntos:

- Obtención de los patrones de comportamiento de reprobación con base en los historiales académicos.
- Obtención de un algoritmo que permita predecir si un alumno tiene tendencias a desertar con base en los historiales académicos.
- Influencia del plan de estudios (seriación excesiva o falta de conocimientos previos) en el rendimiento académico.

En el primer capítulo, *minería de datos*, se presenta los conceptos básicos de la *minería de datos* así como las aplicaciones que puede tener la misma.

En el capítulo dos, *fases del KDD* (*descubrimiento de conocimiento en bases de datos*), se describe el proceso que conduce al descubrimiento de nuevo conocimiento desde las bases de datos: desde la recolección y preparación de los datos, hasta la obtención y la difusión del nuevo conocimiento.

En el capítulo tres, *algoritmos utilizados en la minería de datos*, se presentan algunos de los tantos algoritmos que existen en la minería de datos que se usaron en el presente trabajo. Estos algoritmos utilizados, fueron los que mejores resultados dieron en nuestro estudio.

En el capítulo cuatro, *descripción del desarrollo práctico de las fases con una o más herramientas de minería de datos*, se presenta la parte práctica mostrando paso a paso el procedimiento para obtener los patrones de comportamiento más relevantes que ayudan a describir las características de los alumnos que desertan y de los que terminan todas sus materias. Asimismo se muestra, a manera de propuesta, cómo generar un modelo de predicción de los cuáles alumnos desertarán y cuáles terminarán sus estudios utilizando un algoritmo de clasificación (*IBk*).

Finalmente, en el capítulo de conclusiones, se presentan sugerencias para agregar más variables en la base de datos de la Facultad de Ingeniería de tal manera de que el presente estudio de minería de datos pueda incluir más factores que seguramente intervienen en el desempeño académico.

ALCANCE DE LA TESIS

Básicamente se cuenta con los datos de los historiales académicos. Por lo que no involucramos información del tipo socio-económico ni de otro tipo de estudio.

En esta tesis se pretende ayudar a observar las tendencias de deserción y de terminación de todas las materias intentando hacer predicciones de las mismas utilizando un algoritmo de clasificación. Asimismo se desea encontrar reglas o patrones de comportamiento útil y novedoso de los alumnos que desertan y de los que sí terminan todas sus materias para intentar conocer sus características que hacen de cada alumno desertar o concluir sus estudios.

Los resultados pueden ayudar a visualizar mejor cómo se va dando la deserción o terminación de materias, generar nuevas necesidades del uso de Minería de Datos para la detección o de descubrimiento de conocimiento específico que se desea encontrar que sea también de utilidad para la Facultad de Ingeniería.

CAPÍTULO I
"MINERÍA DE DATOS"

1.1 Definición de Minería de Datos

En la actualidad las bases de datos han acumulado una gran cantidad de datos de diversa índole, en la cual la información útil no es fácil de encontrar o de inferir a simple vista. Para muchas empresas o entidades sería de mucha utilidad contar con esa información y, por lo tanto, estarían interesadas en rescatar esa información. (1)

Es por ello que se busca una forma, para que con los datos existentes, se pudieran encontrar soluciones de estadísticas o resumen del gran volumen de la información. Así se origina la Minería de Datos (MD).

La minería de datos se puede comparar en una forma similar con la minería que se realiza para encontrar minerales, ya que la segunda se dedica a escarbar en grandes volúmenes de tierra y piedra para así encontrar minerales preciosos como pueden ser diamantes y oro, por nombrar algunos; la primera realiza la misma acción, pero en lugar de encontrar minerales preciosos, se encuentra información útil y muy valiosa.

Es por ello que se puede definir a la minería de datos como:

"Un mecanismo de explotación, consistente en la búsqueda de información valiosa en grandes volúmenes de datos. Está muy ligada a las bodegas de datos que proporcionan la información histórica con la cual los algoritmos de minería de datos obtienen información necesaria para la toma de decisiones." (2)

La minería de datos se fundamenta en la intersección de diversas áreas de estudio, entre las cuales cabe destacar: análisis estadístico, bases de datos, inteligencia artificial y visualización gráfica.

Los cuales se apoyan en el empleo de algoritmos y procedimientos que se utilizan para sacar a la luz asociaciones, correlaciones, reglas, patrones e incluso excepciones interesantes o potencialmente útiles, desconocidos y escondidos en bases de datos.

La minería de datos deriva patrones y tendencias que se encuentran ocultas en los datos, los cuales se pueden recopilar y definir como un modelo de minería datos.

Asimismo se puede dividir a la minería de datos en:

- Minería de datos predicativa (mdp): usa primordialmente técnicas estadísticas.
- Minería de datos para descubrimiento de conocimiento o descriptiva (mddc): usa principalmente técnicas de inteligencia artificial. (2)

Esto se debe a que, dependiendo del objetivo del descubrimiento de conocimiento (problema) que se le dé a la Minería de Datos se escoge el tipo de ésta (describir o predecir) que mejor convenga.

Un modelo de minería de datos forma parte de un proceso mayor que incluye desde la definición del problema básico que resolverá el modelo hasta la implementación del modelo en un entorno de trabajo. (3)

La obtención y realización del modelo es un factor relevante en la minería de datos, ya que es pieza fundamental para localizar toda aquella información valiosa que proporciona la MD.

La realización del modelo de un proyecto de minería de datos sigue siempre los mismos pasos, independientemente de la técnica o tecnología que se desee o se deba utilizar, ya que en algunas ocasiones depende de la naturaleza de la información.

La creación de un modelo de minería de datos es un proceso dinámico e iterativo. Una vez que ha explorado los datos, puede que descubra que resultan insuficientes para crear los modelos de Minería de Datos adecuados y que, por tanto, debe buscar más datos. Puede generar varios modelos y descubrir que no responden al problema planteado cuando lo definió y que, por tanto, debe volver a definir el problema.

Es posible que deba actualizar los modelos una vez implementados debido a que hay más datos disponibles. Por esto es importante comprender que la creación de un modelo de minería de datos es un proceso y que, cada paso del proceso, puede repetirse tantas veces como sea necesario para crear un modelo válido. (3)

1.2 TIPOS DE DATOS

Una pregunta obligada para toda persona que esté interesada en aplicar la Minería de Datos para un fin específico o simplemente por conocimiento es: ¿a qué tipos de datos puede aplicarse la minería de datos? La respuesta en primera instancia es que ésta puede aplicarse a cualquier tipo de información, siendo las técnicas de minería diferentes para cada una de ellas; ya que éstos pueden ser simplemente texto y/o imágenes.

Para establecer una clasificación de ***tipos de datos*** se tiene que diferenciar entre datos estructurados provenientes de bases de datos relacionales, otros tipos de datos semiestructurados en bases de datos (espaciales, temporales, textuales y multimedia) y datos no estructurados provenientes de Web o de otros tipos de documentos.

Bases de datos relacionales

Una base de datos relacional es una colección de relaciones (tablas). Cada tabla consta de un conjunto de atributos (columnas y campos) y puede tener un gran número de tuplas (registros o filas). La figura 1.1 ilustra una base de datos con dos relaciones: empleados y departamentos.

Empleados

idE	Enombre	Sueldo	Puesto	Edad	Sexo	idD
1	Israel	3.500	Supervisor	23	H	Ma
2	Juan	2.400	Ténico	30	H	Ge
3	Lucía	2.900	Secretaria	25	M	Ma
4	Daniel	3.200	Supervisor	29	H	Ma
5	Ana	2.100	Secretaria	34	M	Ge

Departamentos

IdD	Dnombre
Ge	Gestión
Ve	Ventas
Ma	Marketing

Figura 1.1 Base de datos relacionales

> "Un atributo es una propiedad de un objeto y puede describirse por un valor. Es decir, alguna medición que se aplica a un objeto."(4)

Una de las principales características de las bases de datos relacionales es que los datos deben seguir una estructura y son, por lo tanto, estructurados.

Las bases de datos relacionales (recogidas o no en un almacén de datos, normalizadas o estructuradas de una manera multidimensional) son la fuente de datos para la mayoría

de las aplicaciones de Minería de Datos. Muchas de las técnicas que se utilizan en la Minería de Datos no son capaces de trabajar con toda la base de datos, sólo son capaces de tratar con una tabla a la vez; esto por medio de una consulta se puede combinar en una sola vista llamada *vista minable*, que es aquella información que se encuentra en varias tablas que son requeridas para cada tarea concreta de Minería de Datos. Por lo que, la representación tabular, conocida también como atributo-valor, es la más utilizada por las técnicas de Minería de Datos. (5)

En las bases de datos existen muchos tipos de datos (enteros, reales, fechas, texto, etc.). En las técnicas que se utilizan en la Minería de Datos, generalmente se usan los tipos numéricos y categóricos o nominales.

- Los atributos numéricos contienen valores enteros o reales (por ejemplo, edad o calificaciones).
- Los atributos categóricos o nominales toman valores en un conjunto finito y preestablecido, como el sexo (H, M), el nombre de la materia (Cálculo, Álgebra, Estadística, etc.). (5)

Otros tipos de bases de datos

- **Base de datos espaciales**: Son las que contienen información relacionada con el espacio físico (una ciudad, zonas montañosas...). En este tipo de bases de datos se incluyen datos de redes de transporte, información del tráfico, datos geográficos, etc., donde las relaciones espaciales son de alta relevancia. Por lo que la minería de datos se convierte en una herramienta magnífica para encontrar patrones entre los datos, como la planificación de nuevas líneas del metro en función a la distancia que existe entre las ya existentes.

- **Base de datos temporales**: Son las que contienen muchos atributos relacionados con el tiempo, los cuales pueden referirse a distintos instantes o intervalos temporales. Este tipo de bases de datos es muy popular en la estadística y, con la ayuda de la minería de datos, se puede encontrar las características de evolución, tendencias de cambio, etc.

- **Base de datos documentales**. Son las que contienen una colección de documentos y que su contenido es básicamente texto, que pueden ir de simples palabras clave a los resúmenes. Estas bases de datos pueden contener documentos no estructurados (como cualquier tipo de biblioteca digital), semi-estructurados (se tienen índices donde se puede extraer la información por partes) o los estructurados (la información se encuentra en fichas bibliográficas). En este tipo de datos la minería se realiza para obtener asociaciones entre los contenidos, agrupar o clasificar el texto. Para ello, los métodos de minería se integran con otras técnicas y/o métodos como es el lingüístico.
- **Base de datos multimedia**: Son las que almacenan imágenes, audio y video. Soportan objetos de gran tamaño. Para la minería de estas bases de datos es necesario que se integren los métodos de minería con técnicas de búsqueda y almacenamiento.

La World Wide Web

La World Wide Web en la actualidad es donde se almacena la mayor cantidad de información de todo tipo. Por ello en la Web hay una gran cantidad de datos, en los cuales se puede extraer conocimiento relevante y útil. Realizar una Minería de Datos en la Web no es una tarea sencilla, debido a que muchos de los datos que se encuentran en ella no son estructurados o semi-estructurados, y que las páginas Web contienen datos multimedia. Otros aspectos que dificultan este proceso son cómo determinar a qué páginas se debe acceder y cómo seleccionar la información útil para extraer el conocimiento. Por su diversidad, la Minería Web se organiza en tres categorías:

- **Mineria del contenido**, Se utiliza para encontrar patrones de los datos de las páginas Web.
- **Mineria de la estructura**, Se utiliza en Hipervínculos y *URL*s.
- **Mineria del uso**, Se utiliza para saber qué actividades realiza un usuario en las páginas Web.

1.3 TIPOS DE MODELOS

La minería de datos tiene como objetivo analizar los datos para extraer un conocimiento útil y novedoso. Dicho conocimiento puede estar representado en forma de relaciones, patrones o reglas que, a simple vista, se desconoce su existencia o bien en forma de descripción (resumen). A este tipo de relaciones o resúmenes se les denomina **modelo de datos**. Existen diferentes formas de representar los modelos y cada una de ellos determina el tipo de técnica que se requiere.

Los modelos de datos pueden ser de dos tipos: predictivos y descriptivos.

Los modelos predictivos pretenden estimar valores futuros o desconocidos de variables de gran interés que también son llamadas variables objetivo o dependientes, usando a su vez a otro tipo de variables llamadas variables independientes o predictivas. (5)

La tarea de los modelos descriptivos es identificar patrones que explican o resumen a los datos, es decir, sirven para explorar a las propiedades de los datos que son examinados.

1.4 MINERÍA DE DATOS Y EL KDD

La MD en el más amplio contexto es el Descubrimiento de Conocimiento de bases de datos (***Knowledge Discovery in Databases KDD)***. Este término se originó en el campo de la Inteligencia Artificial. El proceso de *KDD* implica varios aspectos: objetivos, selección de los datos, procesamiento de los datos, transformación de los mismos – si fuera necesario, realizar la minería de datos al modelo extraído, interpretación y evaluación del descubrimiento. (6)

El *KDD* se puede definir como "el proceso de identificar patrones válidos, novedosos, potencialmente útiles y comprensibles a partir de los datos". Y los cuales se definen de la siguiente manera:

- válido: los patrones deben ser precisos para datos nuevos, y no sólo para aquellos que han sido utilizados en su obtención.
- novedoso: tienen que aportar algo desconocido tanto para el sistema como para el usuario.
- potencialmente útil: la información obtenida debe de tener algún beneficio para el usuario. (5)

Como se puede notar, el *KDD* es un proceso complejo que incluye la obtención de los modelos o patrones, la evaluación e interpretación del mismo. *Figura. 1.2*.

KDD

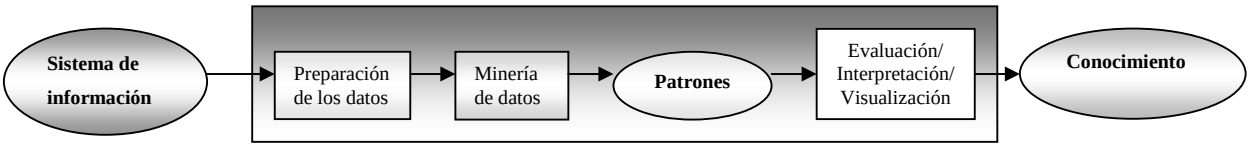


Figura. 1.2 Proceso de KDD

Así, los sistemas *KDD* realizan la selección, limpieza, transformación y la proyección de los datos; también analizan los datos para extraer patrones y los modelos adecuados, interpretan los patrones para convertirlos en conocimiento, consolidan el conocimiento resolviendo los problemas que se presenten y hacen posible el conocimiento para su

uso. Con lo anteriormente mencionado se aprecia la relación que existe entre el KDD y la minería de datos.

Por lo que se puede decir que el *KDD* es el proceso global de descubrir conocimiento útil desde las bases de datos y la Minería de Datos es la aplicación de los métodos que se utilizan para la obtención de patrones y modelos al ser la fase de generación de modelos. (5)

1.5 LA MINERÍA DE DATOS ES UN CAMPO MULTIDISCIPLINAR

La Minería de Datos se ha desarrollado como la prolongación de otras tecnologías. Es por ello que la investigación y los avances en la minería de datos se benefician con lo que producen las áreas relacionadas.

En la siguiente *Figura 1.3* se presentan las áreas más influyentes en la minería de datos.

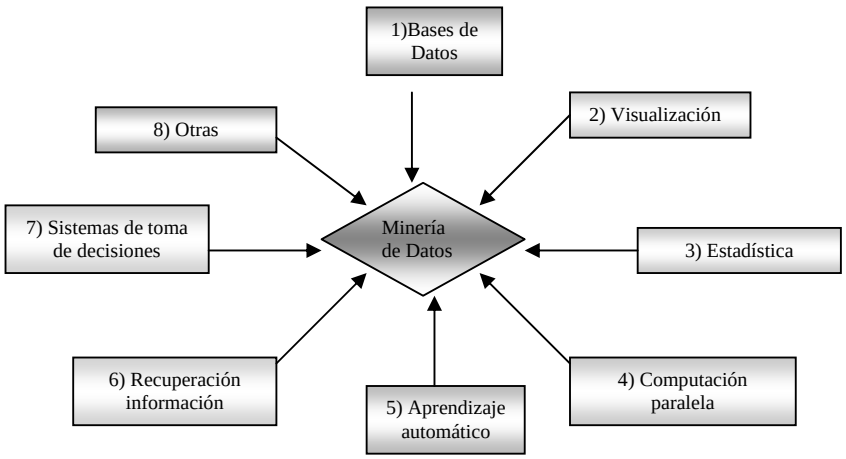


Figura. 1.3 Áreas que contribuyen a la minería de datos

1) **Bases de datos**: tienen una gran relación con la minería de datos; de ellas se extrae el conocimiento novedoso y comprensible. Las técnicas de acceso eficiente a los datos ha sido de gran relevancia para el diseño de algoritmos eficientes de Minería de Datos.
2) **Visualización de datos**: éstas permiten al usuario describir, intuir o entender patrones que son difíciles de ver a partir de descripciones matemáticas o textuales de los resultados, su más claro ejemplo de dichas técnicas de visualización son las gráficas.
3) **Estadística**: esta disciplina ha proporcionado muchos de los conceptos, algoritmos y técnicas que más se utilizan en la Minería de Datos, la regresión lineal y no lineal, la teoría del muestreo, las técnicas bayesianas; sólo por mencionar algunas.
4) **Computación paralela**: actualmente, muchos sistemas de bases de datos incluyen tecnologías de procesamiento paralelo. En este tipo de sistemas las tareas más complejas de Minería de Datos se reparte entre diferentes procesadores en las

computadoras. Una de las principales ventajas de este tipo de aplicaciones en la Minería de Datos es la escalabilidad de los algoritmos.

5) **Aprendizaje automático**: ésta área pertenece a la inteligencia artificial y se ocupa en desarrollar algoritmos y programas que sean capaces de aprender. Los principios de esta disciplina y de la Minería de Datos son los mismos: la máquina aprende de un modelo a partir de ejemplos que usa para resolver el problema.
6) **Recuperación de información**: consiste en obtener información desde datos textuales, por lo que su principal aplicación se ha basado en el uso efectivo de bibliotecas y en la búsqueda por Internet. Un ejemplo de ello es en la búsqueda de documentos a partir de una palabra clave, por medio de un proceso de clasificación de los documentos en función de dichas palabras clave. Para ello se usan medidas de similitud entre los documentos y la consulta. Muchas de estas medidas se han empleado en aplicaciones más específicas de la Minería de Datos.
7) **Sistemas para toma de decisión**: éstos son herramientas y sistemas informatizados que asisten en la resolución de problemas y en la toma de decisiones. Su objetivo es proporcionar información necesaria para realizar decisiones efectivas.
8) **Otras disciplinas**: dependiendo del tipo de datos o de la aplicación que se tenga que realizar, la Minería de Datos usa también técnicas de otras disciplinas como el lenguaje natural, análisis de imágenes, procesamiento de señales, etc. (5)

1.6 APLICACIONES

Actualmente las técnicas que se utilizan en los procesos de Minería de Datos, tienen diversas aplicaciones, ya que pueden ser empleadas para mejorar el rendimiento de los negocios, en los procesos industriales o en los bancos; por mencionar algunos ejemplos, ya que éstos manejan grandes volúmenes de información almacenada en sus bases de datos.

La Minería de Datos se utiliza con gran éxito en aplicaciones de control de procesos productivos; como herramienta de ayuda a la planificación y a la decisión en *marketing*, finanzas, etc.

En el ramo de la mercadotecnia, la Minería de Datos le proporciona a las empresas de Telecomunicaciones, a los bancos y compañías de seguros: la detección de fraudes, optimización de campañas de marketing, predicción de fidelidad de clientes, así como la descripción y segmentación de los mismos.

En la industria del comercio la Minería de Datos es utilizada para diseñar y evaluar campañas de marketing, definir ofertas apropiadas a clientes y predecir riesgos en asignación de créditos a clientes. (7)

Como puede verse, la aplicación de la minería de datos en el uso de la mercadotecnia, da grandes beneficios a las empresas o negocios; ya que a éstas les proporciona el beneficio de no tener que arriesgar el capital de su empresa en decisiones inequívocas y esto se refleja en las utilidades de la misma.

Asimismo, la Minería de Datos es fundamental en la investigación científica y técnica, como herramienta de análisis y descubrimiento de conocimiento a partir de datos de observación o de resultados de experimentos. (5)

En la medicina la minería de datos es utilizada para predecir la efectividad de procedimientos quirúrgicos, exámenes médicos y en la utilización de los medicamentos. Esto con el fin de reducir riegos para las personas y poder prevenir enfermedades oportunamente.

Un ejemplo de este tipo de minería es:

Búsqueda de patrones en pacientes bajo sospecha de Síndrome Metabólico

Aplicación de un proceso de descubrimiento de conocimiento para la obtención de patrones en una base de datos de pacientes que están bajo sospecha de padecer el Síndrome Metabólico. Este proceso parte desde la integración de los datos generales del paciente relacionados con indicadores que miden la resistencia a la insulina y el Síndrome Metabólico, con el objetivo de obtener patrones de asociación entre los factores considerados." (8)

La utilización de la Minería de Datos en esta rama tiene dos grandes logros, la principal es que se puedan atender enfermedades que aún no se presentan en el organismo y económicamente se reducen gastos de tratamiento.

Algunos otros ejemplos de aplicaciones que puede tener la minería de datos son:

- Educación:
  - Detección de abandonos.
  - Detección de estudiantes que tienen posibilidades de terminar sus estudios cuando ya se les ha acabado el tiempo reglamentario.
  - Detección de materias problemáticas o cuellos de botella.
  - Predicciones de alumnos que desertarán o que terminarán todas sus materias.
- Procesos Industriales:
  - Extracción de modelos sobre comportamientos de compuestos.
  - Modelos de calidad.
  - Predicción de fallos y accidentes.
  - Extracción de modelos de calidad.
- Telecomunicaciones:
  - Patrones de llamadas.
  - Modelos de carga de redes.
  - Detección de fraude.
- Seguros y pólizas de vida:
  - Determinación de costo de póliza dependiendo del tipo de cliente.
  - Predicción de qué clientes contratan nuevas pólizas.
  - Identificación de patrones de comportamiento para clientes con riesgo.

- Identificación de comportamiento fraudulento.

- Astronomía:
  - Clasificación de cuerpos celestes.
- Aspectos climatológicos:
  - Predicción de tormentas.
- Inversión en casas de bolsa y banca:
  - Análisis de clientes.
  - Aprobación de préstamos.
  - Determinación de montos de crédito.
  - Análisis de riesgo en créditos.
  - Identificación de reglas de mercado de valores a partir de históricos.
  - Predicción de tendencias en los valores de las divisas, precios del oro y del petróleo.
- Análisis de mercado y distribución
  - Análisis de compra (compras conjuntas, secuenciales, señuelos, etc.).
  - Evaluación de campañas publicitarias.
  - Análisis de fidelidad de los clientes. Riesgo de fuga.
  - Estimación de ventas, costos, etc.
- Ciencias
  - Análisis de secuencias de genes.
  - Determinación de si un compuesto químico causa cáncer.
  - Predicción de recorrido y distribución de inundaciones.
  - Modelos de calidad de aguas, indicadores ecológicos.
- Otras áreas
  - Recursos Humanos: selección de empleados.
  - Turismo: determinación de las características socioeconómicas de los turistas para la determinación de paquetes de viaje.
  - Tráfico: modelos de tráfico.
  - Hacienda: detección de evasión fiscal.
  - Deportes: estudio de la influencia de jugadores y cambios, planificación de eventos.
  - Política: diseño de campañas políticas, estudio de tendencias de grupos, etc.

- Determinación de niveles de audiencia de programas televisivos.

Estos ejemplos muestran que la Minería de Datos se puede aplicar a diversas actividades y que puede ayudar a entender mejor un entorno y, por ende, mejora la toma de decisiones en dicho entorno.

1.7 PROYECTOS EXITOSOS Y DATOS CURIOSOS DE MINERÍA DE DATOS

Dentro de la Minería de Datos hay algunos ejemplos donde se realizó la MD con un resultado exitoso dando datos curiosos que, a simple vista, no se hubieran podido percibir y que fueron de gran ayuda para solucionar problemas o crear estrategias de mercado.

• **BMW Group.** Es uno de los principales constructores de automóviles y motocicletas. En la actualidad aproximadamente 2,100 usuarios monitorean el desarrollo de la producción y los costos de la producción y los costos de material, con una herramienta de reportes basada en web, que les permite reaccionar rápidamente cuando la acción es requerida.

• **Nombres Orientales.** Clientes con nombres cortos en un banco tienden a ahorrar grandes cantidades de dinero y luego retirarlas.

• Los que compran coches de color rojo en Francia tienden a no pagar su préstamo de coche.

• Clientes que compran pañales tienden a comprar cerveza.

Como se puede observar algunas reglas de asociación mostradas podrían tener una interpretación difícil de encontrar e incluso alguna de ellas podría no tener ningún significado.

Conclusiones del capítulo uno.

En el capítulo uno se presenta los conceptos necesarios para entender qué es y para qué sirve la minería de datos, las aplicaciones en las que se puede utilizar, así como los tipos de datos con los que se puede trabajar. Toda esta información de los conceptos básicos se obtuvo de los libros que se leyeron. Además nos aportaron las grandes aplicaciones de la minería (todo su potencial). Definitivamente se considera a la minería de datos una herramienta muy útil que proporciona la información (nuevo conocimiento útil y novedoso) que se necesita para apoyar a la toma de decisiones.

CAPÍTULO II
"FASES DEL KDD (Descubrimiento de conocimiento en Bases de Datos)"

Todo análisis requiere cierta metodología o pasos a seguir y, como ya se mencionó en el capítulo anterior, la Minería de Datos consta de fases que se realizan de una forma secuencial, hasta no haber concluido una, no se pasa a la siguiente; lo cual da cierta seguridad de que el resultado que se obtiene en este proceso sea auténtico y que la información obtenida en él sea verdaderamente útil para nosotros.

El proceso de KDD pasa por las siguientes fases:

1. Sistema de Información (requisitos).
2. Preparación de los datos.
3. Minería de Datos.
4. Patrones.
5. Evaluación, Interpretación y Visualización.
6. Difusión y uso del conocimiento.

2.1 SISTEMA DE INFORMACIÓN (REQUISITOS)

El sistema de información es indispensable para empezar con el análisis, ya que de éste es donde se analizan los requisitos; define el ámbito del problema, define las métricas por las que se evaluará el modelo y define el objetivo final del proyecto de minería de datos. Estas tareas se traducen en preguntas como las siguientes:

- ¿Qué se está buscando?
- ¿Qué atributo del conjunto de datos se desea intentar predecir?
- ¿Qué tipos de relaciones se intenta buscar?
- ¿Desea realizar predicciones a partir del modelo de Minería de Datos o sólo buscar asociaciones y patrones interesantes?
- ¿Cómo se distribuyen los datos?
- ¿Se cuenta con un diagrama Entidad-Relación (DER) para saber cómo se relacionan las tablas? (3)

La naturaleza de las respuestas será quien determine el tipo de Minería de Datos que se deba aplicar, así como la tecnología adecuada.

Por otra parte, para responder a estas preguntas, es probable que se deba dirigir un estudio de disponibilidad de datos para investigar las necesidades de los usuarios de la

empresa con respecto a los datos disponibles. Si los datos no son compatibles con las necesidades de los usuarios, puede que se deba volver a definir el proyecto. (3)

Recopilación de Datos

Para poder analizar y extraer algo útil de los datos es necesario disponer de ellos. Esto en algunos casos puede parecer simple. Se parte de un archivo de datos a analizar. En otros, la diversidad y tamaño de las fuentes hace que el proceso de recopilación de datos sea una tarea compleja. En general, el problema de reunir un conjunto de datos que posibilite la extracción del conocimiento requiere decidir de qué fuentes, internas y externas, se van a obtener los datos; cómo se van a organizar y, finalmente, de qué forma se van a extraer. (5)

Existe una tecnología relativamente reciente denominada “Almacenes de Datos” (Data Warehouses) la cual pretende proporcionar metodologías y tecnología para recopilar e integrar los datos históricos de una organización cuyo fin es el análisis, obtención de resúmenes y extracción de conocimiento. Esta tecnología está diseñada especialmente para almacenar grandes volúmenes de datos de procedencia, generalmente de bases de datos relacionales, aunque es útil para la organización de pequeños conjuntos de datos en aplicaciones más modestas de Minería de Datos. Pero no es imprescindible. Se puede realizar la recopilación de uno o más sistemas transaccionales o incluso archivos de texto *.txt*.

2.2 PREPARACIÓN DE LOS DATOS

Este paso, del proceso de Minería de Datos consiste en recopilar, limpiar, transformar, explorar y seleccionar los datos que se pudieron identificar al definir el problema.

Desafortunadamente, el conjunto de datos está usualmente sucio, compuesto de muchas tablas y tiene propiedades desconocidas. Antes de que cualquier resultado se produzca, los datos deben ser limpiados y explorados – lo cual es una tarea frecuentemente laboriosa.

Hoy en día existen potentes técnicas y herramientas de análisis para la extracción del conocimiento oculto en los datos, pero para poder utilizar dichas técnicas y herramientas, hay que desarrollar una de las partes fundamentales del proyecto y de las que más tiempo lleva, que es la fase de preparación previa a la aplicación de los algoritmos de análisis.

En la *figura 2.1* Se Muestra el esfuerzo requerido en cada una de las fases del KDD

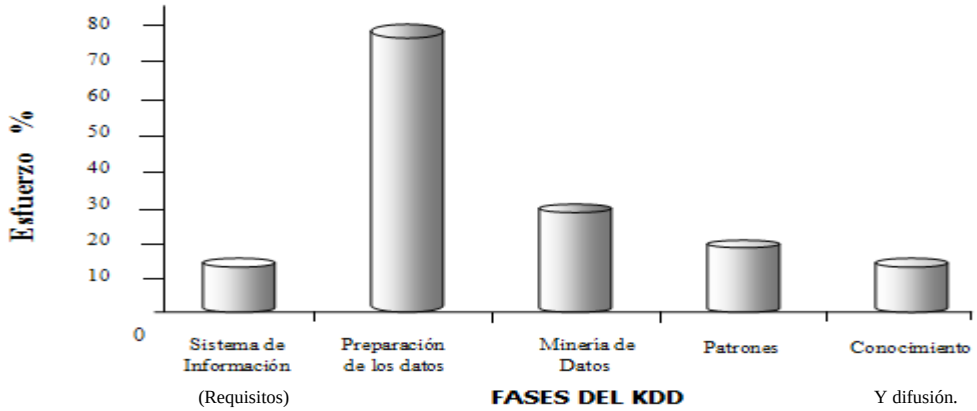


Figura. 2.1. Muestra el esfuerzo requerido en cada etapa del KDD

Los verdaderos desafíos en la tarea de Minería de Datos son:

- Crear un conjunto de datos que contenga la información relevante y exacta y
- Determinar las técnicas de análisis apropiadas.
- Manipulación de grandes volúmenes de información.

Los datos pueden estar dispersos en diferentes bases de datos y almacenados en distintos formatos; también pueden contener incoherencias como entradas que faltan o contienen errores. Antes de empezar a generar modelos, se deben solucionar estos problemas. Normalmente se trabaja con un conjunto de datos muy grande y no se puede

comprobar cada transacción. Por lo tanto, se debe utilizar algún método de automatización, para explorar los datos y encontrar incoherencias. (3)

Los datos contenidos en la fuente de datos nunca es el idóneo, por lo que la mayoría de las veces no es posible utilizar ningún algoritmo de Minería de Datos. Por tal motivo, en el procesamiento se efectúa una filtración de datos para eliminar todos los valores incorrectos, desconocidos, etc.

Desafortunadamente, la fase de selección, exploración y transformación de variables ha sido a la que menos importancia se le ha dado en la bibliografía, por ser una fase de enorme dificultad en la que los datos se analizan y exploran pero no se obtienen resultados definitivos. **La fase de preparación representa la clave del éxito de un proyecto de Minería de Datos**. Puede ser la diferencia entre el éxito y el fracaso, la diferencia entre resultados provechosos, la diferencia entre predicciones interesantes y averiguaciones absurdas. (5)

Limpieza

La recopilación de datos debe ir acompañada de una limpieza e integración de los mismos para que éstos se encuentren en condiciones óptimas para su análisis. Los beneficios del análisis y la extracción de conocimiento a partir de los datos dependen, en su mayoría, de la calidad de los datos recopilados. Por otra parte, debido a las características de las técnicas de Minería de Datos, en ocasiones es necesario realizar una transformación de los datos, para que éstos se adecuen al propósito concreto y las técnicas que se quieren emplear. Por lo que el éxito de un proceso de minería de datos depende no sólo de tener todos los datos necesarios (recopilación), sino de que también éstos estén íntegros, completos y consistentes (limpieza e integración).(9)

La inconsistencia y los valores nulos existen en casi todas las bases de datos. Los datos inconsistentes se ocasionan por distintas razones, como puede ser que los atributos de interés no están siempre disponibles o la información que se tiene es errónea.

Otros datos no se tienen almacenados porque al momento de introducir los datos se pensaba que no eran de interés. (9)

Es por ello que la rutina de la limpieza de los datos se vuelve una pieza fundamental en dicho proceso, ya que ésta ayuda a rellenar valores nulos y va resolviendo las inconsistencias. Los datos sin limpiar pueden ocasionar confusiones para los

procedimientos de análisis pudiendo entonces generar un modelo erróneo. En la *figura 2.2* se simula como se encuentran las bases de datos, llenas de basura y como deben de quedar tras la limpieza.

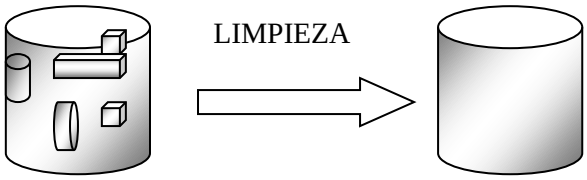


Figura. 2.2 Limpieza de las bases de datos

INTEGRACIÓN

Es importante mencionar que los datos se pueden obtener de distintas fuentes e inclusive se tenga que generar algunas tablas adicionales que no contengan las bases de datos con las que se cuenta, véase la *figura 2.3*. Por lo que antes de proceder con el análisis es importante realizar la integración de dichas bases de datos para que en el futuro no se reencuentren redundancias e inconsistencias debidas a la integración.

El primer problema en realizar la integración de las distintas fuentes de datos es identificar los objetivos, es decir, conseguir que datos sobre el mismo objeto se unifiquen y datos de diferentes objetos permanezcan separados. Existen dos tipos de errores que ocurren en esta integración:

- Dos o más objetos diferentes se unifican. Los datos resultantes mezclarán patrones de diferentes individuos y serán un problema para extraer conocimiento.
- Dos o más fuentes de objetos iguales se dejan separadas. Los patrones del mismo individuo aparecerán repartidos entre varios individuos parciales. (5)

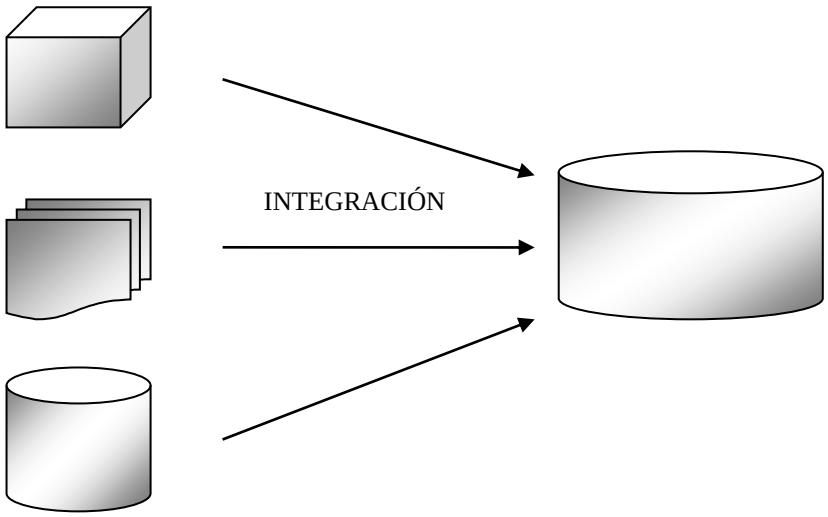


Figura. 2.3 Integración de los datos

TRANSFORMACIÓN

La transformación representa una fase crucial pues el éxito y la exactitud de los modelos que se obtendrán en la fase de Minería de Datos depende de cómo el analista de datos decide estructurar y presentar la entrada de la siguiente fase. Por otra parte, es en esta fase cuando se tienen que codificar los datos para que sean una entrada adecuada para los algoritmos de Minería de Datos que se vayan a utilizar. (9)

Aquí es donde si el algoritmo necesita cambiar la entrada de un atributo por numérica, por categórica o viceversa; por lo que en esta fase, es donde se transforman los datos para adquirir un formato adecuado. Por otra parte, se puede presentar el caso de que no se cuente con todas los atributos necesarios para el algoritmo a utilizar, por ello es que también en esta fase de transformación se generan nuevos atributos.

La creación de nuevos atributos, a partir de los existentes, sirve para crear o agregar características que conllevan a mejorar la calidad, visualización o comprensibilidad del conocimiento extraído. La mayoría o todos los atributos se preservan, por lo que se añaden, no se sustituyen.

Otra transformación frecuente que se realiza es la reducción de atributos, donde en lugar de eliminar atributos redundantes o no necesarios como en casos anteriores, es una consolidación de varios atributos en una mayor semántica (9).

2.3. MINERÍA DE DATOS

La fase de Minería de Datos es la más característica del *KDD*; y, por esta razón, muchas veces se utiliza esta fase para nombrar todo el proceso. El objetivo de esta fase es producir nuevo conocimiento que se pueda utilizar. Esto se realiza construyendo un modelo basado en los datos recopilados para este efecto. El modelo es una representación de los patrones, reglas o relaciones entre los datos que, a simple vista, se desconoce su existencia y que pueden usarse para hacer predicciones, para entender mejor los datos o para explicar situaciones pasadas. (5)

Para poder continuar con el proceso del *KDD* es necesario tomar las siguientes decisiones:

- Determinar que tipo de tarea de minería es el más apropiado (clasificación, predicción, estimación, etc.).
- Elegir el tipo de modelo (árbol de dedición, redes neuronales, etc.).
- Elegir el algoritmo de minería que resuelva la tarea y obtenga el tipo de modelo que se esta buscando.

En el capitulo 3 se detallará en qué consiste una tarea de Minería de Datos, los modelos y se explicarán los algoritmos utilizados.

Continuando con la explicación de la fase de Minería de Datos a nivel general, se dice comúnmente que este proceso de Minería de Datos convierte datos en conocimiento, tal cual alquimista pudiera convertir espigas de trigo en lingotes de oro, o como un minero puede obtener metales preciosos de un montón de rocas. También para algunos autores o estudiosos de dicha materia llegan a decir que el objetivo es extraer “verdad a partir de basura”.

Si se pudiera referir al contexto de trabajo de las fases anteriores del *KDD*, en donde se prepararon los datos para al fin aplicar una técnica de minería de datos, se podrían representar en la siguiente *figura 2.4*.

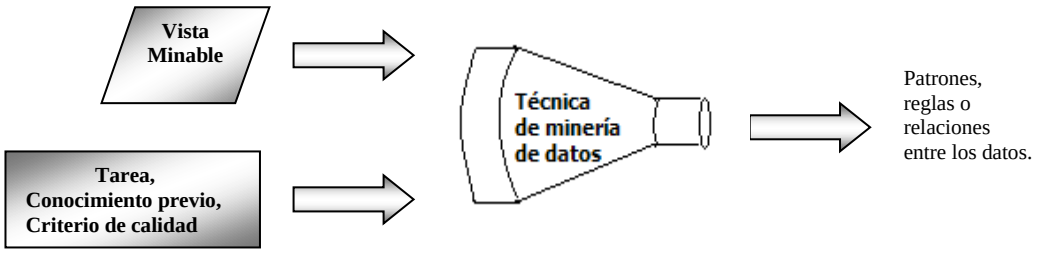


Figura. 2.4 Proceso ideal de minería de datos

En la *figura 2.4* las técnicas de Minería de Datos aparecen como una especie de colador que, al introducirle los datos junto con criterios asociados, descubre patrones de comportamiento o reglas.

Pero ciertamente las cosas no son tan fáciles como meter los datos en dicho colador y que por arte de magia se conviertan en patrones, detrás de todo esto hay muchas horas de trabajo y de aprendizaje.

2.4. OBTENCIÓN DE PATRONES

La extracción de conocimiento a partir de datos tiene como objetivo descubrir patrones que, entre otras cosas, deben ser válidos, novedosos, interesantes y, en última instancia, comprensibles. La gente tiene la capacidad innata de observar patrones a su alrededor; sin embargo, hablando de observar a simple vista patrones de comportamiento en grandes volúmenes de datos con el fin de obtener conocimiento útil y novedoso, se requiere el uso de la Minería de Datos, la cual a su vez hace uso evidentemente de las Tecnologías de la Información.

En cierta forma los patrones presentan una característica en cualquier tipo de aprendizaje, y en cualquier tipo de técnica de Minería de Datos, estos tienen un carácter *hipotético*; es decir, lo aprendido puede, en cualquier momento, ser refutado por evidencia futura. En muchos casos, los modelos no aspiran a ser modelos perfectos, sino modelos aproximados. En cualquier caso, al estar trabajando con hipótesis, es necesario realizar una evaluación de los patrones obtenidos, con el objetivo de estimar su validez y poder compararlos con otros.

Una característica de los modelos de aprendizaje es la manera en la que se expresan los patrones aprendidos. Esta es la razón fundamental del por qué unos métodos van mejor para unos problemas que para otros. En realidad cada método permite expresar mejor ciertos tipos de patrones. De ahí el hecho de que existan tantos métodos; la variedad de métodos permite capturar distintos tipos de patrones. Si falla uno se puede probar con otros. (5)

2.5. EVALUACIÓN, INTERPRETACIÓN, VISUALIZACIÓN

Cuando se llega a estas fases del KDD, surge la duda de cómo poder saber si dicho modelo es lo suficientemente válido para nuestros propósitos. Es por ello que la fase siguiente es la de evaluar qué tan fiable es el modelo obtenido, interpretar los resultados y visualizarlos.

Evaluación

Los métodos de aprendizaje permiten construir modelos o hipótesis a partir de un conjunto de datos, o evidencia. En la mayoría de los casos, es necesario evaluar la calidad de las hipótesis de la manera más exacta posible.

Por ejemplo, si en el ámbito de aplicación de un modelo surge un error en la predicción conlleva a importantes consecuencias (por ejemplo, la detección de células cancerígenas), es importante conocer la exactitud del nivel de precisión de los modelos aprendidos. (5)

Por lo tanto, la fase de evaluación es crucial en la aplicación real de la Minería de Datos. Sin embargo, esta tarea no es fácil ya que ésta depende del tipo del modelo a evaluar.
A continuación se mencionarán las técnicas de evaluación que se utilizarán para evaluar los modelos y reglas o patrones de comportamiento que se obtendrán en esta tesis.

Evaluación mediante validación cruzada.

La validación cruzada es una técnica para evaluar cómo los resultados de un análisis estadístico se generalizarán a un conjunto de datos independiente. Se utiliza principalmente en conjuntos en donde el objetivo es la predicción y se quiere estimar qué tan exacto se va a desempeñar el modelo predictivo.

Una ronda de validación cruzada involucra el particionamiento de una muestra de datos en subconjuntos complementarios llevándose a cabo el análisis en n solo subconjunto (llámese el *conjunto de entrenamiento*) y validando el análisis en otro subconjunto (el *conjunto prueba*). Para reducir variaciones en los resultados, se llevan a cabo múltiples

rondas de validación cruzada usando distintas particiones y los resultados de cada validación se promedian. Véase la *figura 2.5*. (10)

Análisis ROC

Esta técnica de evaluación ROC (Reciver Operating Characteristc) provee herramientas que permiten seleccionar el subconjunto de clasificadores que tienen un comportamiento óptimo general.

Al análisis ROC se utiliza normalmente para problemas de dos tipos de clases (positiva y negativa), y para este tipo de problemas utiliza la siguiente notación para la matriz de confusión:

Estimado	**Real**	
	Positivos Verdaderos	Falsos Positivos
	Falsos Negativos	Negativos Verdaderos

Por ejemplo, para las predicciones de deserción de un conjunto de alumnos se tiene (*figura 2.5*):

	true SI	true NO	class precisic
pred. SI	7934	278	96.61%
pred. NO	1030	13218	92.77%
class recall	88.51%	97.94%	

Figura 2.5 Tabla de matriz de confusión.

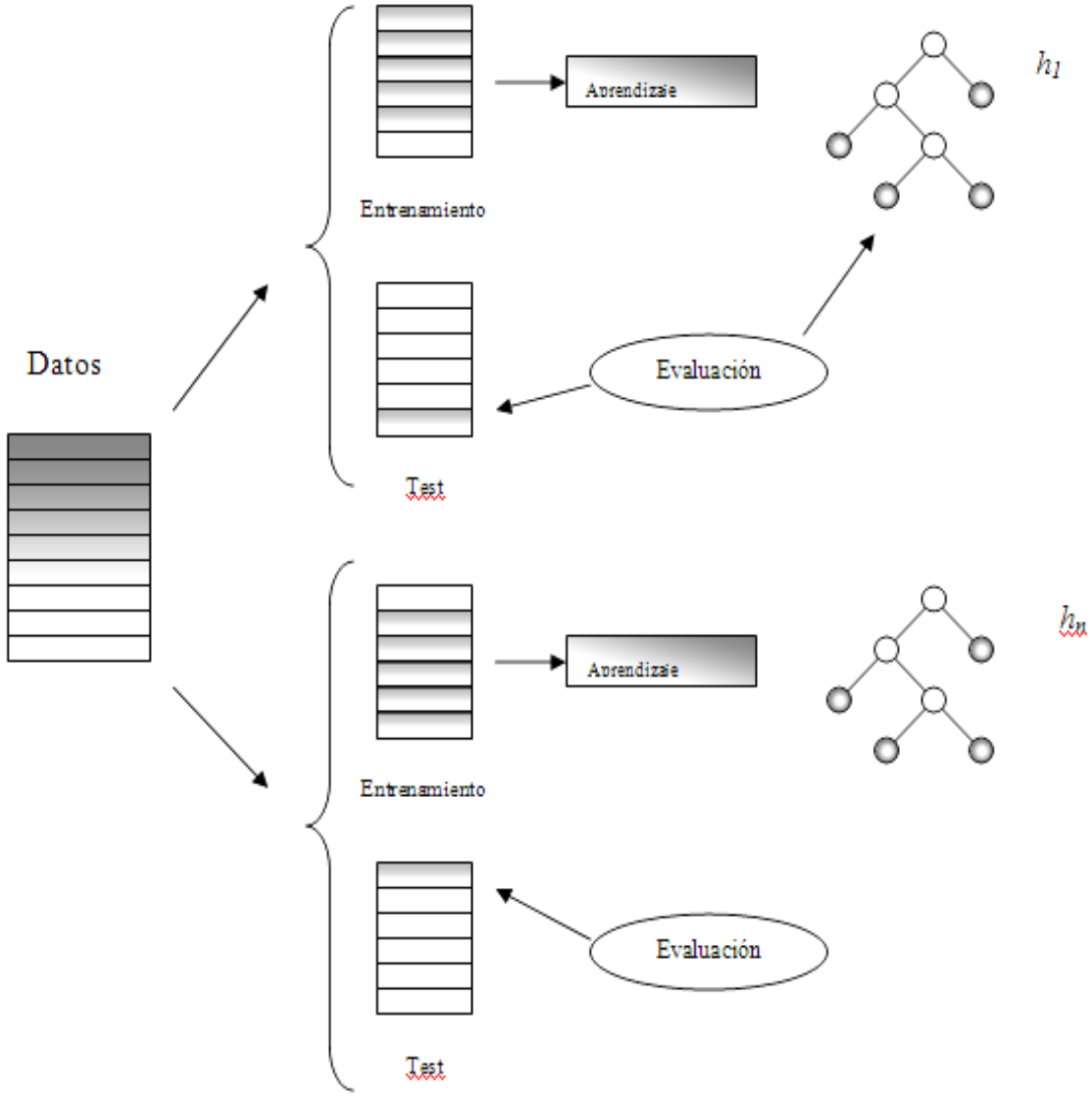


Figura. 2.6 Evaluación mediante validación cruzada.

De la tabla anterior, se lee que para las predicciones de deserción, 7934 casos se predijeron correctamente contra 278 que no se predijeron bien dando una precisión del 96.61%. Asimismo para los casos de la no deserción se observa que 13218 casos contra 1030 resultaron correctos con una precisión del 92.77%.

A partir de la matriz de confusión, se calcula directamente los valores TRP y FPR, dando estas coordenadas para la construcción del clasificador en el espacio ROC, ejemplificadas en la *Figura 2.7*.

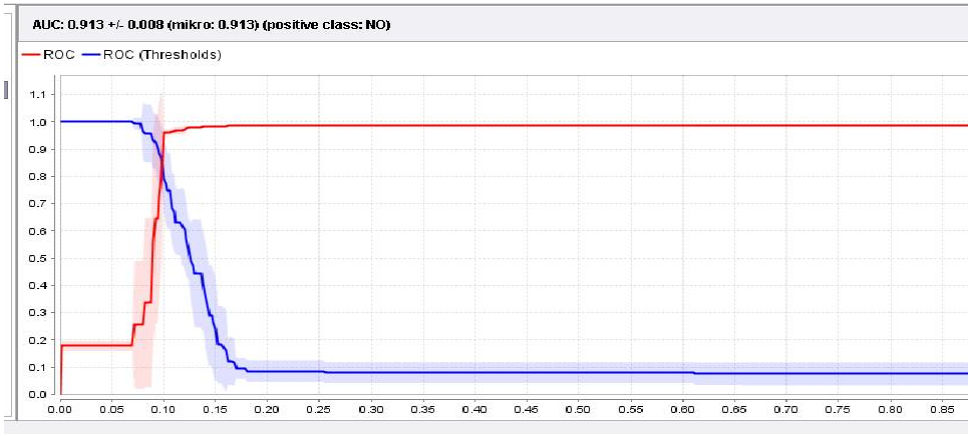


Figura. 2.7 Clasificador en el espacio ROC

De la gráfica anterior se puede observar que la línea roja (la primera que sale en la parte inferior izquierda) tiende hacia la parte superior izquierda. Cuando sucede esto, significa que la precisión del modelo es alta o aceptable, tal y como se observó en el valor de los porcentajes. Si la línea roja formara un ángulo de 45 grados proyectándose hacia la parte superior derecha, significa que el modelo es inexacto y no es bueno para utilizarse. Si sucede lo contrario, que la línea se doble en la parte inferior derecha, definitivamente el modelo es muy malo.

La línea azul (la que empieza en la parte superior izquierda) es el umbral de la línea roja. Para fines prácticos sólo concierne ésta última para indicarnos si el modelo ha tenido un desempeño bueno o malo al momento de clasificar correctamente cada caso.

No hay mucho de qué preocuparse al realizar esta fase, ya que muchas de las herramientas proporcionan elementos que automáticamente generan la evaluación.

Criterios Subjetivos de Evaluación

Las dos técnicas anteriores apuntan a otros criterios que evalúan los modelos tales como:

- *Interés*: es medir la capacidad de ese modelo para suscitar la atención del usuario al modelo.

- *Novedad*: criterio relacionado con la capacidad de un modelo de sorprender al usuario con respecto al conocimiento previo que tenía sobre determinado problema.
- *Comprensibilidad*: la comprensibilidad de un modelo es una factor muy importante y es una cuestión subjetiva desde que un modelo puede ser poco comprensible para un usuario y muy comprensible para otro
- *Simplicidad*: este criterio se basa en establecer el tamaño o complejidad del modelo. Este criterio está muy relacionado con el criterio de comprensibilidad.
- *Aplicabilidad*: en este caso, la calidad de un modelo se basa en su capacidad de ser utilizado con éxito en el contexto real donde va a ser aplicado.

Interpretación

En esta fase interviene el sentido humano. Si aún no son muy claros los patrones generados anteriormente, se puede llegar a pensar que no se ha obtenido un resultado bueno ya que depende de la visión que tenga el analista para analizarlos y con ayuda de otras herramientas; por ejemplo, las estadísticas o incluso de las bases de datos se podrá tener una mejor visualización de los patrones o reglas generadas.

Visualización

La visualización de modelos permite que los usuarios puedan identificar fácilmente y, de manera directa, los patrones más significativos que ha descubierto el modelo. También los métodos de visualización permiten que los propios usuarios modifiquen los modelos para refinarlos o adaptarlos según su conocimiento o circunstancias del ámbito de aplicación.

El uso de técnicas de visualización en su sentido más amplio, permite al usuario descubrir, intuir o entender patrones que serían más difíciles de “ver” a partir de descripciones matemáticas o textuales de los resultados. Existen técnicas de visualización como, por ejemplo, diagramas de barras, gráficas de dispersión, histogramas, etc.; las icónicas (basadas en figuras, colores, etc.), las basadas en píxeles

(cada dato se representa como un único píxel) y las jerárquicas (dividiendo el área de representación en regiones dependiendo de los datos) entre otras.

2.6. DIFUSIÓN Y USO DEL CONOCIMIENTO

Una vez construido y validado el modelo puede usarse principalmente con dos finalidades: para que el analista recomiende acciones basándose en el modelo y en sus resultados, o bien para aplicar el modelo a diferentes conjuntos de datos. También puede incorporarse a otras aplicaciones; por ejemplo, un sistema de análisis de créditos bancarios, que asista al empleado bancario a la hora de evaluar a los solicitantes de los créditos, o incluso automáticamente, como los filtros de spam o la detección de compras con tarjetas de crédito fraudulentas.

En el sector educativo se pueden obtener modelos para explicar las tendencias en las deserciones y los casos en los que los alumnos terminan las materias, detección de las materias más reprobadas y su impacto en el resto de las materias.

La finalidad del proceso del *KDD* es obtener conocimiento para ayudar a entender mejor el entorno donde se desenvuelve la organización y, en definitiva, mejorar la toma de decisiones en dicho entorno.

Conclusiones del capítulo dos.

En el capítulo dos se presenta el proceso del KDD el cual lleva al descubrimiento de conocimiento útil y novedoso. Se explica desde la recopilación de la información, cómo esta se trata y cómo se obtienen los modelos o patrones de comportamiento que llevan finalmente al descubrimiento de dicho conocimiento que funge como un apoyo adicional en la toma de decisiones.

Esta información se obtuvo de los libros citados en las referencias así como de asesorías por parte de especialistas en el ramo. Aporta el procedimiento que se debe seguir para lograr el objetivo principal que es la de encontrar nuevo conocimiento.

CAPÍTULO III
"ALGORITMOS UTILIZADOS EN LA MINERÍA DE DATOS"

El algoritmo de minería de datos es el mecanismo que crea modelos de minería de datos. Para crear un modelo, un algoritmo analiza primero un conjunto de datos, buscando patrones y tendencias específicos. Después, el algoritmo utiliza los resultados de este análisis para definir los parámetros del modelo de minería de datos.

Los algoritmos de minería de datos se clasifican en dos grandes categorías:

- supervisados o predictivos.
- No supervisados o descriptivos. (11)

En los modelos de datos supervisados o predictivos, se utilizan los datos disponibles para construir un modelo que describa una variable particular de interés en términos del resto de los datos disponibles, por lo que los algoritmos predictivos predicen el valor de un atributo (etiqueta) de un conjunto de datos, conocidos otros atributos (atributos descriptivos) a partir de datos cuya etiqueta se conoce cómo se induce una relación entre dicha etiqueta y otra serie de atributos. Esas relaciones sirven para realizar la predicción en datos cuya etiqueta se desconoce. Se desarrolla en dos fases:

1) Entrenamiento.- Es la construcción de un modelo usando un subconjunto de datos con etiquetas conocidas.
2) Prueba.- Prueba del modelo sobre el resto de los datos.

Cuando una aplicación no es lo suficientemente madura, no tiene el potencial necesario para una solución predictiva, en ese caso hay que recurrir a los métodos no supervisados o descriptivos (descubrimiento del conocimiento) que descubren patrones y tendencias en los datos. El descubrimiento de esa información sirve para llevar a cabo acciones y obtener un beneficio de ellas. (11)

3.1 TAREAS DE MINERÍA DE DATOS

Dentro de la minería de datos existen diferentes tipos de tareas, y cada una de ellas puede considerarse como un tipo de problema que se puede resolver con un algoritmo de minería de datos, ya que cada tarea tiene sus propios requisitos, el resultado obtenido (información) de cada una de ellas puede diferir del resultado con otra tarea.

Como ya se mencionó anteriormente, existen dos tipos de modelos: los predictivos y los descriptivos, por lo que las distintas tareas de Minería de Datos se tienen que enfocar a alguno de estos dos tipos de modelos.

La minería de datos sirve para trabajar con tareas y las principales son las siguientes:

- Clasificación
- Regresión
- Predicción
- Agrupamiento
- Agrupación por afinidad o reglas de asociación
- Correlaciones

Las tres primeras tareas (clasificación, regresión y predicción) son ejemplos de minería de datos predictiva. Las tres restantes son ejemplos de minería de datos descriptiva (clustering, reglas de asociación y correlación).

Clasificación

La clasificación consiste en examinar las características de un objeto asignándolo a una clase predefinida. Consiste en actualizar cada registro llenando un campo con una "marca" que lo identifica como integrante de una cierta clase.

La tarea de la clasificación se caracteriza por una buena construcción, definición de las clases y un conjunto consistente de ejemplos preclasificados para entrenamiento. El objetivo es construir un modelo que pueda ser aplicado a datos sin clasificar para clasificarlos. (9)

Ejemplos de las tareas de la clasificación incluyen:

- Clasificación de clientes de bajo, medio y alto riesgo en aplicaciones de crédito.
- Clasificación de clientes a segmentos predefinidos.

En estos ejemplos, hay un número limitado de clases conocidas y se espera poder asignar cualquier registro dentro de cualquiera de esas clases.

Existen variantes de la tarea de la clasificación, como son el aprendizaje de "rankings" (es decir, orden de resultados o listas de clasificaciones; por ejemplo, en competencias deportivas: *México ocupa el tercer lugar*), el aprendizaje de preferencias (por ejemplo, en la generación de predicciones acerca de quién ganará las elecciones o que un modelo aprenda a ordenar cosas), el aprendizaje de estimadores de probabilidad (aprendizaje con ayuda de la estimación de probabilidades de los atributos etiqueta sobre un cierto número de datos de entrada), etc.

Regresión

Esta tarea consiste en aprender una función real que asigna a cada valor del registro de la base de datos un valor real. Esta es la principal diferencia respecto a la clasificación; el valor a predecir es numérico. El objetivo principal de esta tarea es minimizar el error entre el predicho y el real. (9)

Con ciertos datos de entrada se puede usar la estimación para alcanzar con una variable desconocida algo como créditos ó balances de tarjetas de crédito. En la práctica, la estimación es utilizada para mejorar los resultados arrojados por la clasificación.

Frecuentemente la clasificación y la estimación se utilizan juntas, como cuando la Minería de Datos es usada para predecir quiénes realizarán una transferencia y entonces se estima el volumen de la transferencia de acuerdo a lo *aprendido* previamente por el clasificador. Los datos con los que el clasificador ha aprendido, bien pueden ser datos históricos de ingresos, gastos u otros que se consideren relevantes para el buen aprendizaje; siempre y cuando se cuente con ellos (que se tengan disponibles en alguna base de datos o almacén de datos).

Predicción

Cualquier predicción puede ser directamente una clasificación o estimación. La diferencia es el énfasis. Cuando la Minería de Datos es usada para clasificar el uso primario de una línea telefónica casera (llamadas normales, Internet) o detectar transacciones fraudulentas en una tarjeta de crédito, no se espera regresar para verificar si la clasificación fue correcta. (9)

En la tarea de predicción los registros son clasificados según algún comportamiento futuro predicho o un valor estimado. Con la predicción, la única forma de verificar la predicción de la clasificación es esperar y ver. Sin embargo, el desempeño de dicha clasificación se puede ir verificando con los datos históricos; es decir, ir evaluando si el clasificador clasifica correctamente la mayor parte de los datos pasados. Pero no todos tendrán que ser correctos desde que lo que sucede en el universo en general es dinámico, las cosas siempre están cambiando constantemente por lo que las tendencias son diferentes en los distintos intervalos del tiempo.

Algunos ejemplos de las tareas de predicción son:

- Predecir el valor del dólar en los próximos días (por ejemplo, según los precios del petróleo y del oro y otros índices de algunas bolsas de valores).
- Predicción del estado del clima en los próximos días.
- Predicción de cuáles clientes de un banco devolverán el préstamo y cuáles no.

Cualquiera de las técnicas usadas como clasificación o estimación puede ser adaptada para usar ejemplos de entrenamiento donde el valor de la variable a predecir es conocido, junto con datos históricos del ejemplo.

Los datos históricos son usados para construir un modelo que explique el comportamiento observado. Cuando éste modelo es aplicado a las entradas actuales, los resultados son la predicción de un comportamiento futuro.

Agrupamiento

La tarea descriptiva más representativa es la de agrupamiento (*clustering*) y consiste en obtener grupos "naturales" a partir de los datos. La gran diferencia entre la agrupación y la clasificación es: que en lugar de analizar datos etiquetados con una clase, los analiza para generar una etiqueta. (9)

En el agrupamiento, los datos se agrupan basándose en el principio de maximizar la similitud entre los elementos de un grupo y minimizando la similitud entre los distintos grupos. Es decir, se forman grupos que los objetos de un mismo grupo son muy similares entre sí y son muy diferentes a los objetos de cualquier otro grupo.

Otro nombre que también se le da al agrupamiento es el de segmentación, ya que el principio de éste es: partir o segmentar en grupos a los datos. Un claro ejemplo donde se puede aplicar la agrupación es en una tienda de discos que desea identificar grupos de clientes: clientes que compran videos y discos de música, indican a qué tipo de grupo cultural pertenecen.

Agrupación por Afinidad o Reglas de Asociación

La agrupación por afinidad o mejor conocida como reglas de asociación es una tarea que tiene como objetivo identificar relaciones no explícitas entre atributos categóricos. Las reglas de asociación no implican una situación causa-efecto, por lo que no precisamente debe existir una causa para que los datos estén asociados. (9)

El objetivo principal de las reglas de asociación es poder determinar cuáles cosas van juntas. El ejemplo prototipo es determinar qué cosas van juntas en un carro de compras de un cliente en un supermercado. Las ventas en "cadena" pueden usar reglas de asociación para planear un arreglo de elementos en venta o ser puestas en un catálogo juntas para ser vistas por el cliente para su adquisición. Las reglas de asociación pueden entonces ser usadas para identificar las oportunidades de ventas cruzadas y poder diseñar paquetes atractivos o grupos de productos y servicios.

Hay que mencionar que existe un caso especial de las reglas de asociación el cual es nombrado como reglas de asociación secuenciales. En instancia se utilizan para determinar patrones secuenciales en los datos, los cuales se basan en secuencias temporales de acciones, por lo que la gran diferencia entre las reglas de asociación es que la relación de los datos es basada en el tiempo. (9)

Correlaciones

Esta tarea se utiliza básicamente para examinar el grado de similitud de los valores de dos variables numéricas.

Un ejemplo de este tipo de tareas es: Si un inspector de incendios desea obtener información útil para la prevención de incendios probablemente esté interesado en conocer las correlaciones negativas que existen entre el empleo de distintos grosores de protección del material eléctrico y la ocurrencia de frecuencia de los incendios.

3.2 TÉCNICAS DE MINERÍA DE DATOS

Dado que la Minería de Datos es un campo muy interdisciplinario, existen diferentes paradigmas detrás de las técnicas utilizadas para ésta fase: árboles de decisión, redes neuronales, aprendizaje bayesiano, técnicas estadísticas, entre otras. Cada uno de estos paradigmas tiene diferentes algoritmos y variaciones de los mismos, así como restricciones que hacen que la efectividad del algoritmo dependa del dominio de aplicación, no adoptando así el método universal aplicable a cualquier tipo de aplicación.

3.2.1 ÁRBOL DE DECISIÓN

Inicialmente desarrollados por Morgan y Sonquist en 1963, los árboles de decisión son una técnica para el aprendizaje de ***modelos comprensibles de decisión*** elaborados a partir de una muestra de datos disponible. El término “modelo” indica que este tipo de técnicas constituyen una “hipótesis” o “representación” del comportamiento de los datos. Y es “comprensible”. (9)

Los sistemas de aprendizaje basados en árboles de decisión son quizás el método más fácil de utilizar y entender. Un árbol de decisión es un conjunto de condiciones organizadas en una estructura jerárquica, de tal manera que la decisión final a tomar se puede determinar siguiendo las condiciones que se cumplen desde la raíz del árbol hasta alguna de sus hojas. Los árboles de decisión se utilizan desde hace siglos y son apropiados para expresar procedimientos médicos, legales, comerciales, estratégicos, matemáticos, lógicos, etc. (9)

El principio básico de los árboles de decisión es que cada nodo del árbol representa un atributo de los registros del problema debiendo partir de un nodo raíz el cual hace la mejor discriminación sobre el punto de partida de clasificación o predicción que se desea obtener. Se realiza una prueba al registro en análisis con respecto a este nodo raíz, después se continúa con un nodo hijo el cual a su vez también realiza una prueba al registro y así sucesivamente hasta llegar a un nodo final llamado nodo hoja. Esta arquitectura arborescente invertida es la que da el nombre a los árboles de decisión.

Una de las grandes ventajas de los árboles de decisión es que, en su forma más general, las opciones posibles a partir de una determinada condición son excluyentes. Lo que permite analizar una situación y, siguiendo el camino apropiadamente de dicho árbol, se puede llegar a una sola solución a tomar.

Existen dos tipos principales de árboles de decisión:

- **Árboles de decisión para clasificación**. Proveen un nivel de confianza de clasificación y asignan a los registros en una clase según sus características.
- **Árboles de decisión para regresión**. Estiman el valor de la variable objetivo (valores numéricos). (9)

Árboles de decisión para clasificación

Los árboles de decisión se adecuan mejor a la clasificación desde el hecho de que clasificar es identificar a qué clase pertenece algún objeto, por lo que la estructura de condición y ramificación es ideal para dicho problema.

La característica principal de la clasificación es que se asume que las clases son disjuntas; es decir, alguna instancia de una clase es diferente a la instancia de otra clase puesto que no puede pertenecer a la misma clase. (12)

Como la clasificación trata con clases o etiquetas disjuntas, un árbol de decisión siempre conducirá un ejemplo hasta una y sólo una hoja. Para ello, las particiones existentes en el árbol deben ser también disjuntas; cada instancia cumple o no cumple una condición. Además, siempre se debe cumplir alguna de las dos condiciones.

EJEMPLO: Se trata de un conjunto de datos que muestra condiciones climatológicas (pronóstico, humedad y viento) adecuadas para jugar un cierto deporte. Se trata de decir si se debe jugar o no, se muestra en la *Figura 3.1*

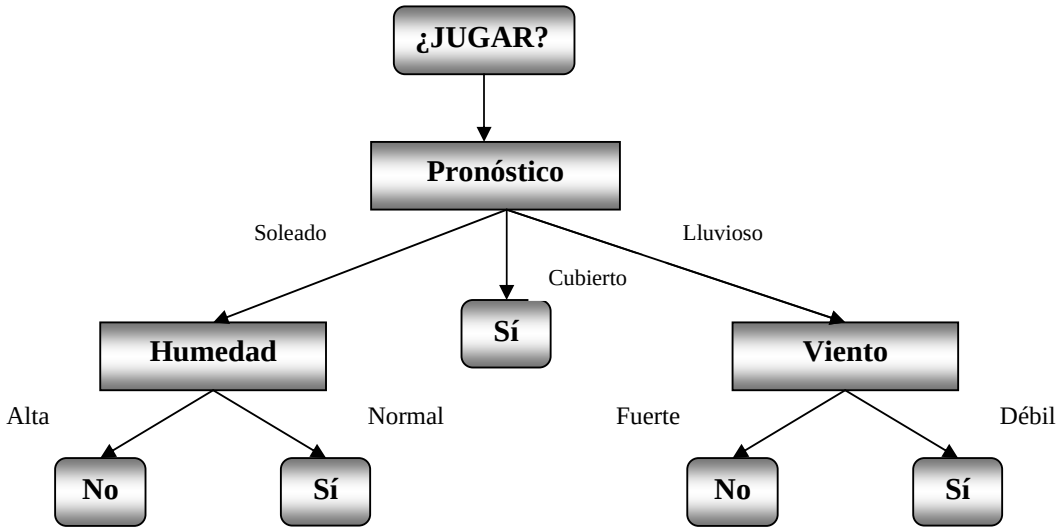


Figura. 3.1 Árbol de decisión para determinar si se juega o no cierto deporte.

Random Forest – Bosque Aleatorio.

Un bosque aleatorio es un clasificador que consiste de muchos árboles de decisión y despliega como resultado la clase que es la moda de las clases de salida de árboles individuales.

Cada árbol se construye usando el siguiente algoritmo:

• Sea *N* el número de casos y *M* el número de variables en el clasificador.
• Se sabe el número *m* de variables de entrada que se usa para determinar la decisión en el nodo del árbol; *m* debe ser mucho menor que *M*.
• Elegir un conjunto de entrenamiento para este árbol escogiendo *N* veces con remplazamiento de todas los casos *N* de entrenamiento disponibles. Utilizar el resto de los casos para estimar el error del árbol prediciendo sus clases.
• Para cada nodo del árbol, escoger aleatoriamente *m* variables en las cuales se base la decisión en ese nodo. Calcular la mejor partición basada en estas *m* variables en el conjunto de entrenamiento.
• Cada árbol crece plenamente y no está podado (tal y como se construyen los árboles clasificadores normales).

Ventajas.

• Para muchos conjuntos de datos, produce un clasificador altamente exacto.

• Puede manejar un número muy grande de variables de entrada.

• Estima la importancia de las variables para determinar la clasificación.

• Genera una estimación interna imparcial del error de generalización conforme la construcción del bosque progresa.

• Incluye un buen método para estimar los datos faltantes y mantiene la exactitud cuando una enorme porción de los datos faltan.

• Provee una manera experimental para detectar interacciones variables.

• Puede balancear el error en conjuntos de datos de población no balanceada.

• Calcula proximidades entre casos, lo cual es útil para el agrupamiento, detección de casos extraños y visualización de los datos.

• El aprendizaje es rápido.

Desventajas.

• Los bosques aleatorios son propensos al sobre juste. Esto se da en tareas de regresión o clasificación con mucho ruido.

El índice Gini – *Gini index*

Al utilizar el algoritmo del bosque aleatorio o *random forest*, se tienen varios tipos de índices a escoger, de los índices que mejores resultados ofrecieron para nuestro estudio, es el índice *Gini*.

Para el problema de clasificación de la clase *k*, el índice *Gini* se define como $G = \sum_k p_k(1 - p_k)$, donde p_k es la proporción de observaciones en el nodo en la *k-ésima* clase. El índice es minimizado cuando uno de los p_k toma el valor de 1 y todos los demás tienen el valor 0. En este caso, se dice que el nodo es *puro* y no más particiones adicionales de las observaciones en ese nodo tomarán lugar. El índice *Gini* toma su valor máximo cuando todos los p_k adoptan el valle $1/K$; de esta manera, las observaciones en el nodo son expandidas por igual entre las clases *K*. El índice *Gini*, para todo un árbol de clasificación, es una suma ponderada de los valores del índice *Gini* en los nodos terminales, con los pesos que son los números de observaciones en los nodos. Así, en la selección del siguiente nodo a separar, los nodos que tienen grandes números de observaciones pero para los cuales sólo pequeñas mejoras en los p_ks pueden realizarse,

pueden compensarse contra los nodos que tienen pequeños números de observaciones para las cuales grandes mejoras en las p_ks son posibles.

Árboles de decisión para regresión

Los árboles de decisión no sólo se utilizan para la clasificación, sino que también se pueden adaptar para otras tareas como son la regresión, el agrupamiento o la estimación. Un árbol de regresión se construye de manera muy similar al árbol de decisión para clasificación, pero con las siguientes diferencias:

- La función aprendida tiene dominio real y no discreto.
- Los nodos de las hojas el árbol se etiquetan con valores reales.

Existe una variedad de algoritmos para construir árboles de decisión para regresión, con los cuales se obtienen resultados ligeramente diferentes. Estos algoritmos son:

a) CART
b) CHAID
c) C4.5
d) ID3

¿Cuándo utilizar árboles de decisión?

Los árboles de decisión son útiles para problemas cuyo objetivo es hacer predicciones o clasificaciones categóricas extensas. Asimismo son muy útiles para encontrar valores anómalos o extraordinarios que bien pueden guiar a la detección de fraudes o casos poco comunes por lo que es posible hacer descubrimientos inesperados útiles. Los árboles de decisión son más útiles en dominios en los que los valores de las variables pueden romperse en números relativamente pequeños (en algunos casos sólo dos valores, dependiendo de la herramienta que se utilice).

3.2.2 REDES NEURONALES ARTIFICIALES

Las redes neuronales artificiales (RNA) son todo un paradigma en la computación. Sus inicios datan del año 1943 con los trabajos de McCulloch y Pitts. El acceso al conocimiento neuronal ha permitido que hoy en día sea una eficaz técnica predictiva para modelar problemas complejos de la minería de datos. (13)

Las RNA son el resultado de los intentos por reproducir mediante computadoras el funcionamiento del cerebro humano. Los modelos de las RNA creados hasta ahora son extremadamente simples y lo que se busca no es imitar las neuronas auténticas, sino lograr una máquina (computadora) en paralelo formada por la interconexión de muchos elementos simples de cálculo. (9)

Las redes neuronales artificiales son un método de aprendizaje cuya idea básica es imitar ciertos aspectos de la arquitectura del cerebro, para que ésta resuelva los problemas modelados.

Las RNA son sistemas conexionistas dentro del campo de la Inteligencia Artificial las cuales, dependiendo del tipo de arquitectura neuronal, pueden tener diferentes aplicaciones. Pueden utilizarse para el reconocimiento de patrones, compresión de información, agrupamiento, clasificación, visualización, etc.

Una red neuronal puede verse como un grafo dirigido con muchos nodos (elementos del proceso) y arcos entre ellos (sus interconexiones). Cada uno de estos elementos funciona independientemente de los demás, usando sus datos de entrada y de salida del nodo para dirigir su procesamiento. En la *figura 3.2* se ejemplifica cómo funciona una red.

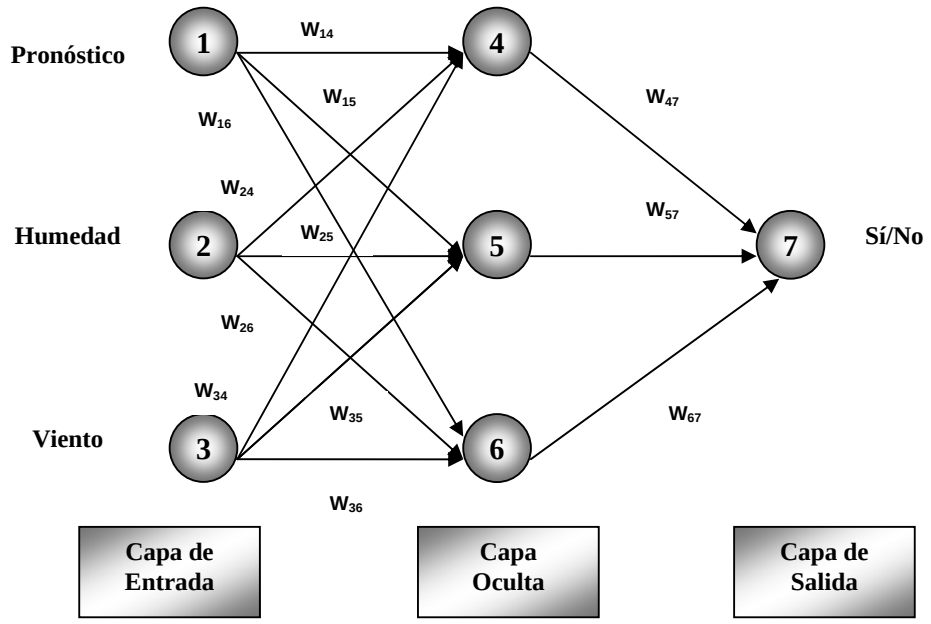


Figura. 3.2 Red Neuronal para el problema de jugar un cierto deporte.

La organización más simple de una RNA consta de una capa de entrada, en la que cada nodo corresponde a una variable independiente, nodos internos organizados en una o varias etapas ocultas y una capa de salida con los nodos de salida que son las posibles respuestas a nuestro problema. Cada nodo de la capa de entrada está conectado a cada nodo de la capa oculta; estos nodos de la capa oculta pueden estar conectados a otras capas ocultas o bien a nodos de la capa de salida. Cada arco está etiquetado por un peso de conexión *w*. (12)

Los pesos de conexión son paramétricos y desconocidos, los cuales deben estimarse por un método de entrenamiento. El método más utilizado es el de retropropagación (*backpropagation*). La idea es reducir el valor del error de la salida de la red. (12)

Para usar una red neuronal ya entrenada se introducen los valores de los atributos de una instancia en los nodos de entrada y los nodos de salida determinan la predicción para dicha instancia. Suena muy sencillo como si se tratara de una caja negra donde de un lado se introducen datos y del otro lado sale la respuesta. Tal vez sí sea así, pero lo complejo de las redes neuronales está en el aprendizaje y en lograr el mínimo error de los valores de salida.

Las RNA no necesitan volver a ser programadas al cambiar de entorno, pero no precisamente quiere decir que sus comportamientos no cambien con la finalidad de adaptarse a nuevos entornos. Estos cambios son debidos a variaciones en los pesos de la red, lo que da lugar al aprendizaje. Dichos cambios son producidos para modelar los cambios en el rendimiento de las sinapsis, dado que la forma de aprendizaje humana se debe a los cambios en el rendimiento o eficiencia de las sinapsis, ya que a través de ellas se transmite la información entre las neuronas.

La faceta más interesante del aprendizaje no es sólo la posibilidad de que los patrones de entrada puedan ser aprendidos, clasificados, identificados; sino la capacidad de generalización que posee, ya que en el aprendizaje se dan los patrones de entrenamiento. Lo más importante es que la red tenga la capacidad de generalizar sus resultados en un conjunto de prueba los cuales no fueron vistos dentro del aprendizaje.

Uno de los problemas que se tiene en las redes neuronales es el “sobreaprendizaje”, también llamado sobre ajuste que, como su nombre lo indica, hace que el resultado no sea totalmente confiable.

Existen dos tipos principales de aprendizaje en RNA:

- Aprendizaje supervisado.
- Aprendizaje no supervisado.

Aprendizaje supervisado.

El aprendizaje ocurre en un modo supervisado cuando los sistemas se enfocan a un conjunto de objetivos conocidos para que esos objetivos sean fácilmente identificados cuando se presenten como entradas al sistema. En cada prueba que se hace con el aprendizaje supervisado, una entrada se presenta al sistema; la entrada activa ciertos nodos y el sistema proporciona una respuesta de salida basado en el patrón de activación. Si la salida no es igual a la respuesta deseada durante el aprendizaje, se le da al sistema una retroalimentación diseñada para modificar la respuesta incorrecta.

Una variedad de algoritmos puede utilizarse para proporcionar retroalimentación al sistema durante las pruebas de aprendizaje de manera que la fuerza de las conexiones entre los nodos se altere, de tal forma que produzcan respuestas correctas consistentemente. Una vez que el sistema ha aprendido las respuestas correctas del conjunto de entradas de preparación, el modo aprendizaje estará terminado y la red neuronal puede entonces utilizarse para detectar y clasificar aquellos patrones entre las nuevas entradas.

De esta forma, habiendo preparado, el sistema puede utilizarse para automatizar la detección de patrones y alertar al usuario cuando patrones entrantes sean iguales a una respuesta de salida aprendida previamente. Este es el tipo de formulación en el corazón de las redes de propagación hacia atrás y otros métodos utilizados en sistemas de aprendizaje supervisado. (14)

Cuándo utilizar las redes neuronales supervisadas.

Los sistemas de aprendizaje supervisado son apropiados en aquellos casos en el que los trazos o mapeos de la entrada-salida son conocidos de antemano, pero no pueden decirle nada nuevo sobre sus datos.

Aprendizaje no supervisado.

Los modelos de red neuronal que pueden utilizarse para la Minería de Datos aprenden en un modo no supervisado. Los sistemas de aprendizaje no supervisado no requieren que el conjunto de respuestas de salida permisibles y su mapeo a entradas sean definidas a priori. En vez de eso, lo que ocurre en el aprendizaje no supervisado es que la red forma su propio conjunto de salidas durante la preparación basado en características extraídas por la red.

El método más popular del aprendizaje no supervisado es el mapa de características de *Kohonen*. Hay dos capas en el mapa de características, una capa de entrada y una capa de salida. Los nodos en la capa de salida pueden tener conexiones excitatorias así como inhibitorias con nodos vecinos. Dada una entrada, los nodos dentro de la capa de salida pasan por un proceso de activación competitiva por el que los nodos activados intentan

e inhiben a otros nodos para “ganar” la competencia. El ganador de la competencia es usualmente el nodo (o el conjunto de nodos) cuya conexión entrante se pondera más cercanamente a ser igual que la entrada. Habiendo ganado ya la competencia, el ganador tiene sus ponderaciones ajustadas incluso más allá en la dirección del patrón de entrada.

Un mapa de características se crea ajustando no sólo las ponderaciones del ganador después de cada entrada, sino también las ponderaciones de sus nodos vecinos. De esta manera, sobre las pruebas, la vecindad entera de nodos responderá a algún grado hacia la misma entrada. La preparación usualmente progresa reduciendo el tamaño de la vecindad elegible para el ajuste hasta que la vecindad se convierte en un solo nodo de salida.

Hasta este punto, puede haber un nodo que responda más fuertemente a una entrada dada, pero este nodo estará conectado cercanamente a una red de nodos que también tiene una tasa alta de respuesta a esa misma entrada. Esto significa que entradas similares tenderán a activar la misma vecindad de nodos.

Lo que es interesante sobre esto, desde una perspectiva de Minería de Datos, es que se pueden alimentar independientemente las entradas en una red de aprendizaje no supervisado y utilizar el proceso competitivo para dejar que la red determine cuáles características podrían utilizarse para separar clases de entradas en categorías o en grupos (*agrupamientos*).

El mapa puede segmentarse para dividir los datos en subclases para un análisis posterior por parte de otro mapa de características o por algún otro método. Ya que la salida de ésta computación de red neuronal se representa por un conjunto de vectores, necesitará usualmente combinar este enfoque con otras metodologías de visualización para apreciar los resultados.

Cuándo utilizar las redes neuronales no supervisadas.

La ventaja más importante del aprendizaje no supervisado sobre otros métodos analíticos es que no necesita empezar con hipótesis sobre las diferencias esperadas entre grupos representados en el conjunto de datos. Esto puede ser usado como un análisis

exploratorio. Generalmente hablando, el método de la red neuronal para la Minería de Datos es de lo más útil cuando están buscando maneras ingeniosas de segmentar el conjunto de datos. Este método puede utilizarse para descubrir subgrupos de datos que están definidos en términos de alguna característica común que los separa de otras porciones de la población completa.

Las aplicaciones en las que se podría usar tal análisis incluyen la identificación de los tipos de clientes que más probablemente comprarán ciertos productos; designación de constelaciones de síntomas, que podrían utilizarse para clasificar varias enfermedades y la caracterización de las características que podrían segregar patrones de comercio sospechosas de un grupo.

A continuación se presenta una tabla (*tabla 3.1*) de diferencias que existe entre el cerebro humano y la computadora donde se visualizan las propiedades más interesantes de cada uno de ellos.

CEREBRO	**COMPUTADORA**
• 100.000 millones de unidades de proceso. • Cientos de operaciones por segundo. • Precisión aritmética muy escasa. • Paralelismo masivo. • Lógica difusa. • Memoria de tipo asociativo y almacenada en forma dispersa. • Tolerancia a los fallos (muerte de neuronas). • Maneja todo tipo de información incluso sujeta a incertidumbre, en poco tiempo pero no necesariamente con exactitud.	• Una unidad de proceso • Millones de operaciones por segundo. • Precisión aritmética absoluta. • Operaciones en serie. • Lógica rígida. • La información se guarda en posiciones de memoria de acceso directo. • Los pequeños fallos son críticos. • Sistemas altamente especializados con capacidad para procesar información muy concreta, siguiendo instrucciones Dadas.

Tabla 3.1. Diferencias entre el cerebro humano y la computadora.

3.2.3 ALGORITMOS DE CLASIFICACIÓN

Algoritmo del vecino *k* más cercano.

En cuanto se refiere a reconocimiento de patrones, el algoritmo de los vecinos *k* más cercanos (*k*-NN) es un método de clasificación de objetos que se basa en los ejemplos de entrenamiento más cercanos en el espacio característico. *k*-NN es un tipo de aprendizaje basado en instancias o aprendizaje perezoso donde la función es aproximada solamente localmente y todo el cómputo se pospone para la clasificación. También puede utilizarse para la regresión.

El algoritmo de los vecinos *k* más cercanos es, de entre los algoritmos de aprendizaje máquina, el más simple. Un objeto es clasificado por un voto de mayoría de sus vecinos junto con el objeto más común asignado entre sus vecinos más cercanos *k*. *k* es un entero positivo típicamente pequeño. Si $k = 1$, entonces el objeto simplemente es asignado a la clase a la que pertenece su vecino más cercano. En problemas de clasificación binarios (dos clases), es útil escoger que *k* sea un número entero impar de tal manera que se eviten votos empatados.

Los vecinos se toman de un conjunto de objetos para los cuales la correcta clasificación (o en el caso de la regresión, el valor de la propiedad) es conocida. Esto se puede considerar como el conjunto de entrenamiento para el algoritmo aunque ningún paso de entrenamiento sea requerido. Para identificar a los vecinos, los objetos son representados por vectores de posición en un espacio característico multidimensional. Usualmente se usa la distancia Euclidiana $d(X,Y) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ aunque otras medidas de distancia tales como la distancia Manhattan $d = \sum_{i=1}^{n} | x_i - y_i |$ podría utilizarse en principio.

El algoritmo del vecino más cercano es sensible a la estructura local de los datos.

Por ejemplo, si se quiere clasificar una nueva fruta (*figura 3.3*):

Figura 3.3. Clasificando a la nueva fruta. Resulta ser una manzana.

- Si k=5, estas cinco frutas son las más similares al ejemplo no clasificado.
- Ya que la mayoría de las frutas son manzanas, se decide que la fruta desconocida es una manzana.

Este tipo de clasificador requiere de tres cosas:

- El conjunto de los registros almacenados.
- La distancia métrica para calcular la distancia entre los registros.
- El valor de *k*, el número de vecinos a obtener.

Lo que hace este clasificador para clasificar un nuevo registro es:

- Calcula la distancia a otros registros de entrenamiento.
- Identifica los vecinos *k* más cercanos.
- Utiliza las etiquetas de clase de los vecinos más cercanos para determinar la etiqueta de la clase del registro desconocido.

La clase no es más que el atributo a predecir.

En la siguiente ilustración se tiene al nuevo registro a clasificar. $K = 3$, por lo que el nuevo registro, por mayoría de votos de estos tres valores, es un valor positivo o de los valores de las cruces.

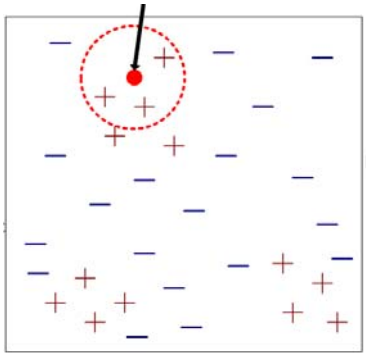


Figura 3.4. Clasificación del nuevo objeto cuando *k*=3.

Pero si el valor de *k* es distinto, la clasificación del nuevo registro puede ser distinta (*figura 3.5*):

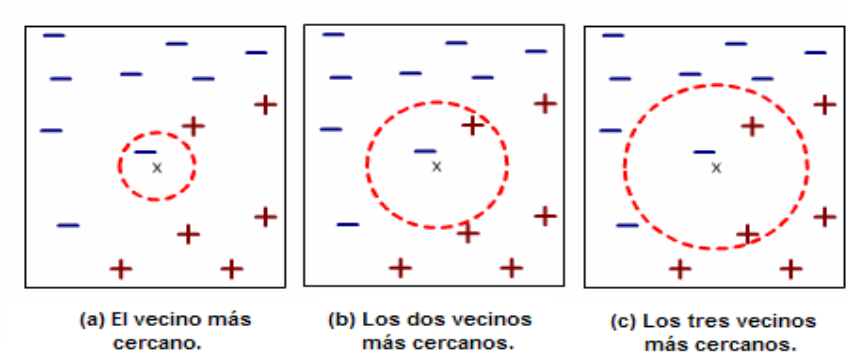


Figura 3.5. Clasificación del nuevo objeto dependiendo del valor de *k*. (15)

Para la figura anterior (*figura 3.5*), si *k* = 1, el nuevo registro se clasifica como un valor negativo. Si *k* = 2 el nuevo registro puede ser clasificado ya sea como valor negativo o positivo puesto que no existe ninguna mayoría de votos. Si *k* = 3, el nuevo valor se clasifica como un valor positivo por mayoría de votos. Por lo anterior, es conveniente elegir un valor de *k* impar. Pero también se observa que con *k* = 1 el resultado es distinto a que si *k* = 3. Una manera de decidir qué valor de *k* (impar) es más conveniente, es observar las medidas de evaluación como el alcance y la precisión. La precisión mide la probabilidad de que si un sistema clasifica un documento en una cierta categoría, el documento realmente pertenezca a la categoría. El alcance da una medida de cobertura o completitud.

Se utilizará este algoritmo en nuestro estudio debido a la gran capacidad que presentó para clasificar correctamente la mayoría de los casos.

La que resulte mejor, es la que entonces se acopla mejor a los resultados que se desean obtener.

Al momento de escoger el valor de *k* se debe tomar en cuenta también que si el valor es demasiado pequeño, la clasificación será sensible a valores de ruido. Y si es demasiado grande, se pueden escoger otros valores que pertenecen a otras clases.

IBk

El algoritmo *IBk* pertenece al grupo de algoritmos utilizados en las RNA y es de aprendizaje retardado porque no construye modelos explícitamente (es una caja negra), el aprendizaje lo deja al final por lo que necesita menos tiempo para el entrenamiento pero más tiempo para las predicciones y tiene una alta exactitud en las predicciones o clasificaciones.

El algoritmo funciona así:

1) Calcula la distancia Euclidiana del objetivo a aquellos que fueron muestreados.
2) Ordena las muestras tomando en cuenta las distancias calculadas.
3) Escoge heurísticamente la *k* más óptima con ayuda de la validación cruzada.
4) Calcula un promedio ponderado de distancia inversa con los vecinos multivariados *k* más cercanos.

Por lo que se puede decir que las redes neuronales artificiales son técnicas analíticas que permiten modelar el proceso de aprendizaje de una forma similar al funcionamiento del cerebro humano, básicamente, la capacidad de aprender a partir de nuevas experiencias.

Estas técnicas han tenido un desarrollo impresionante en la última década, con aplicaciones tanto a la medida como generales (comúnmente llamados shell) y tienen como objetivo fundamental sustituir la función de un experto humano.

Conclusiones del capítulo tres.

En el capítulo tres se muestran algoritmos que se utilizan en la minería de datos. Para el caso de reglas y patrones, se eligió el árbol de decisión (el cual está dentro del bosque aleatorio) por ser fácil de utilización y entendimiento. Para el caso de las predicciones, se mostraron dos algoritmos distintos que se pueden usar en las predicciones. Uno es el algoritmo de clasificación *IBk* basado en la cercanía de vecinos y, el otro, la red neuronal, basada en la estructura cerebral humana.

Se determinó considerar estos dos algoritmos dado que en *rapidminer* existen éstos y, por tanto, se investigó en la literatura y fuentes bibliográficas sobre su uso, su eficiciencia, cómo funcionan; después de esto, se consideraron muy factibles para que se obtuvieran buenos modelos a partir de nuestros datos. Para determinar cuál es el mejor para nuestro estudio, se llevó a cabo con la ayuda de las validaciones (las que determinan qué tan bien *aprende* el modelo generado por cada uno de estos dos algoritmos).

CAPÍTULO IV
"DESCRIPCIÓN DEL DESARROLLO PRÁCTICO DE LAS FASES CON UNO O MÁS HERRAMIENTAS DE MINERÍA DE DATOS"

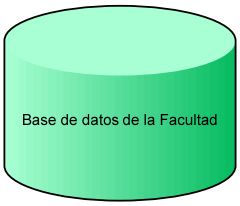
4.1 FASE 1 SISTEMAS DE INFORMACIÓN

En este capítulo se desarrolla de forma práctica el proceso del KDD. Se comenzará con el desarrollo de la primera fase del KDD: Sistemas de información (requisitos), por lo cual se plantearon las preguntas más frecuentes como son:

- **¿Qué se está buscando?** Encontrar patrones de desempeño en el avance escolar de los alumnos.
- **¿Qué atributo del conjunto de datos se desea intentar predecir?** El atributo primordial es la deserción con base en los historiales académicos solamente, dado que no se cuenta con otras fuentes, tales como las socioeconómicas ni demográficas.
- **¿Qué tipos de relaciones se intenta buscar?** Se buscan patrones o reglas y modelos que muestre quién es factible que deserte solamente con base en los historiales académicos. Esto ayudaría a identificar a los alumnos que muy probablemente desertarán. Ir en contra de la predicción. Cada alumno cuesta dinero y demás; dejar que deserte es un desperdicio.
- **¿Desea realizar predicciones a partir del modelo de Minería de Datos o sólo buscar asociaciones y patrones interesantes?** En este caso se aplicarán los dos tipos de predicciones.
- **¿Cómo se distribuyen los datos?** Se tienen en una sola base de datos.
- **¿Cómo se relacionan las columnas? O en caso de haber varias tablas, ¿cómo se relacionan las tablas?** Con el número de cuenta del alumno.

Vista del procedimiento para llevar a cabo la *minería de datos*.

Definición del problema

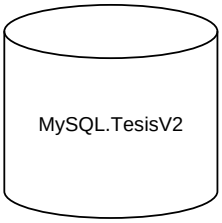


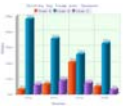
Se proporciona Archivos .sql

copia de la base de datos en y en archivos en Excel.

Almacenamiento de los datos a MySQL

Se crea un *script* para la migración de los datos.





Exploración de los datos

Preparación y limpieza de los datos

Minería de Datos usando *Rapidminer*

Patrones y modelos

Evaluación e Interpretación

Difusión del Conocimiento

Figura 4.1. Procedimiento de nuestro estudio de minería de datos.

Cabe resaltar que se delimitará a estudiar dichos criterios de deserción en los planes 1994 y 2006, ya que la copia de la base de datos que se proporcionó contenía registros de otros planes de estudio.

Una vez establecido qué es lo que se está buscando y las delimitaciones, se procede a revisar la base de datos con la que se cuenta; para conocer las tablas que servirán y las que harán falta para el proceso. En la *Figura 4.2* se muestran las tablas que contienen nuestra base de datos inicial.

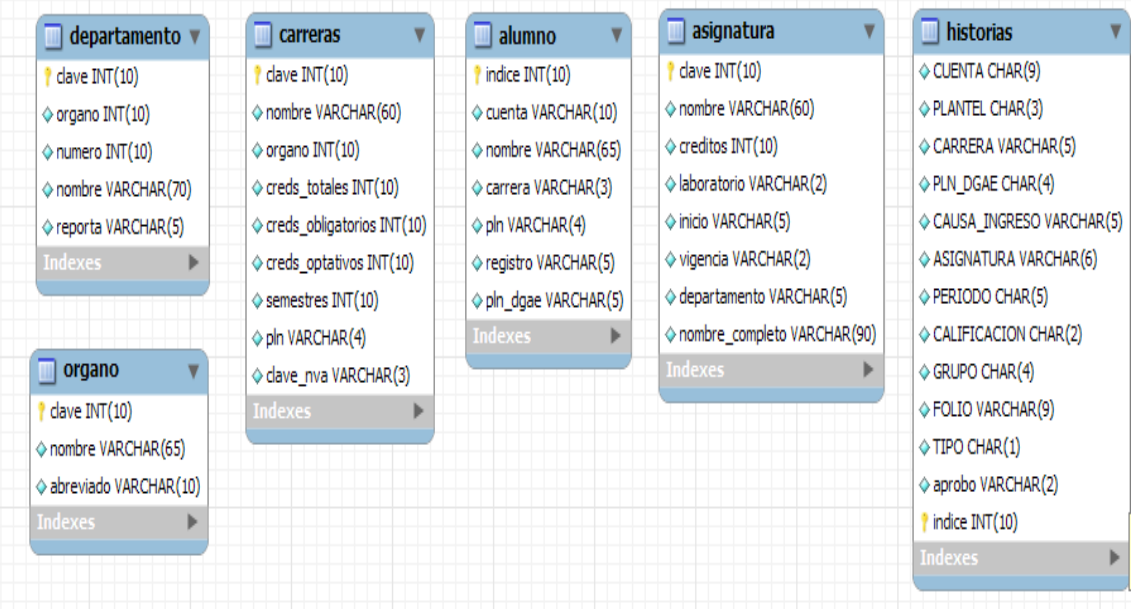


Figura 4.2 Tablas que conforman nuestra base de datos

Se empezará la establecer los criterios para lograr determinar si un alumno desertó o no

Criterios:

- Un alumno desertó si el número de créditos en su registro es menor al número de créditos requeridos para terminar sus estudios y que ya no tenga registros de calificaciones en el semestre anterior.
- Si el alumno tiene menos créditos de los que se requieren para terminar sus estudios pero tiene registros de calificaciones en el semestre anterior, significa que el alumno no ha desertado puesto que sigue cursando su carrera.

Cabe mencionar además otros detalles: existen registros de alumnos en los que, para terminar totalmente sus estudios, les faltan 3 créditos o menos. Esto se debe probablemente a que haya materias de 4 créditos o errores en el cálculo de créditos para terminar una carrera. Por lo que si a un alumno le faltan 3 créditos, ya se puede considerar que el alumno sí terminó sus estudios. Por lo que éste se clasificaría como un alumno que sí terminó sus materias. Cabe recalcar que, con los datos proporcionados, no se puede saber si el alumno se tituló o no. Al menos no desde el conjunto de Datos que se tiene.

Resumiendo lo que se va hacer:

1) Si el alumno tiene más de los créditos mínimos, desertó = 'NO' (es decir, que terminó todas sus materias).

2) Si el alumno tiene menos de los créditos mínimos, pero tiene registro de calificaciones del semestre pasado, desertó = 'NO'. Esto quiere decir que el alumno está inscrito aún. Por lo que todavía no se puede considerar que ya desertó.

3) Si el alumno tiene menos de los créditos mínimos y además ya no tiene registros de calificaciones de este último semestre; entonces desertó = 'SÍ'.

Primero se procede a realizar una consulta o *query* en la que se desea dividir a los alumnos en dos grupos: los que ya tienen más de los créditos necesarios y a los que todavía les falta. Además esto deberá hacerse para cada carrera, ya que el número de créditos de cada carrera requerido es diferente.

En la Página http://www.ingenieria.unam.mx se obtuvieron los créditos que requiere cada carrera y se obtuvierón los planes DGAE (*pln_dgae*) de los dos planes – 1994 y 2006, que son las claves con las que se identifican las carreras y se realizó una nueva tabla para nuestra base de datos agregando el atributo *status_plan*: 1 si es plan 1994 y 2 si es plan 2006. Esto con la finalidad de hacer más simple la clasificación de ¿Qué plan DGAE corresponde a qué plan de estudios? (Véase la *tabla 4.1* en el Anexo al final de este trabajo).

Cabe aclarar que ciertas columnas se repiten en varias tablas, por lo cual se podría decir que las tablas no están normalizadas. Hay que recordar que el contenido está en una Base de Datos del tipo OLAP (*Online Analytical Processing*) cuyas características particulares es que en ellas se llevan a cabo mayoritariamente consultas para el análisis de datos y búsqueda de patrones con el fin de optimizar las consultas que se desean; por lo que conviene frecuentemente tener casi las mismas columnas en distintas tablas.

Hasta este punto puede parecer que fue una tarea demasiado sencilla, pero no fue así; ya que las tablas que proporcionaron fueron en archivos de *Excel* que tuvieron que cargarse a las tablas con ayuda de la herramienta para cargar los 1.2 millones de datos que proporcionaron.

PhpMyAdmin provee un ambiente sencillo y práctico para empezar a cargar los datos y generar consultas. Sin embargo, el equipo de cómputo con el que se cuenta para hacer dicho estudio no posee mucha memoria ni tanta rapidez en procesador: cargar cada tabla por medio de *PHPMyAdmin* implicó varias horas, incluso días. Para el caso de la tabla de los historiales (académicos), se realizaron varias particiones de datos para poder cargar cada una a la base de datos ya que fue imposible cargar todos los datos en una sola vez. Después se realizó la selección de los planes dgae que pertenecieran a los planes de estudio 1994 y 2006, por lo que se redujo la cantidad de registros a aproximadamente 600 000 mil registros.

Para mayor detalle sobre el *software* (o programas) utilizado, consultar el apéndice I.

4.2 FASE 2. PREPARACIÓN DE LOS DATOS.

El primer paso que se realizó de la preparación de los datos fue quitar aquellos registros de los planes de estudio anteriores a 1994.

Aunque la teoría marca que, en primera instancia, la *vista minable* debe quedar absolutamente limpia, es cierto que en la práctica no es del todo así ya que se van encontrando inconsistencias. También es cierto que el proceso de preparación conlleva a un proceso cíclico de limpieza, integración y transformación.
Con las tablas de nuestra base de datos no se podía sacar alguna información relevante porque no contenían todos los atributos necesarios para que fueran relacionados e iniciar a construir de la *vista minable*.

Se procedió a crear la vista minable *alumnodesercion* que contuviera los siguientes atributos:

1) *cuenta* varchar(10). Contiene el número de cuenta.
2) *causa_ingreso* varchar(3). Contiene la clave del tipo de ingreso del alumno a la facultad.
3) *plan_dgae* varchar(4). Contiene la clave del plan de estudios del alumno.
4) *creditos* integer. Contiene el número de créditos que tiene el alumno.
5) *periodos* integer. Contiene el número de períodos o semestres que ha cursado el alumno.
6) *primerperiodo* varchar(5). Contiene el primer semestre que cursó el alumno.
7) *ultimoperiodo* varchar(5). Contiene el último semestre que ha cursado hasta ahora. Éste período se utiliza para saber si por lo menos el alumno sigue estando inscrito (para saber si ya desertó o no).
8) *generacion* varchar(4). Contiene el año en que el alumno inició la carrera.
9) *deserto* varchar(2). Se marca si el alumno desertó o no.
10) *terminomaterias* varchar(2). Se marca si el alumno terminó todas sus materias de la carrera o no.
11) *promedio* float(4,2). Contiene el promedio del alumno.
12) *hareprobado* integer. Indica el número de veces que ha reprobado el alumno a lo largo de la carrera.
13) *haaprobado* integer. Indica el número de materias que ha aprobado el alumno a lo largo de la carrera.
14) *duplicidad* integer. Se usa para saber qué números de cuenta tienen duplicados, ya que existe el caso de que algunos números de cuenta tienen dos o tres planes de estudio, lo cual significa que se cambiaron de plan. Cabe aclarar que, los alumnos que cambiaron de plan, se les hizo una revalidación de estudios en las materias que sólo eran compatibles (que tuvieran como mínimo el

80% del contenido del programa de las materias del nuevo plan). Ahora para los nuevos números de cuenta, su plan de estudios DGAE cambia según la especialidad que tomen. Por ejemplo, un alumno de la carrera de Ingeniería en Computación del nuevo plan 2006 cuya clave es 1190, tendrá registrado ese plan. Al momento de seleccionar el módulo de Bases de Datos, su plan de estudios DGAE cambiará por 1193. En su historial académico tendrá ambas claves. Es por eso que las cuentas se duplican o triplican. Es por eso que al final se analizarán todos los datos incluyendo los números de cuenta duplicados. Esto ha servido para entender el porqué se duplican. Sabiendo esto, se toma la decisión de dejarlos así. Existen casos en los que cuando un alumno ha seleccionado un módulo de salida, el plan DGAE en su historial se actualiza en todo su historial por lo que deja sin la posibilidad de saber con qué plan de estudios DGAE comenzó el estudiante. En este campo, si la cuenta resulta duplicada se asigna el valor de 1; en caso contrario, un 0.

15) *indice* integer. Es la llave primaria de la tabla.

Dado que *PHPMyAdmin* es sólo una aplicación Web para consultar y administrar la base de datos, en la construcción de la tabla alumnodeserción ya no es tan práctico seguir utilizando únicamente esta herramienta, por lo que se empezará a trabajar con *MySQL Query Browser* el cual realiza y procesa las consultas mucho más rápido.

El script que crea la vista minable *alumnodesercion* se muestra a continuación (usando *MySQL*):

```
CREATE TABLE `alumnodesercion` (
  `indice` int(10) unsigned NOT NULL auto_increment,
  `cuenta` varchar(10) NOT NULL,
  `causa_ingreso` varchar(3) default NULL,
  `pln_dgae` varchar(4) default NULL,
  `plan` int(4) default NULL,
  `creditos` int(10) unsigned default NULL,
  `periodos` varchar(5) NOT NULL,
  `primerperiodo` varchar(5) default NULL,
  `ultimoperiodo` varchar(5) default NULL,
  `generacion` varchar(4) default NULL,
  `deserto` varchar(2) NOT NULL,
  `terminomaterias` varchar(2) default NULL,
  `promedio` float(4,2) default NULL,
  `hareprobado` int(3) default NULL,
  `haaprobado` int(3) default NULL,
  `duplicidad` int(10) unsigned default '0',
  PRIMARY KEY  USING BTREE (`indice`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 ROW_FORMAT=DYNAMIC COMMENT='tabla que contiene datos sobre
desercion o no de alumnos';
```

Luego se requiere llenar los datos a esta nueva tabla. Para ello, se programó 12 procedimientos. A continuación, se muestra una breve explicación de cada uno de ellos:

- *alumnodesercion.sql*. Este procedimiento obtiene los datos para llenar los atributos cuenta, causa_ingreso, pln_dgae, plan, creditos, periodos, primerperiodo, ultimoperiodo, deserto.
- *desercion_null.sql*. Debido a que existen alumnos que no tienen créditos, es decir, no han aprobado ninguna materia aún, hace que para estas cuentas, los atributos de primerperiodo, ultimoperiodo y creditos queden nulos.
- *ultimoperiodo_check.sql*. Debido a la existencia de casos nulos, el último período que se registra para algunas cuentas sale defasado un semestre antes. Entonces hay que corregir esos datos. Después, verificar manualmente si por esta causa, se indica si un alumno desertó o no erróneamente.
- *duplicidad.sql*. Sirve para detectar números de cuenta duplicados. Esto se debe a que los alumnos cambian de carrera o debido a que coexisten dos planes de estudio, los del plan viejo deben tomar materias del plan nuevo porque en su plan estas materias ya no se dan, lo que provoca que en nuestro análisis se dupliquen las cuentas.
- *eliminar_duplicados.sql*. Elimina las cuentas duplicadas en donde el plan de estudios sólo representa haber tomado una materia o si fue producto de las materias tomadas en el anexo o antes de tomar un módulo de salida.
- *generaciones.sql*. Determina la generación a la que pertenece cada cuenta según el primer período en el que fue inscrito.
- *terminomaterias.sql*. Determina si el alumno terminó todas sus materias o no según el número mínimo de créditos que exige cada carrera para terminar.
- *cuantasreprobadas.sql*. Obtiene el número de veces que el alumno ha reprobado a lo largo de su estancia.
- *cuantasaprobadas.sql*. Calcula el número de materias que el alumno ha aprobado.
- *promedio_des_cursan*.sql. Calcula el promedio de los alumnos que están cursando y de aquellos que ya han desertado.
- *promedio_zero_creds.sql*. Obtiene el promedio de aquellos alumnos que no tienen créditos. Determina si tienen promedio de cero o 5.

- *promedio_termino.sql*. Determina el promedio de los alumnos que ya terminaron todas sus materias.

De los procedimientos anteriores, el primero crea la tabla (*create*), los procedimientos 2, 3, 4, 6, 7, 8, 9, 10, 11 y 12 son de actualización (*update*) y el 5 es de eliminación (*delete*).

En lo siguiente, se muestra cómo y en qué *software* se programaron los procedimientos.

Para programar el procedimiento *1) alumnodesercion*, el algoritmo es el siguiente:

a) Obtener las carreras con su respectivo número de créditos:

```
SELECT pln_dgae,creditos,status_plan FROM carreras_planes
ORDER BY pln_dgae;
```

b) Para una carrera dada, obtener los números de cuenta que existen (de la tabla historias) y, de paso, la causa de ingreso:

```
SELECT DISTINCT cuenta,causa_ingreso FROM historias
WHERE pln_dgae=plan_temp;
```

c) Para cada número de cuenta, contar el número de semestres que ha cursado:

```
SELECT COUNT(DISTINCT periodo) FROM historias
WHERE cuenta=cuenta_;
```

d) Ahora, para ese número de cuenta, obtener su número de créditos, el último semestre cursado *(max(periodo))* y el primer semestre cursado (*min(periodo)*):

```
SELECT MIN(DISTINCT h.periodo) AS primerperiodo, MAX(DISTINCT h.periodo) AS
ultimoperiodo,SUM(a.creditos) AS creditos
FROM historias h, asignatura a
WHERE a.clave=h.asignatura
AND h.aprobo='SI'
AND cuenta=cuenta_;
```

e) Finalmente, determinar si el alumno desertó o no utilizando la sentencia *if* dentro del procedimiento y determinar si es del plan 1994 o de los nuevos planes.

Como se dijo se empezara a utilizar el programa *MySQL Query Browser* el cual sirve para hacer consultas a la Base de Datos y programación de procedimientos, disparadores y funciones.

Teniendo esto, el procedimiento se crea (Usando *MySQL query browser*. *Figura 4.3*):

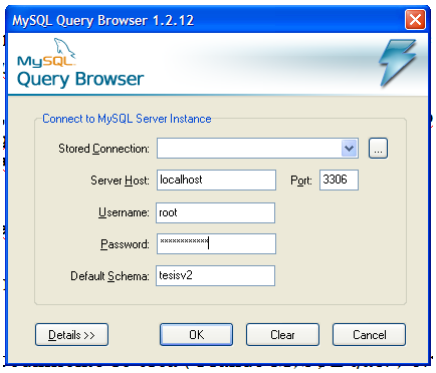


Figura 4.3. Entrando al programa *MySQL Query Browser*.

Se abre el programa tecleando los parámetros que se piden. *Server host* es el nombre del servidor. *Username* y *password* son el usuario y la contraseña respectivamente y *default schema* es el nombre de la base de datos. Dar clic en *OK*. Véase la *figura 4.3*.

Estando dentro del programa, seleccionar del menú principal *script*, *create stored procedure* (*véase la figura 4.4*):

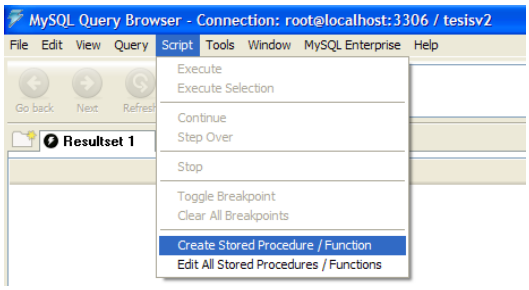


Figura 4.4. Abriendo una ventana para crear un nuevo procedimiento.

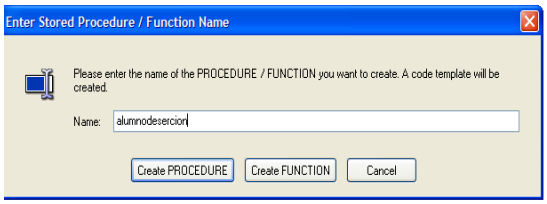


Figura 4.5. Tecleando el nombre del nuevo procedimiento.

Y se pedirá darle un nombre al procedimiento. En este caso teclear *alumnodesercion* y dar clic en *createprocedure*. Véase la *figura 4.5*. El programa genera automáticamente el código básico del procedimiento. Véase la *figura 4.6*.

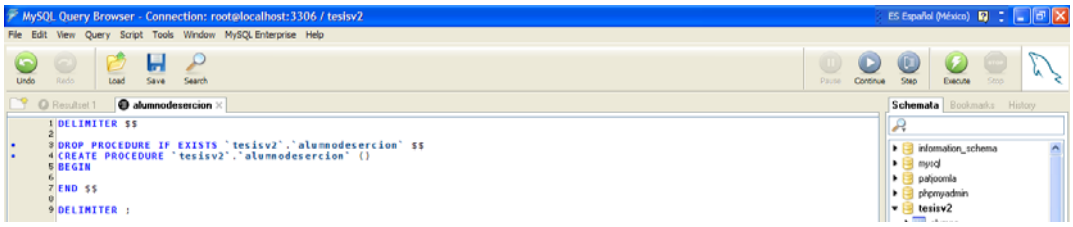


Figura 4.6. Se abre un nuevo procedimiento y se muestra automáticamente el código básico para comenzar a programar.

El procedimiento (*alumnodesercion.sql*) queda como se muestra en el anexo. Al final de este documento.

Terminado esto, se procede a guardar el procedimiento (*figura 4.7*). El nombre que se elige en este caso es *alumnodesercion.sql* (Véase *figura 4.8*). Después verificar que el procedimiento no tenga errores: dar clic en *continue*. Si todo está bien, el procedimiento aparecerá como tal en la ventana de la derecha (Véase la *figura 4.9*).

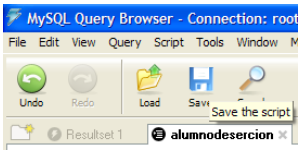


Figura *4.7*. Guardando el procedimiento.

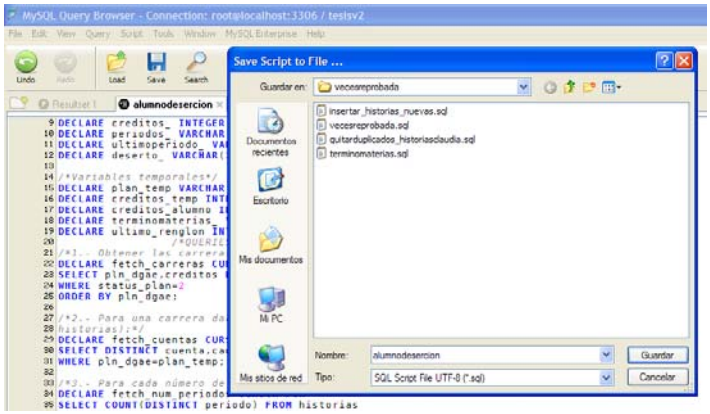


Figura *4.8*. tecleando el nombre del procedimiento para guardarlo en la ruta especificada.

Para correr el procedimiento, sólo dar doble clic en él (donde está en la ventana de la derecha) y finalmente dar clic en *Execute* y esperar a que termine de ejecutarse el procedimiento. Véase la *figura 4.10*.

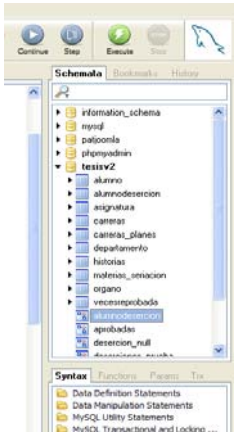


Figura *4.9*. Se da *clic* en *continue* y se observa que el procedimiento ya está listo para ejecutarse.

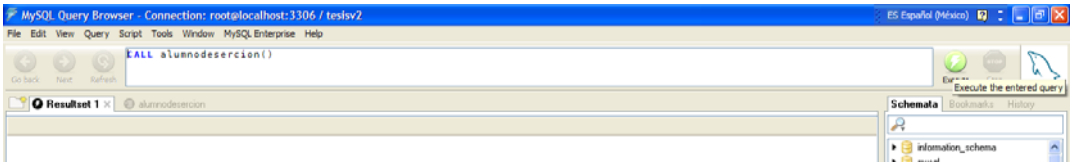


Figura 4.10. Ejecutando el procedimiento.

Los datos se van llenando en la tabla *alumnodesercion*. Como se puede observar, en el procedimiento se da por hecho que *todos* los alumnos llegan a pasar sus materias, que *todos* los alumnos tienen créditos en sus historiales. Pero resulta que salen algunos registros con valores nulos (*null*) en las columnas del último semestre cursado *(ultimoperiodo)* y en la columna *creditos*. ¿Por qué?

La razón, analizando estos alumnos en particular, es que no han pasado ninguna materia y es por eso que estas columnas tienen valores nulos. Entonces se decidió crear un nuevo procedimiento que se ocupe de estos valores nulos en el que actualice el valor nulo por el valor de cero en la columna de *creditos* (ya que no han aprobado ninguna materia) y que obtenga el último semestre cursado de estos alumnos.

Se hace el procedimiento 2) *deserción_null.sql* de la misma manera que se hizo el procedimiento *alumnodesercion.sql* (par ver el código fuente, véase el anexo al final de este documento).

El problema ahora es que el último período para algunos casos no es el que es en realidad. Esto se debe a que, cuando se hace una consulta si el alumno reprueba todas en el último período, se obtiene como el último semestre cursado aquél en el que se aprobaron materias. Claro, se pudo esto programar, pero se requería de más código anidado y sería más susceptible a errores. Sale menos laborioso realizar un procedimiento que sólo corrija estos detalles. Se creó el procedimiento 3) *ultimoperiodo_check.sql* (véase el anexo al final de este documento).

Este detalle también provocó que se haya marcado a algunos alumnos sin haber desertado aún y que el procedimiento los haya marcado como alumnos que desertaron (de acuerdo a los criterios tomados), por lo que se ejecuta la siguiente actualización:

```
UPDATE alumnodesercion SET deserto='NO'
WHERE deserto='SI'
AND ultimoperiodo='20072' /*según haya sido el último semestre*/;
```

Ahora hay que llenar la columna de *terminomaterias* para determinar si el alumno (por cada carrera) terminó todas sus materias. Para hacerlo, se programa un *procedimiento 4)* que determina si el alumno terminó sus materias o no. Se vuelve a hacer lo mismo:

Primero se crea un procedimiento que se llame *terminomaterias* (véanse las *figura 4.10b* y *4.11*).

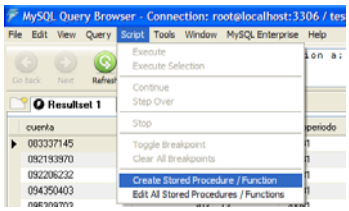


Figura 4.10b. Seleccionando del menú la creación del procedimiento.

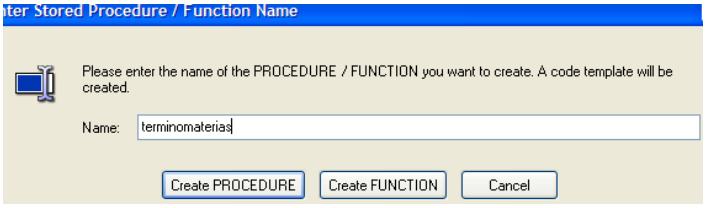
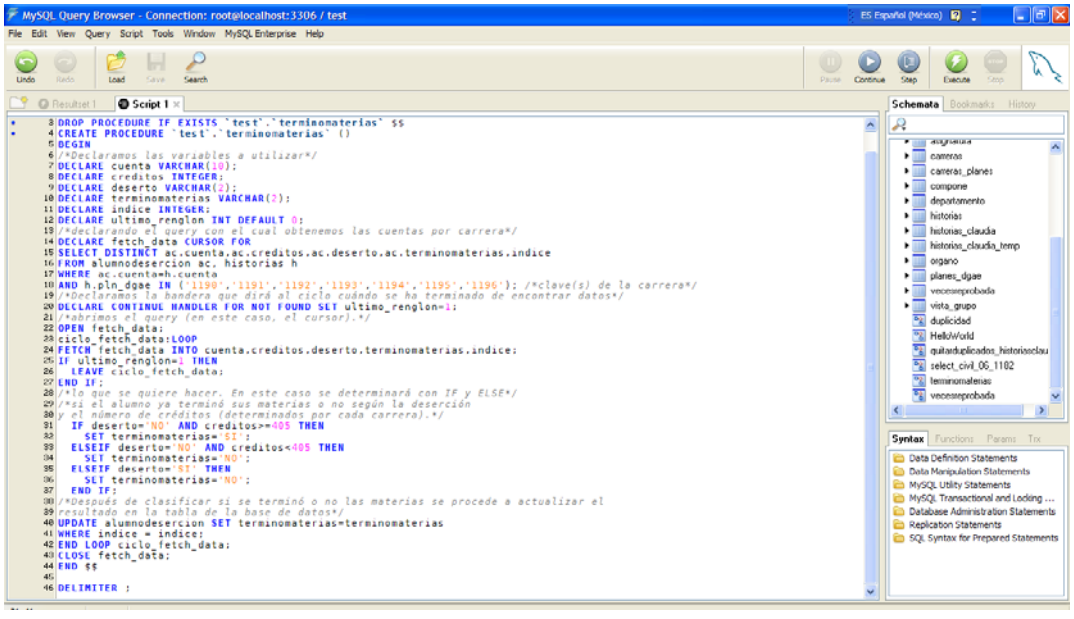


Figura 4.11. Nombrando el procedimiento.

Para ver el código fuente del procedimiento *Terminomaterias.sql*, véase el anexo al final de este documento.

Se guarda el procedimiento con el nombre de *terminomaterias.sql* y luego se da clic en *continue* para comprobar si tiene errores y al mismo tiempo para compilarlo e integrarlo a la lista de procedimientos. *Figura 4.12.*




```
DROP PROCEDURE IF EXISTS `test`.`terminomaterias` $$
CREATE PROCEDURE `test`.`terminomaterias` ()
BEGIN
/*Declaramos las variables a utilizar*/
DECLARE cuenta VARCHAR(10);
DECLARE creditos INTEGER;
DECLARE deserto VARCHAR(2);
DECLARE terminomaterias VARCHAR(2);
DECLARE indice INTEGER;
DECLARE ultimo_renglon INT DEFAULT 0;
/*Declarando el query con el cual obtenemos las cuentas por carrera*/
DECLARE fetch_data CURSOR FOR
SELECT DISTINCT ac.cuenta,ac.creditos,ac.deserto,ac.terminomaterias,indice
FROM alumnodesercion ac, historias h
WHERE ac.cuenta=h.cuenta
AND h.pln_dgae IN ('1190','1191','1192','1193','1194','1195','1196'); /*clave(s) de la carrera*/
/*Declaramos la bandera que dirá al ciclo cuándo se ha terminado de encontrar datos*/
DECLARE CONTINUE HANDLER FOR NOT FOUND SET ultimo_renglon=1;
/*Abrimos el query (en este caso, el cursor).*/
OPEN fetch_data;
ciclo_fetch_data:LOOP
FETCH fetch_data INTO cuenta,creditos,deserto,terminomaterias,indice;
IF ultimo_renglon=1 THEN
  LEAVE ciclo_fetch_data;
END IF;
/*lo que se quiere hacer. En este caso se determinará con IF y ELSE*/
/*si el alumno ya terminó sus materias o no según la deserción
y el número de créditos (determinados por cada carrera).*/
  IF deserto='NO' AND creditos>=405 THEN
    SET terminomaterias='SI';
  ELSEIF deserto='NO' AND creditos<405 THEN
    SET terminomaterias='NO';
  ELSEIF deserto='SI' THEN
    SET terminomaterias='NO';
  END IF;
/*Después de clasificar si se terminó o no las materias se procede a actualizar el
resultado en la tabla de la base de datos*/
UPDATE alumnodesercion SET terminomaterias=terminomaterias
WHERE indice = indice;
END LOOP ciclo_fetch_data;
CLOSE fetch_data;
END $$

DELIMITER ;
```

Figura 4.12. Editando, guardando y verificando la sintaxis del procedimiento en cuestión.

Hecho esto, se procede a ejecutarlo tecleando doble clic en el procedimiento (de la lista de procedimientos de la ventanita del lado derecho. Cuando se hace esto, en el cuadro de texto aparecerá *call terminomaterias()* lo cual indica que se va a llamar al

procedimiento para su ejecución. Entonces se da clic en *execute* y el procedimiento se ejecutará diciendo *query is being executed* (la consulta se está ejecutando). Cuando termina, en la ventana de resultados aparecerá un mensaje indicando *no resultset returned* lo cual no devuelve ningún resultado; sólo ejecutó, en este caso, el procedimiento (*figura 4.13*).

Finalmente se procede a llenar la columna que queda de la tabla: *duplicidad*, con la cual se quiere saber si hay números de cuenta que están duplicados, es decir, que si están registrados con otros planes de estudios. El procedimiento 5), *duplicidad.sql*, se muestra a detalle en el anexo al final de este documento.

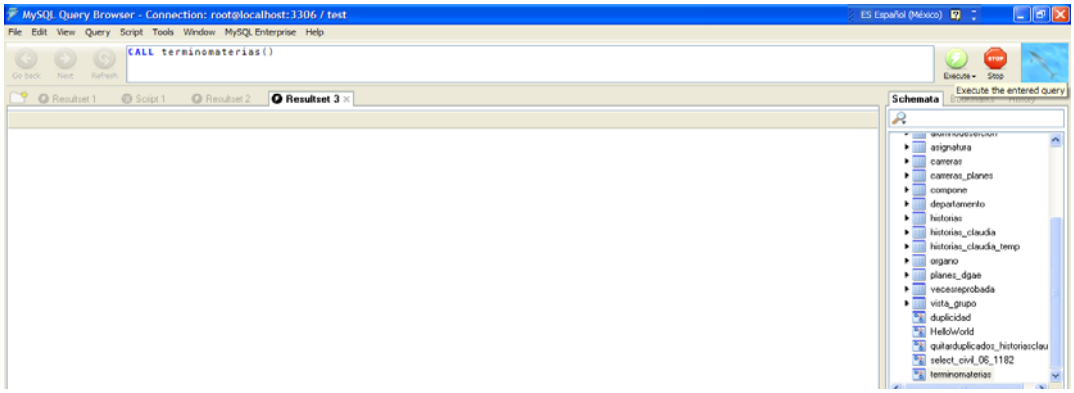


Figura 4.13. Ejecución del procedimiento. *Call terminomaterias()*.

Al igual que en los procedimientos anteriores, éste código se copia en el editor de procedimientos, se guarda y se da clic en *continue* para su revisión y después se ejecuta.

Después de haber encontrado cuentas que se duplican, se procede a eliminarlas con el procedimiento 6), *eliminar_duplicados.sql*, el cual está a detalle en el anexo al final de este documento.

Ahora se desea saber la generación a la que pertenece cada alumno con el procedimiento 7), *generaciones.sql*, el cual se encuentra en el anexo al final de este documento.

Se necesita también llenar las columnas de cuántas veces ha reprobado y cuántas materias ha pasado cada alumno con el procedimiento 8, *cuantasreprobadas.sql*, el cual se encuentra en el anexo al final de este documento.

Continuando con el procedimiento 9) *cuantasaprobadas.sql*, el cual se encuentra en el anexo al final de este documento.

Por último se tienen los procedimientos 10, 11 y 12 para obtener los promedios de calificaciones de los alumnos. Para hacerlo se siguen los siguientes pasos:

1) Seleccionar los números de cuenta:

```
SELECT DISTINCT cuenta FROM alumnodesercion
WHERE deserto='SI'
```

2) seleccionar distintamente la materia

```
SELECT DISTINCT asignatura FROM historias
WHERE cuenta=cuenta_
AND calificacion NOT IN ('NP','AC','RE')
```

Para este caso, hay alumnos que no aprobaron ninguna por lo que el resultado que se obtiene es *0 rows fetched* o cero renglones. Hay que comprobar primero con un *if*, si este es el caso. Al salir de este ciclo, hay que actualizar el promedio poniendo como valor 0.

3) Para cada materia seleccionar aquel registro cuyo semestre sea el más reciente. Luego, de esas materias, obtener el promedio sin tomar en cuenta las calificaciones nominales, sólo las numéricas.

```
SELECT MAX(periodo) AS periodo, calificacion FROM historias
WHERE cuenta=cuenta_
AND asignatura=asignatura_
AND calificacion NOT IN ('NP','AC','RE')
GROUP BY calificacion
```

4) Se obtiene el promedio (manualmente, usando un contador y un sumador de las calificaciones).

5) Se actualizan los promedios obtenidos.

Entonces se programan tres procedimientos. El primero calcula el promedio de aquellos que ya terminaron sus materias. El segundo obtiene los promedios de los alumnos que están cursando y el último sólo determina si se pone promedio de 5 o de 0 a los alumnos que no han acreditado ninguna materia. Esto último depende si tienen calificaciones numéricas como 5. Ya que las calificaciones en letra como *NP* no promedian. Estos tres procedimientos se muestran a detalle en el anexo al final de este documento (*promedio_termino.sql*, *promedio_des_cursan.sql* y *promedio_zero_creds.sql*. Procedimientos 10, 11 y 12 respectivamente).

Finalmente se puede revisar nuestra tabla *alumnodesercion* con los nuevos datos obtenidos con una consulta. En este caso usando *MySQL Query Browser* (véase la *figura 4.14*).

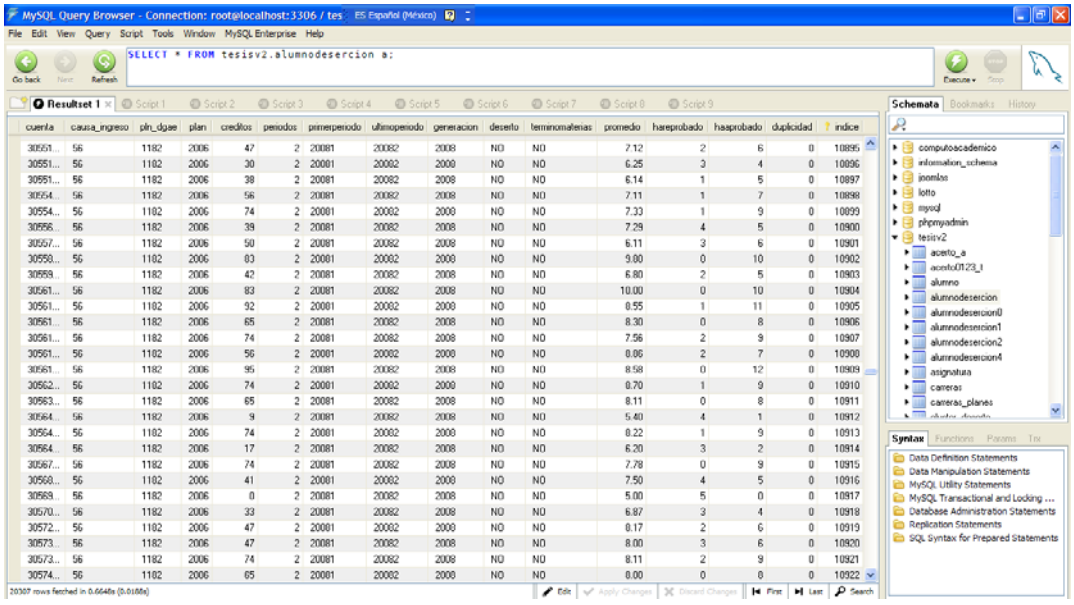


Figura 4.14. Abriendo el contenido de la nueva tabla *alumnodesercion*.

Revisiones:

Se va a revisar ahora que en los datos que se han generado no haya inconsistencias. Por ejemplo, se va a verificar para los alumnos que tienen todos sus créditos, efectivamente se les haya marcado con que sí concluyeron y que no desertaron.

Por cada carrera se emplea el siguiente *query* o consulta; por ejemplo, empezando por la carrera de ing. Geomática cuya clave es 1181 y cuyo número de créditos son 353:

```
SELECT * FROM alumnodesercion
WHERE pln_dgae='1181'
AND creditos >=(353-3);
```

Después de revisar los datos para cada carrera, se ha constatado que los datos son consistentes por lo que ahora se procede a obtener datos desde un punto de vista general hasta un punto de vista particular.

¤ Para el punto de vista general, se obtienen todos los datos de la tabla deserciones y se realiza su análisis estadístico más adelante.

Se desea ahora ir viendo el progreso de las generaciones. Se han recibido los datos desde el semestre 2007-1 y se tiene hasta el semestre 2009-1. De esta manera, se crean 3 tablas más de deserciones: *alumnodesercion0*, *alumnodesercion*1, *alumnodesercion*2 y *alumnodesercion* 4.

La primera *alumnodesercion0*, contiene los resultados hasta el semestre 2007-1 y así sucesivamente hasta llegar a la tabla *alumnodesercion4* que contiene los resultados hasta el semestre 2009-1. Por lo que todo el procedimiento descrito en lo anterior para crear la tabla, se repitió 3 veces cuidando solamente hasta qué semestre obtener los datos de la tabla *historias*. Por ende el proceso de construcción de esta tabla generó mucho esfuerzo y tiempo; y, tal como lo marca la teoría, se puede decir que sí es aproximadamente un 80% de esfuerzo el que se le tiene que aplicar en esta fase del KDD.

Cabe mencionar que no sólo se construyó esta tabla, se construyó otra llamada *vecesreprobada*. Su realización se realizó de la misma forma por medio de procedimientos la cual sí es útil para ser vista minable y sí contribuyó en la parte de visualización e interpretación.

```
SELECT * FROM alumnodesercion4;
```

De este *query* o consulta se obtienen 22460 registros los cuales son todos los alumnos (los que desertaron, los que están cursando y los que ya terminaron todas sus materias) desde la generación 1994. Sólo con las estadísticas se tendrá un mejor panorama general de la información que se tiene en esta tabla. Más adelante se explica cómo se obtienen estas estadísticas usando el programa estadístico *SPSS* (versión) *15*.

¤ Para casos más particulares se puede, por ejemplo, analizar primero el grupo de alumnos que no desertaron. Es decir, *deserto='NO'*:

```
SELECT pln_dgae,creditos,periodos,terminomaterias,deserto
FROM alumnodesercion4
WHERE deserto='NO';
```

Se obtienen 13496 registros (los cuales son aquellos alumnos que están cursando y los que ya terminaron todas sus materias). Véase la *figura 4.15*:

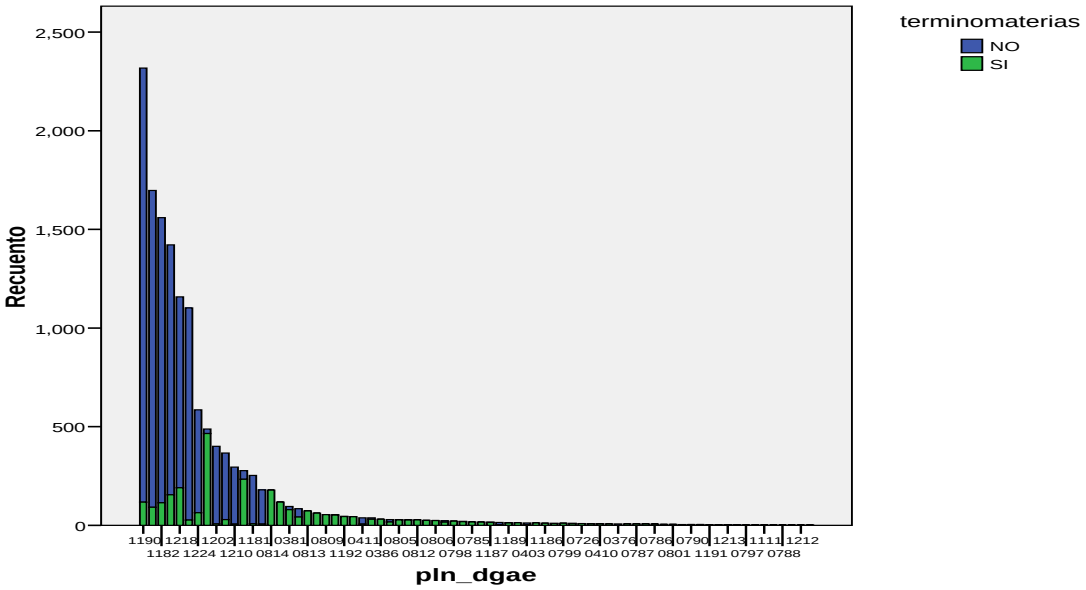


Figura 4.15. Gráfica del número de estudiantes por plan de estudios apilados por si terminaron materias o si siguen cursando.

¤ Ahora otro caso más particular es obtener el grupo de alumnos que no desertaron y que ya terminaron sus materias. Es decir, *deserto='NO'* y *terminomaterias='SI'*:

```
SELECT pln_dgae,creditos,periodos,terminomaterias,deserto
FROM alumnodesercion4
WHERE deserto='NO'
AND terminomaterias='SI';
```

Se obtienen 2746 registros (de aquellos alumnos que ya terminaron todas sus materias). La consulta evidentemente sería igual si se omite *WHERE deserto='NO'*. Véase la *figura 4.16*:

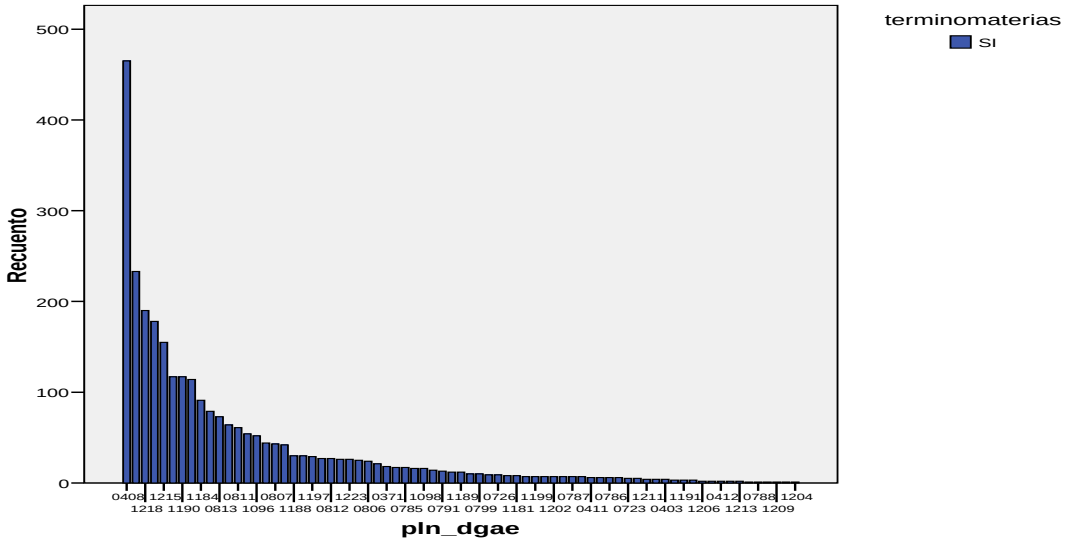


Figura 4.16. Gráfica de los estudiantes que sí terminaron todas sus materias por plan de estudios.

¤ Luego otro caso particular sería lo contrario: los alumnos que sí desertaron. Es decir, *deserto='SI'*:

```
SELECT pln_dgae,creditos,periodos,terminomaterias,deserto
FROM alumnodesercion4
WHERE deserto='SI';
```

Se obtienen 8964 registros (los cuales desertaron para el nuevo plan 2006). Véase la *figura 4.17*:

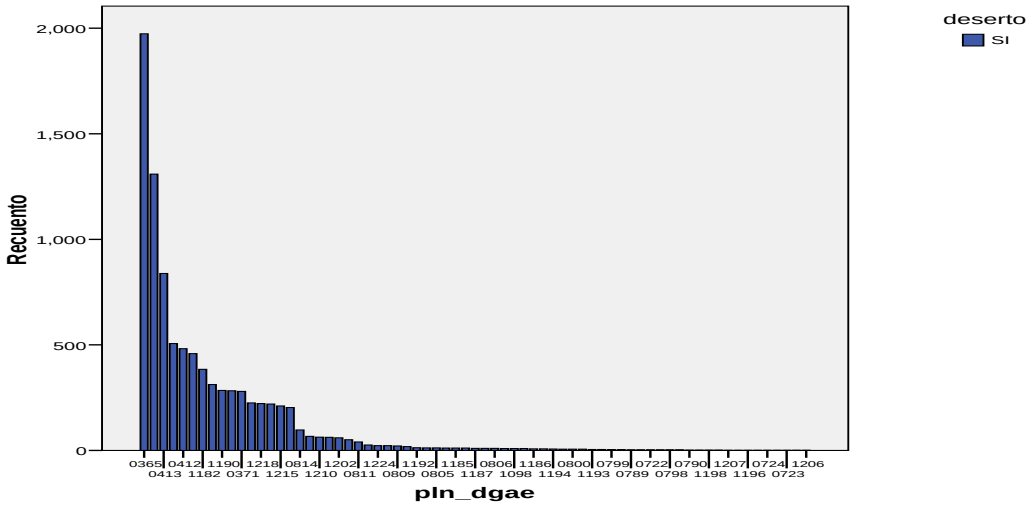


Figura 4.17: Figura de los alumnos que han desertado por plan de estudios.

¤ Y finalmente donde el alumno no ha desertado y no ha terminado sus materias (el alumno está cursando). Es decir, *deserto='NO'* y *terminomaterias='NO'*.

```
SELECT pln_dgae,creditos,periodos,terminomaterias,deserto
FROM alumnodesercion4
WHERE deserto='NO'
AND terminomaterias='NO';
```

Se obtienen 10750 registros (de los alumnos del nuevo plan 2006 que están cursando). Véase la *figura 4.18*:

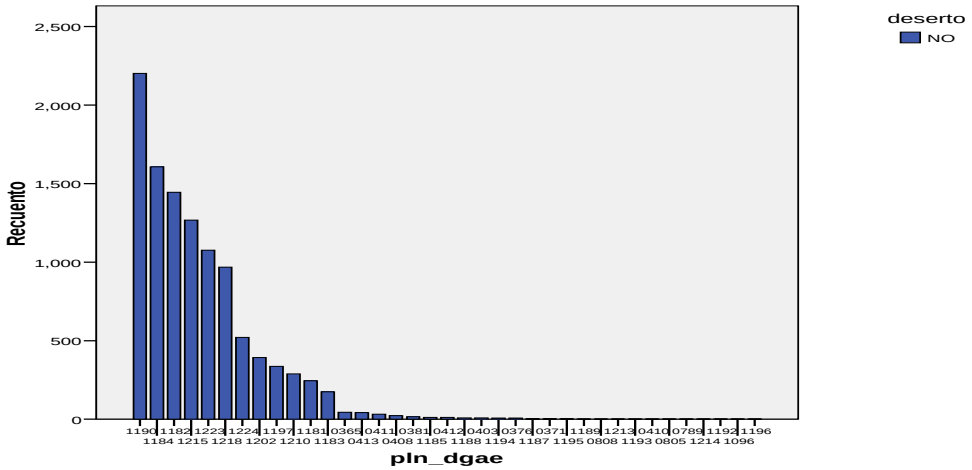


Figura 4.18: Gráfica de los alumnos que están cursando por plan de estudios.

Estos *queries* o peticiones del caso general y de los casos particulares que se acaban de mencionar, son para obtener los registros que se desean de tal forma que se puedan exportar y ser analizados con el paquete estadístico SPSS.

4.3 FASE 3 MINERÍA DE DATOS

Antes de empezar a realizar esta fase, es importante realizar un análisis estadístico previo.

Primeramente**,** se tiene el objetivo de obtener las estadísticas para todos los casos posibles empezando por el panorama general hasta los casos particulares; por ejemplo, estadísticas para los casos en que *deserto='SI'* y *terminomaterias='NO'*; esto mismo se usará para la Minería de Datos ya que resulta mucho mejor analizar por casos concretos que por un caso general; cuando se tiene un caso muy general, pueden existir varios casos extremos que probablemente provoquen una cierta inclinación o ponderación de más a los modelos haciendo que éstos varíen más allá de lo que sería en realidad, es decir, analizando un caso específico.

Caracterizar a los datos ayuda a *desyerbar* valores distintos o inconsistentes que se pueden examinar más adelante para detectar problemas en los datos. (16)

Los datos de las tablas pueden ser importados por SPSS directamente desde la base de datos para poder obtener las estadísticas deseadas. Primero se recomienda obtener el manejador de BD de *MySQL* para *Windows* u otro sistema operativo según en el cual se esté trabajando. El programa se llama *mysql-connector-odbc-5.1.5-win32.msi* y puede encontrarse fácilmente en Internet. Luego se ejecuta ese mismo programa y se instala automáticamente. SPSS lo reconocerá automáticamente al momento de querer importar datos desde MySQL. Si no fuera el caso, el mismo asistente de importación de datos de distintos manejadores de bases de datos lo guiará fácilmente.

Figura 4.19. Arrancando SPSS 15.0.

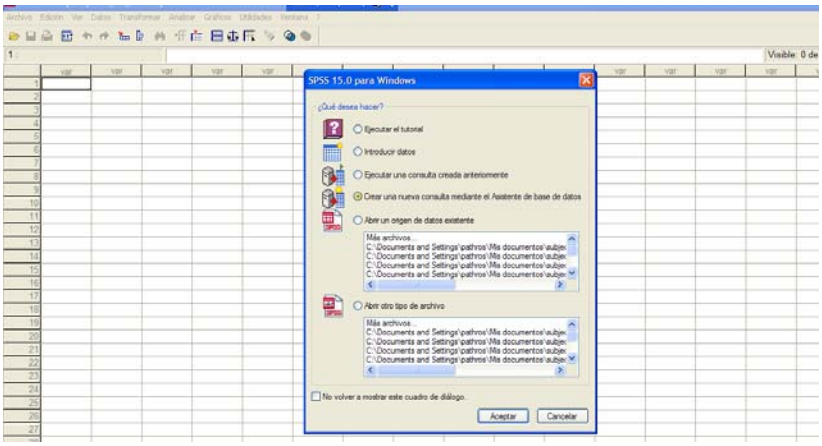


Figura 4.20. Seleccionando el modo de importación de datos desde una Base de Datos.

Una vez llevado a cabo esto, ábrase el *SPSS* (*figura 4.19*) y seleccionar *crear una nueva consulta mediante el Asistente de base de datos* y dar clic en *aceptar* (*figura 4.20)*. En el siguiente cuadro, seleccionar el orígen de datos, en este caso, de *MySQL* y dar clic en *siguiente* (*figura 4.21*). En el siguiente paso, se muestran todas las tablas de la base de datos del lado izquierdo (*figura 4.22*). De ahí, se pueden arrastrar las tablas que se deseen. Las columnas que no se necesiten se arrastran de regreso a la izquierda. Una vez que se han seleccionado las columnas deseadas, dar clic en *siguiente*.

La siguiente pantalla del asistente da la posibilidad de limitar los casos importados, es decir, para importar ciertos datos de acuerdo a una condición. En este caso, no es necesario y dar clic en *siguiente* (*figura 4.23*). Después se pide si se desea cambiar las variables que siendo numéricas, se manejan como nominales o categóricas. No se modifica nada (*figura 4.24*).

Finalmente el asistente muestra, a manera de una consulta SQL, el resultado de los datos que se van a importar. Dar clic en *finalizar* (*figura 4.25*).

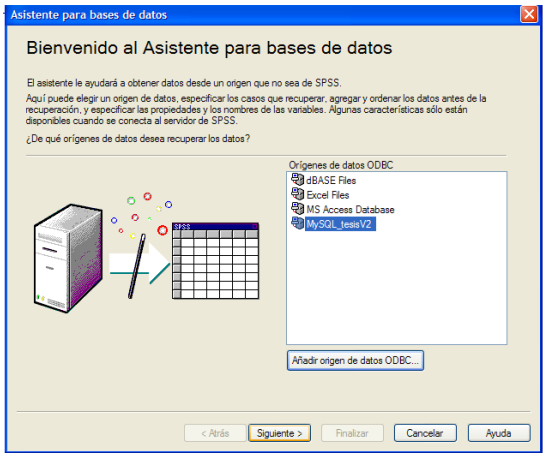


Figura 4.21. Seleccionando cuál es el manejador de la base de datos.

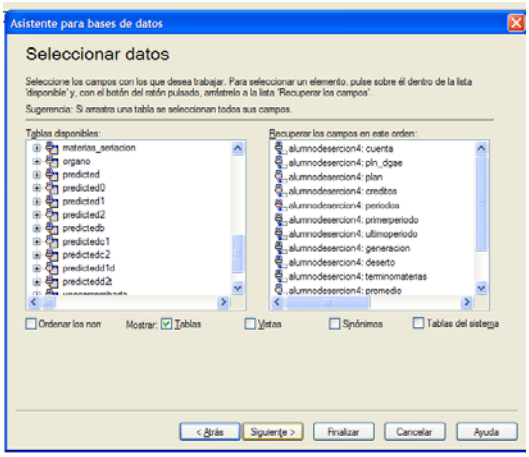


Figura 4.22. Seleccionando la tabla y los atributos.

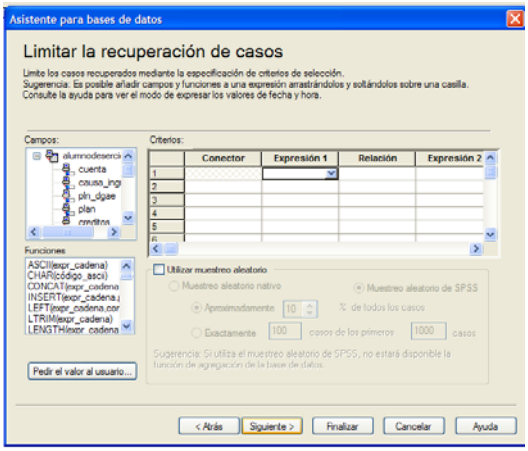


Figura 4.23. Aquí se puede delimitar la consulta.

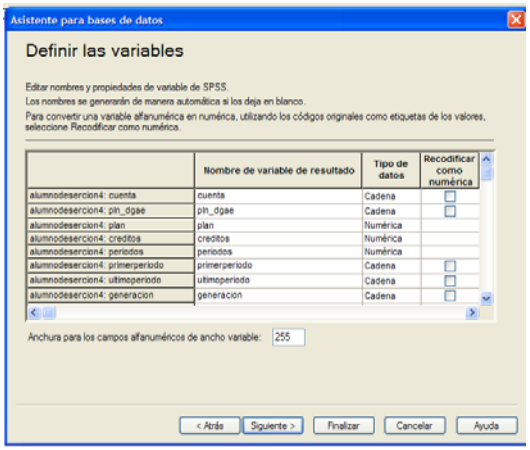


Figura 4.24. Se pregunta si se cambian los datos de nominales a numéricos.

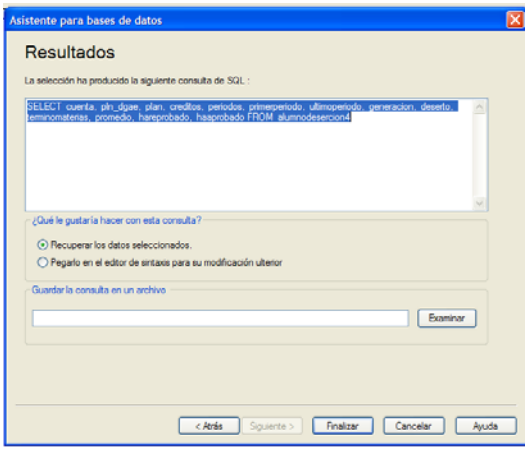


Figura 4.25. Visualizando cómo ha quedado la consulta SQL.

Todos los datos se importan y aparecen en SPSS (*figura 4.26*). Se puede dar clic en la pestaña *Vista de variables* para poder etiquetar el nombre de las columnas; esto es como los metadatos de las columnas (*figura 4.27*). Y se procede primeramente a guardar el archivo. En este caso el archivo tiene extensión *.sav*. Sería *alumnodesercion4.sav* (que son los datos más actuales).

Ahora se van a obtener las estadísticas. Del menú principal seleccionar *analizar*, *estadísticos descriptivos*, *frecuencias* (*figura 4.28*). Seleccionar las variables de las cuales se desean obtener estadísticas. En este caso seleccionar todas las variables con la excepción de *cuenta* (*figura 4.29*). Dar clic en *estadísticos* y seleccionar qué estadísticas se desean obtener y dar clic en *continuar* (*figura 4.30*). Regresando a la *figura 11*, ahora

seleccionar *gráficos*. En este caso seleccionar *gráficos de barras* y *frecuencias* y dar clic en continuar (*figura 4.31*).

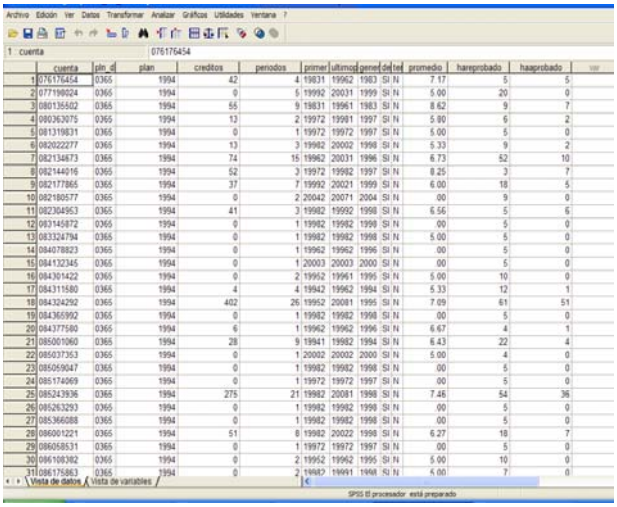
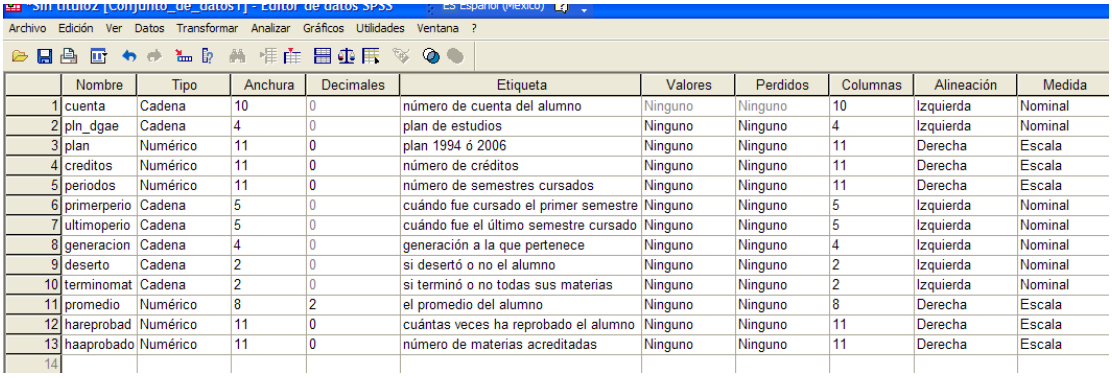


Figura 4.26. Datos ya importados en SPSS.



	Nombre	Tipo	Anchura	Decimales	Etiqueta	Valores	Perdidos	Columnas	Alineación	Medida
1	cuenta	Cadena	10	0	número de cuenta del alumno	Ninguno	Ninguno	10	Izquierda	Nominal
2	pln_dgae	Cadena	4	0	plan de estudios	Ninguno	Ninguno	4	Izquierda	Nominal
3	plan	Numérico	11	0	plan 1994 ó 2006	Ninguno	Ninguno	11	Derecha	Escala
4	creditos	Numérico	11	0	número de créditos	Ninguno	Ninguno	11	Derecha	Escala
5	periodos	Numérico	11	0	número de semestres cursados	Ninguno	Ninguno	11	Derecha	Escala
6	primerperio	Cadena	5	0	cuándo fue cursado el primer semestre	Ninguno	Ninguno	5	Izquierda	Nominal
7	ultimoperio	Cadena	5	0	cuándo fue el último semestre cursado	Ninguno	Ninguno	5	Izquierda	Nominal
8	generacion	Cadena	4	0	generación a la que pertenece	Ninguno	Ninguno	4	Izquierda	Nominal
9	deserto	Cadena	2	0	si desertó o no el alumno	Ninguno	Ninguno	2	Izquierda	Nominal
10	terminomat	Cadena	2	0	si terminó o no todas sus materias	Ninguno	Ninguno	2	Izquierda	Nominal
11	promedio	Numérico	8	2	el promedio del alumno	Ninguno	Ninguno	8	Derecha	Escala
12	hareprobad	Numérico	11	0	cuántas veces ha reprobado el alumno	Ninguno	Ninguno	11	Derecha	Escala
13	haaprobado	Numérico	11	0	número de materias acreditadas	Ninguno	Ninguno	11	Derecha	Escala
14										

Figura 4.27. Etiquetando las columnas.

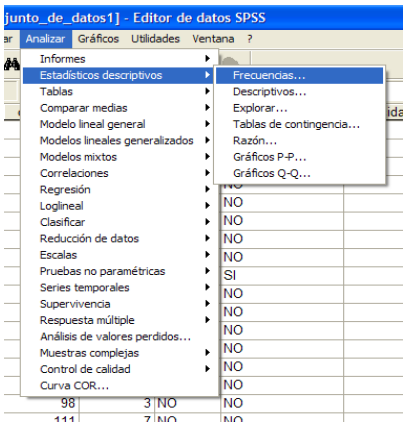


Figura 4.28. Seleccionando del menú las frecuencias.

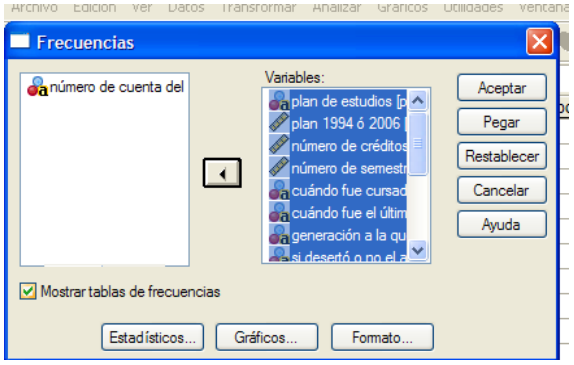


Figura 4.29. Seleccionando las variables.

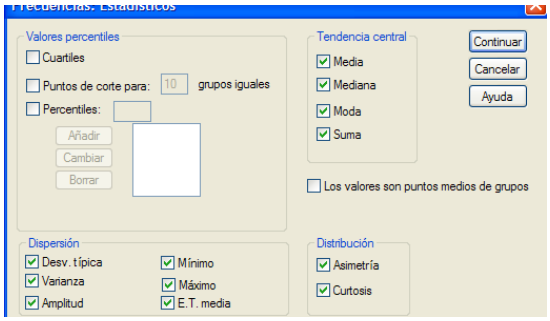


Figura 4.30. Seleccionando las estadísticas a obtener.

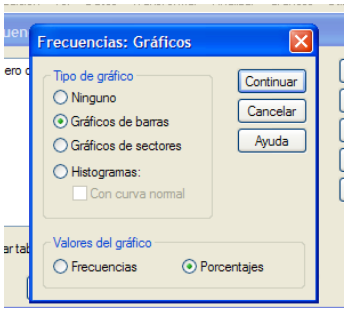


Figura 4.31. Seleccionando gráficos de barras.

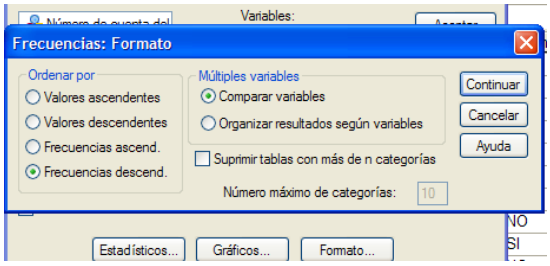


Figura 4.32. Seleccionando frecuencias descendentes.

Regresando de nuevo a la *figura 4.31*, ahora seleccionar *formato*. Ahí sólo seleccionar *frecuencias descendentes* y dar clic en *continuar* (*figura 4.32*). Se regresa de nuevo a la *figura 4.31* y entonces dar clic en *aceptar*. Inmediatamente las estadísticas se procesan (*figura 4.33*).

También se pueden obtener otras gráficas relacionando ciertas variables, por ejemplo hay que probar de la misma pantalla de la *figura 4.33*, seleccionar del menú principal *gráficos* y *generador de gráficos*. Y se mostrará una pantalla de explicación (*figura 4.34*) y simplemente hay que dar en *aceptar* (*figura 4.35*).

Entonces se mostrará un menú de gráficas con las variables a escoger. Simplemente para elegir lo que se desea basta con dar clic pero sin soltar el botón y arrastrar las opciones deseadas a la ventana principal y dar aceptar. Nótese que en todo procesamiento de estadísticas se muestran en otra ventana (marcada con una ventanita de color morado) mientras que los datos están en la ventanita marcada de color rojo).

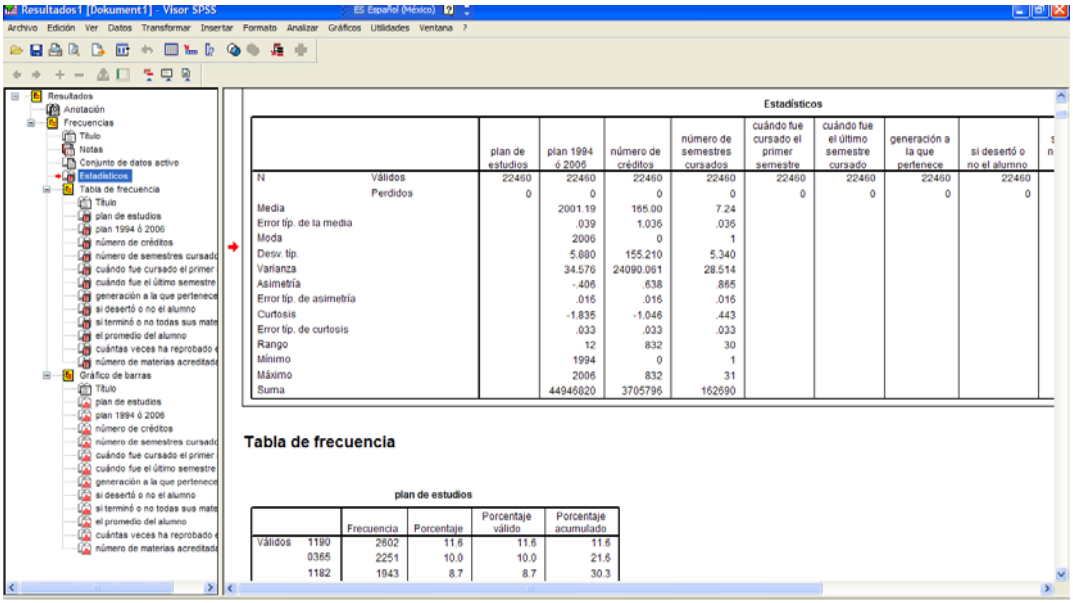


Figura 4.33. Resultados de las estadísticas.

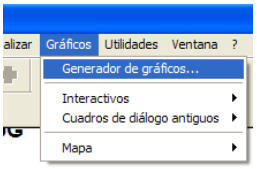


Figura 4.34. Menú para generar gráficos.

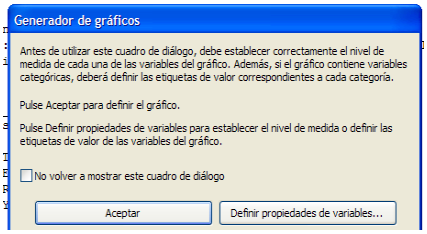


Figura 4.35. Pantalla de explicación.

Entonces se mostrará un menú de gráficas con las variables a escoger. Simplemente para elegir lo que se desea basta con dar clic pero sin soltar el botón y arrastrar las opciones deseadas a la ventana principal y dar aceptar. Nótese que en todo procesamiento de estadísticas se muestran en otra ventana (marcada con una ventanita de color morado) mientras que los datos están en la ventanita marcada de color rojo).

En este caso seleccionar una gráfica de barras de tres variables (arrástrese con el *mouse* en la ventana de arriba) en la que en el eje *y* se tiene el recuento, en el eje *x* se arrastra la variable de *número de semestres cursados* y como variable de *pila*, es decir, la que colorea la gráfica de barras, la variable que indica si un alumno desertó o no (véase la *figura 4.36*).

Dar clic en *aceptar* y en la ventana de resultados se obtiene la gráfica de barras que se muestra en la *figura 4.37*. Se puede notar que el mayor número de deserciones se da en los primeros semestres especialmente después del segundo semestre cursado. Pero se destaca que sigue habiendo deserciones en semestres posteriores pero se recalca también que, relativamente, el número de deserciones no es tan alto como se pensaba al menos para este plan nuevo 2006 pero en números no deja de ser alto.

Ahora se puede también obtener otra gráfica del mismo tipo pero en vez de la variable de deserción se escoge la variable de si el alumno terminó sus materias o no. *Figura 4.35.*

Cabe añadir que, viendo la *figura 4.36*, se puede convertir una variable de un tipo a otro dando clic con el botón derecho del *mouse* y elegir si se quiere nominal o numérico.

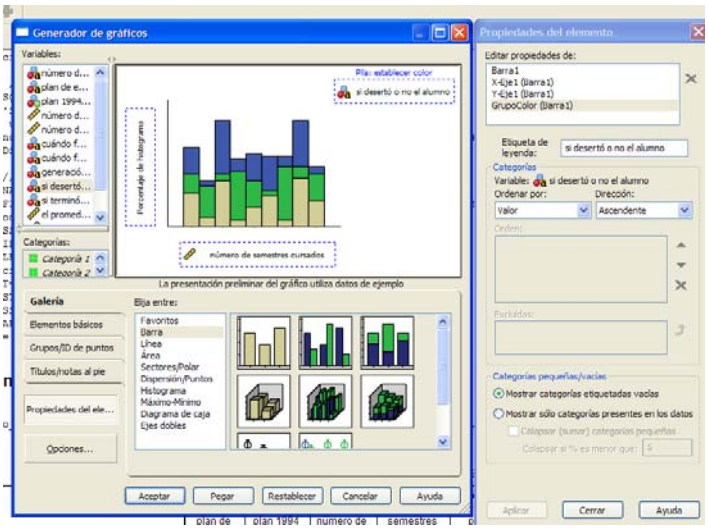


Figura 4.36. Seleccionando el tipo de gráfica y variables.

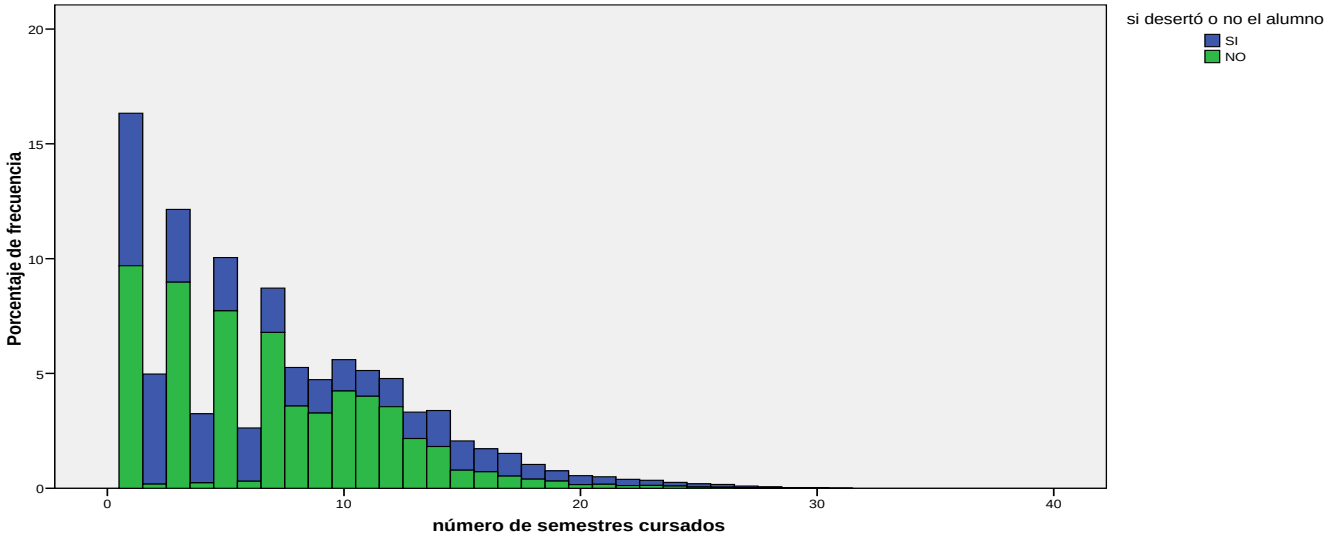


Figura 4.37. Gráfica deserciones contra semestres cursados.

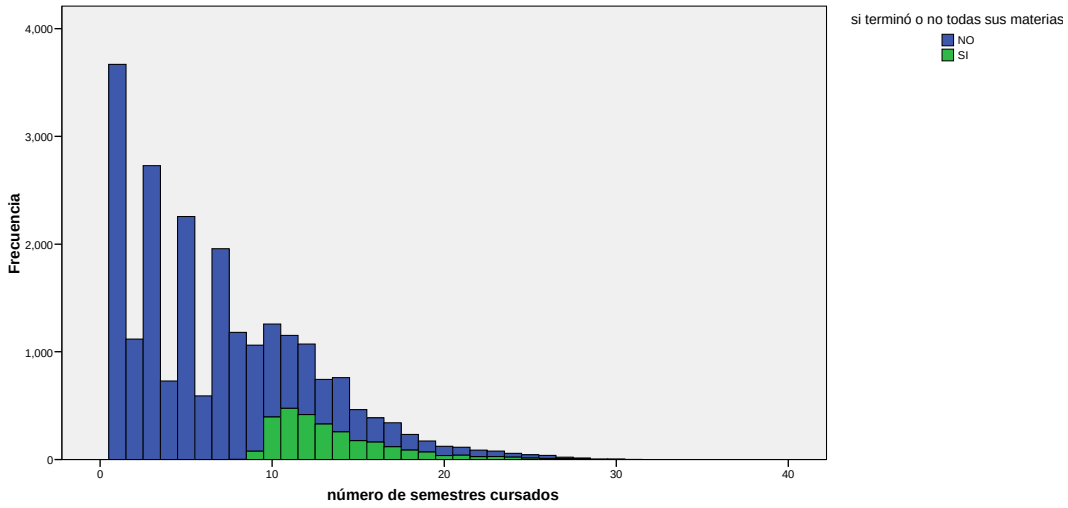


Figura 4.38. Gráfica de terminación de materias contra semestres cursados.

En esta gráfica de la *figura 4.38* se puede notar que los alumnos empiezan a terminar sus materias desde el *octavo* semestre (para el caso de un solo alumno) cursado en adelante. Esta gráfica es para los planes 1994 en adelante. Se nota que de los que terminan sus materias, la mayoría lo hace hasta después del décimo semestre cursado.

Para el caso del alumno que terminó en 8 semestres, se le considera como *outlier* ya que este registro no está *en línea* con el resto de los datos o que no es común (*Dasu & Johnson, 2003 [1] pág. 146*), desde que los planes de estudio son de 9 semestres. Este registro es destacado y también se puede detectar usando SQL:

```
SELECT * FROM alumnodesercion4
WHERE terminomaterias='SI'
ORDER BY periodos
```

Otra gráfica que se puede obtener, es aquella para saber en qué carrera hay más deserciones. Se realiza el mismo procedimiento (véase la *figura 4.39*).

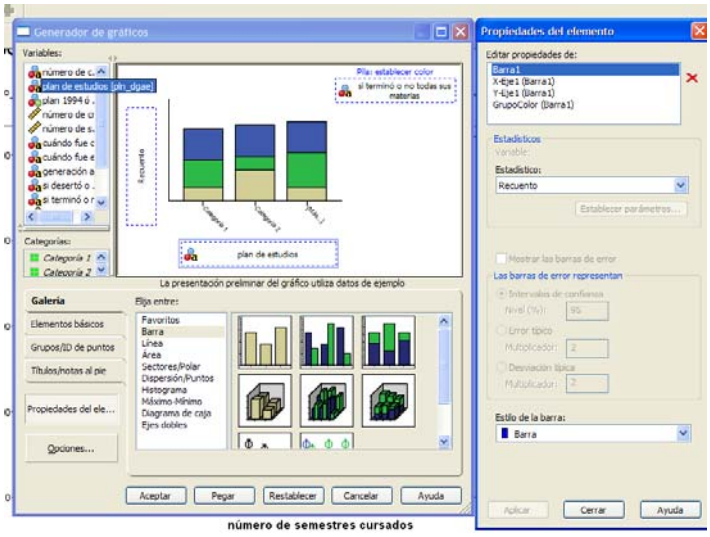


Figura 4.39. Cambiando el tipo de dato de una variable.

El cambio de tipo de variable se hace con la intención de que cada clave de la carrera se maneje como un ente independiente y no sea parte de una agrupación numérica. Al dar clic en *aceptar* se mostrará la gráfica, pero como hay demasiados datos la gráfica, se ve muy estrecha por lo que sobre la gráfica hay que dar doble clic y en el botón *x* le auméntese el ancho de la gráfica como a 1000 unidades y dar clic en *aplicar* y posteriormente ciérrese esta ventana (*figura 4.40*). Así se obtiene la gráfica mostrada en la *figura 4.41*.

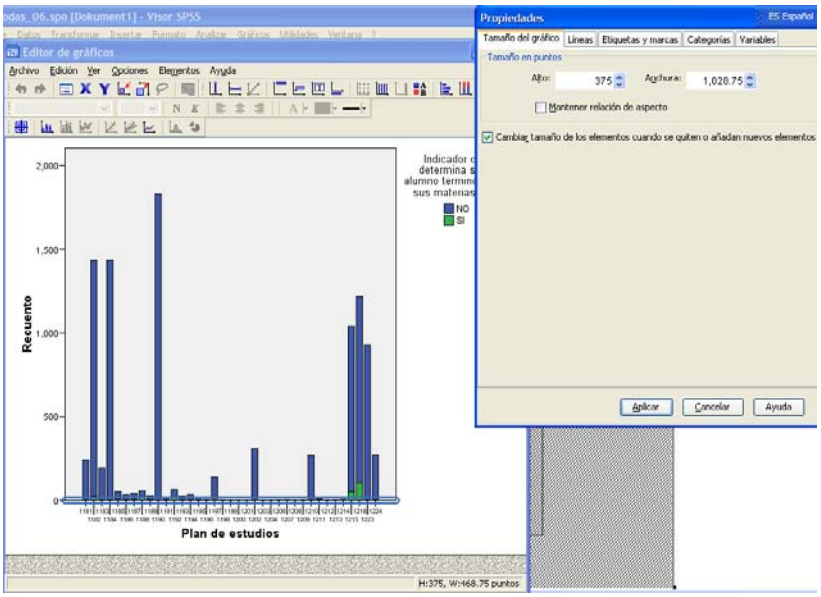


Figura 4.40. Alargando el ancho de la gráfica.

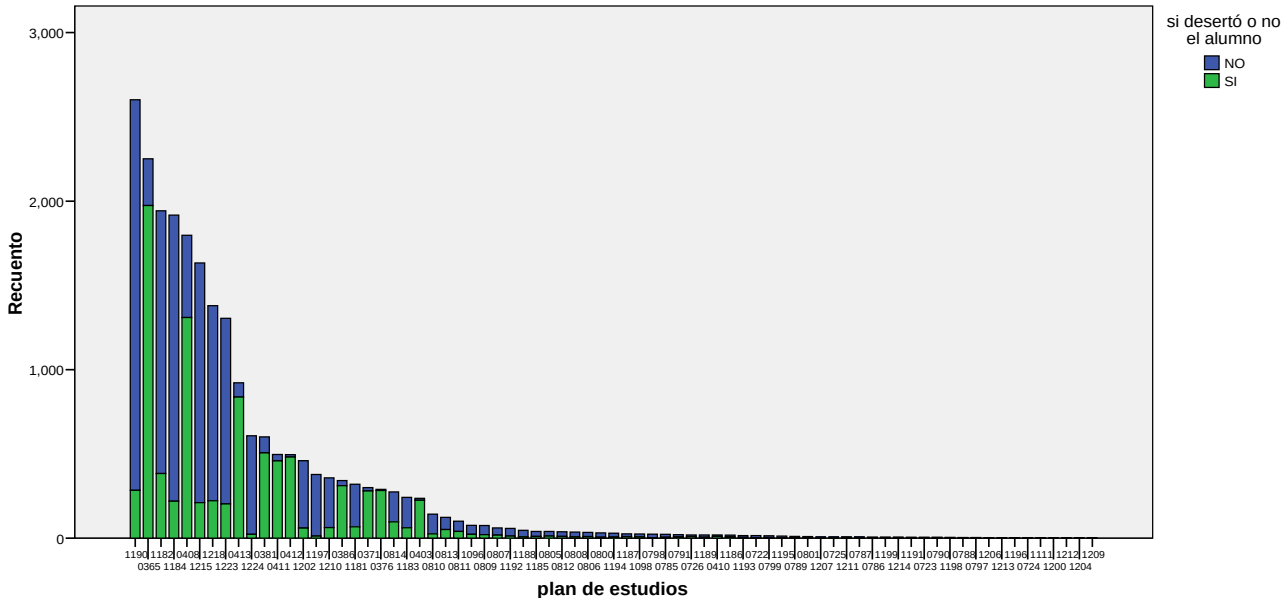


Figura 4.41. Gráfica de planes de estudio respecto a si desertaron o no.

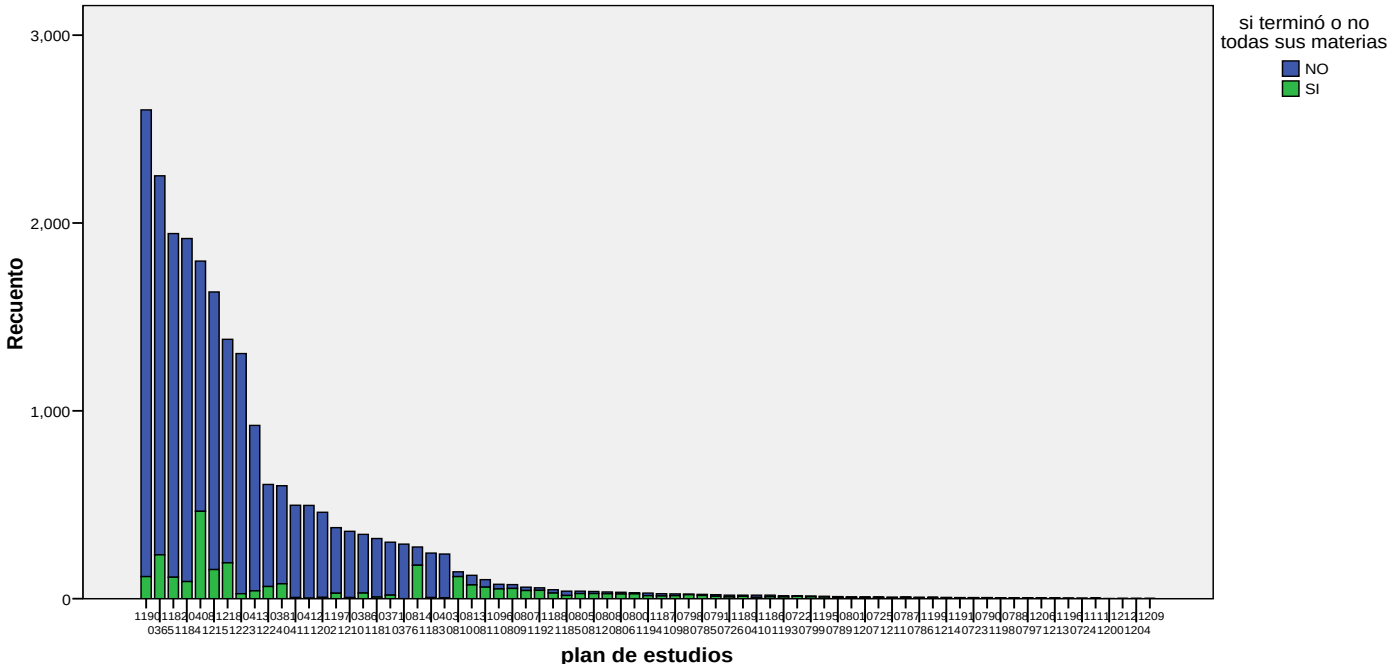


Figura 4.42. Gráfica de planes de estudio respecto a si se terminaron todas las materias.

De la gráfica de la *figura 4.41* se observa que en los planes 0365 (ingeniería civil [1994]), 0408 (ingeniería en computación [1994]) y 0413 (ingeniería eléctrica electrónica [1994]) y en más planes es donde se tienen más deserciones, recalcando que estas tres primeras son del plan de estudios 1994.

De la gráfica de la *figura 4.42* se observa que 0408 (ingeniería en computación [1994]), 0365 (ingeniería civil [1994]) y 1218 (ingeniería mecánica [2006]) es donde se tiene la mayor cantidad de alumnos que terminan sus materias (y que, por ende, tienen más probabilidades de titularse (queda fuera de nuestro alcance determinar si estos alumnos se titulan).

Nótese que en las carreras de civil y computación se tiene un alto número de deserción así como un alto número de estudiantes que terminan todas sus materias. Esto se debe a que son de las carreras de mayor matrícula.

Para tener una mejor perspectiva de estas últimas gráficas, se hace un diagrama de *Pareto*.

Primero se separaron en diferentes casos:

1.- El conjunto de alumnos que no desertan.

El query o consulta es:

```
SELECT * FROM alumnodesercion4
WHERE deserto='NO';
```

Se puede ejecutar una nueva consulta desde SPSS como en la consulta anterior. Sígase los mismos pasos. Para agregar esta condición, cuando el asistente muestre la parte de *limitar la recuperación de casos*, en la expresión 1 seleccionar, en este caso donde diga *alumnodesercion4: deserto*. En la columna *relación* elíjase el signo igual = y en la columna *expresión* teclear *'NO'* que es la condición (*figura 4.43*). Al final, el asistente muestra el SQL de SPSS (*figura 4.44*). Se observa que la condición se encierra entre paréntesis. Al terminar SPSS cargará los nuevos datos en otra ventana.

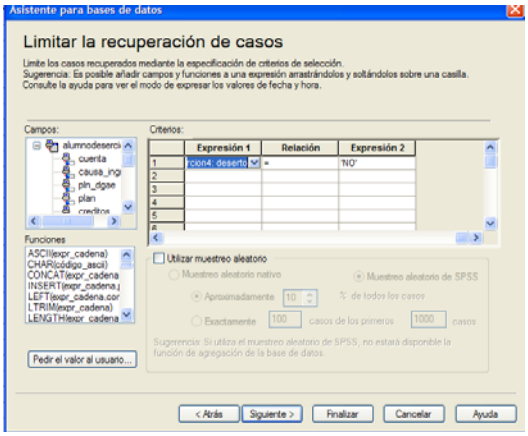


Figura 4.43. Limitando los datos de la consulta según la condición deseada.

Ábranse estos datos con el SPSS de la misma manera en que se obtuvieron las estadísticas anteriores. Se desea hacer un diagrama de Pareto, ya que este dice que el 20% de cualquier cosa producirá el 80% de los efectos, mientras que el 80% restante sólo cuenta para el 20% de los efectos.

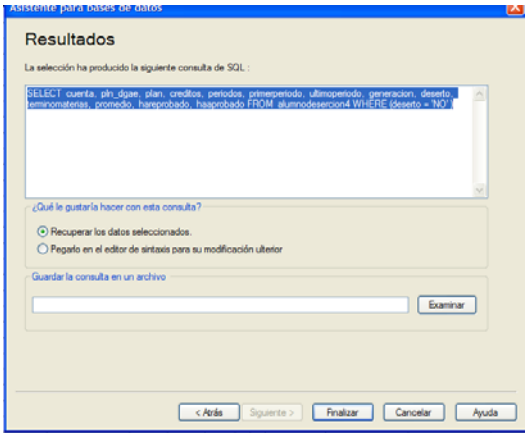


Figura 4.44. Al final se observa cómo queda la consulta en SQL de SPSS.

Para realizar un diagrama de Pareto en SPSS, hay que seleccionar del menú principal *analizar*, *control de calidad* y *gráficos de pareto*. Véase la *figura 4.45*. Luego escójase en este caso la opción *simple* (la opción de *apilado* sirve cuando se tienen dos variables a analizar) y dar clic en *definir*. De ahí, hay que seleccionar la variable o las variables, en este caso elegir *pln_dgae* y finalmente dar clic en *aceptar*. Esto es para saber qué planes de estudio conforman el 20% del total que produce el 80% de los efectos. Es decir, para fines prácticos de la Minería de Datos, saber qué planes de estudio seleccionar para realizar la búsqueda de patrones.

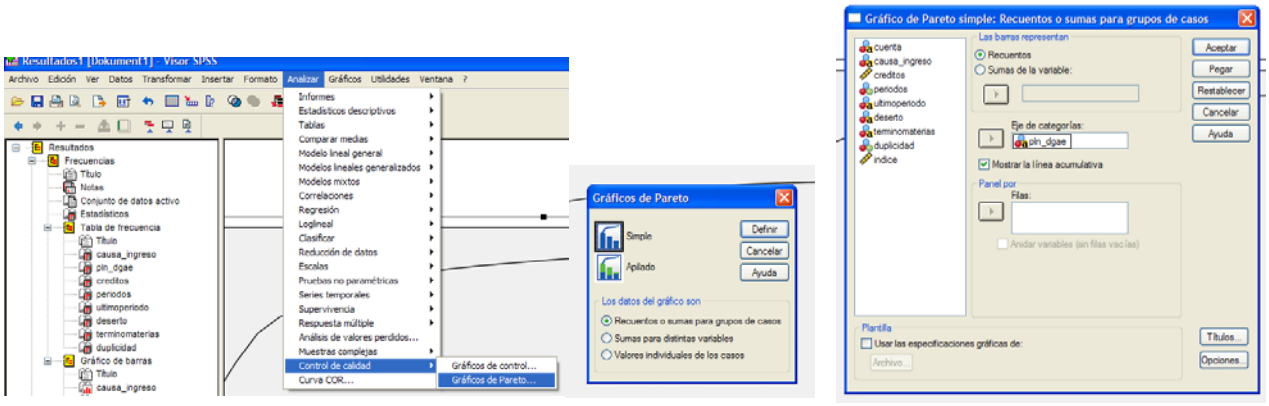


Figura 4.45. Seleccionando las gráficas de Pareto. Eligiendo la opción *simple* y seleccionando la variable *pln_dgae*.

La gráfica de *Pareto* que se obtiene es la siguiente (véase la *figura 4.46*):

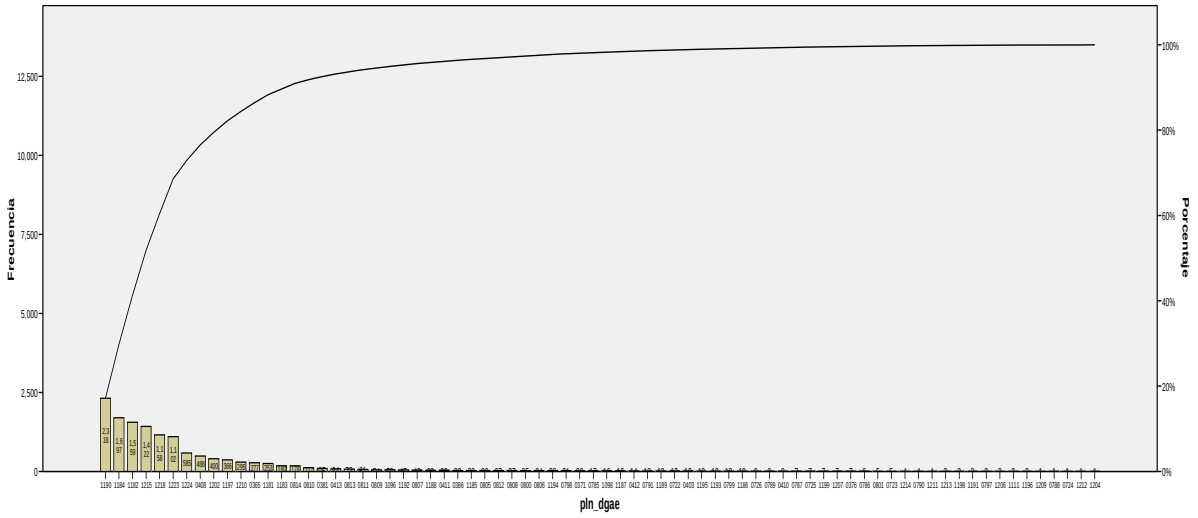


Figura 4.46. Gráfica de pareto para el caso de los estudiantes que no han desertado.

A esta gráfica aproximadamente se le marca con unas líneas para saber dónde se concentra ese 20% de los planes de estudio que más influyen. Se observa que abarca los planes de estudio 1190 (Ing. en computación), 1184 (ing. Eléctrica Electrónica), 1182 (Ing. Civil), 1215 (Ing. Industrial), 1218 (Ing. Mecánica), 1223 (Ing. Petrolera), 1224 (Ing. Mecatrónico), 0408 (Ing. en computación [1994]) y 1202 (Ing. Geofísica). Estos resultados son normales puesto que corresponden a los planes de estudio que son comunes, es decir, sin especialidad, ya que, cuando un alumno cursa los últimos semestres, tiene que elegir un módulo de especialización, los cuales usualmente no son tan numerosos como en los primeros semestres.

2.- El conjunto de alumnos que sí desertan.

El *query* o la consulta es:

```
SELECT * FROM alumnodesercion4
WHERE deserto='SI';
```

Se hace lo mismo que en el *query* anterior y se obtienen las gráficas de Pareto. En la *figura 4.47* se tiene que el 20% concentra a las carreras, mayoritariamente del plan 1994, 0365 (Ing. Civil [1994]), 0408 (Ing. en Computación [1994]), 0413 (Ing. Eléctrica Electrónica [1994]), 0381 (Ing. Petrolera [1994]), 0412 (Ing. Industrial [1994]), 0411 (Ing. Mecánica [1994]), 1182 (Ing. Civil), 0386 (Ing. Topográfica y Geodesta [1994]), 1190 (Ing. en Computación), 0376 (Ing. Geológica [1994]) y 0371 (Ing. de Minas y Metalurgia [1994]). Que son los planes de estudio en donde se concentra el 80% de los alumnos que desertan.

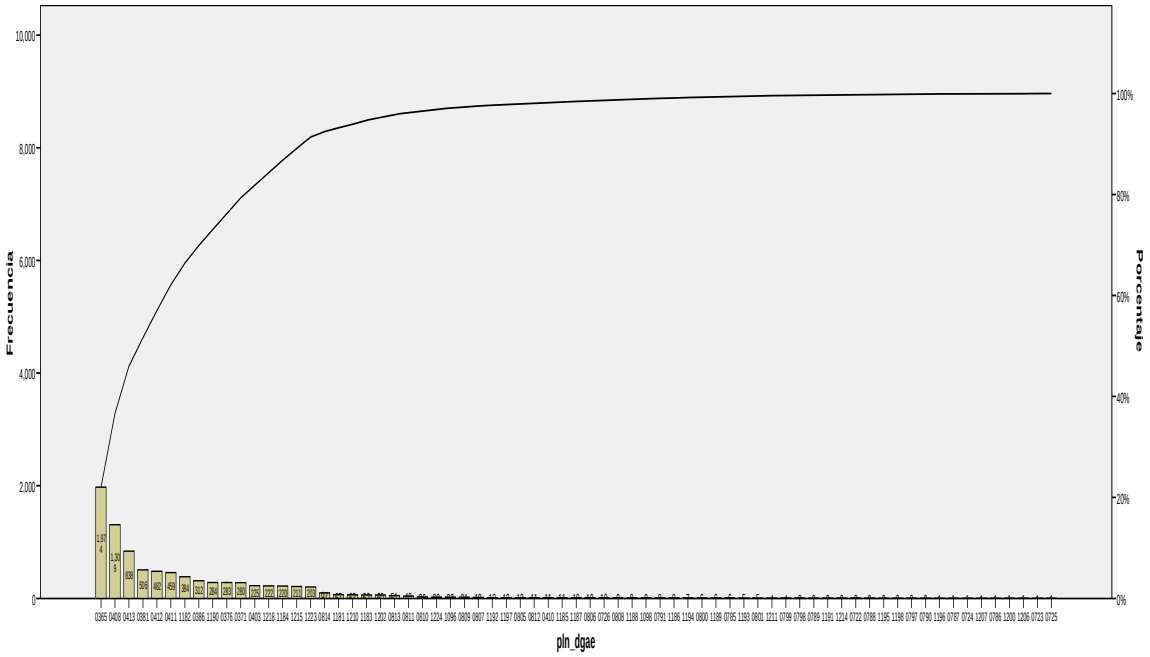


Figura 4.47. Alumnos que desertaron según el plan de estudios.

En la *figura 4.48* se muestra que en los primeros 11 semestres es cuando el 80% de los alumnos desertan. Especialmente en el 1ro, 2do, 3ro y 4to semestre pero también destacan los alumnos que desertan más allá del décimo semestre y se podría pensar que a esas alturas un estudiante ya pasó el anexo, es decir, el tronco común donde se vieron todas las materias de matemáticas y física. Pero la realidad es que los alumnos que

desertan rara vez completan el tronco común. Pocos desertan faltándoles pocas materias para terminar.

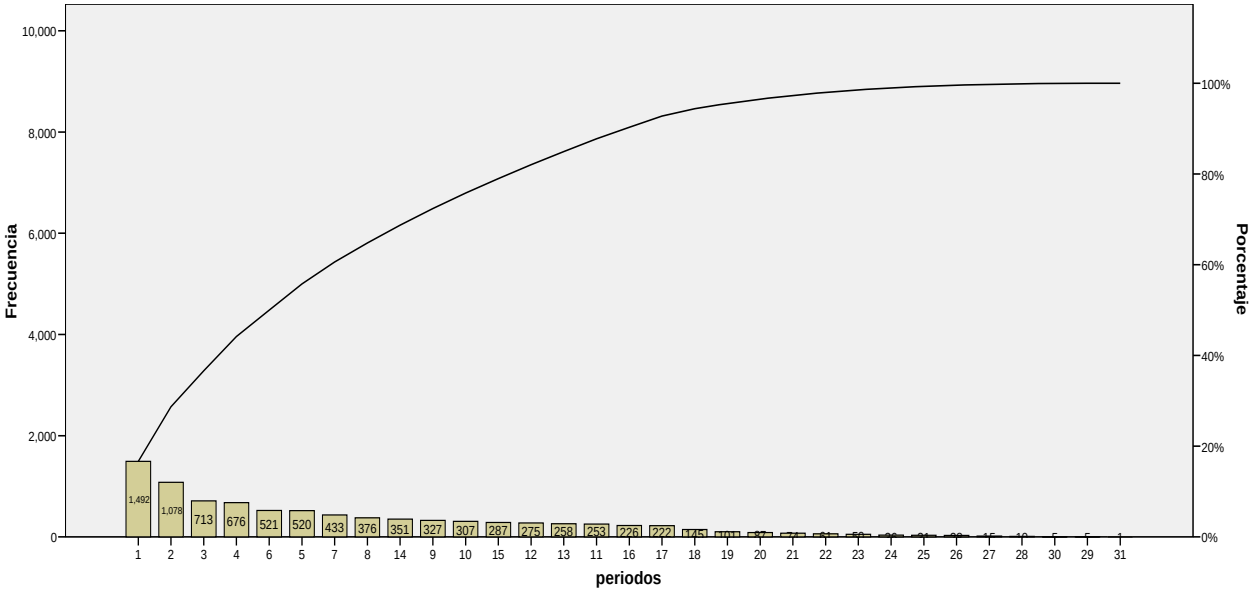


Figura 4.48. En qué semestres desertan los alumnos.

3.- El conjunto de alumnos que sí terminan sus materias (y que no desertaron).

El *query* es el siguiente:

```
SELECT * FROM alumnodesercion4
WHERE terminomaterias='SI';
```

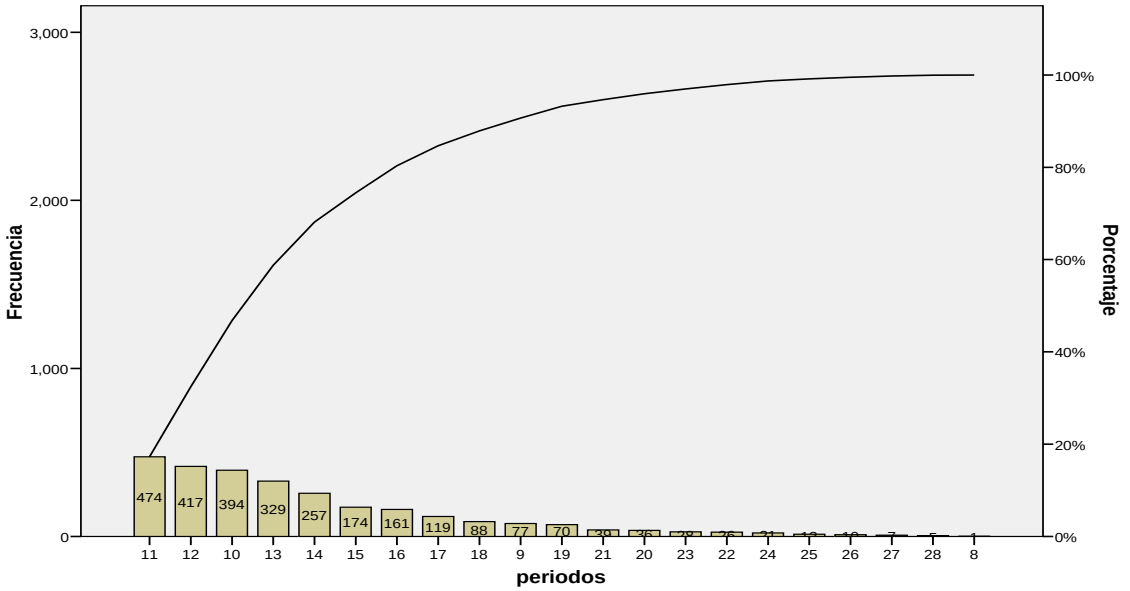


Figura 4.49. Gráfica de Pareto de los alumnos que sí terminan sus materias por número de semestres cursados.

En la gráfica de la *figura 4.49* se muestra que el 80% de los alumnos que sí terminan sus materias, terminan entre el décimo y el décimo sexto semestre. El rango del promedio de calificaciones va desde 6.77 a 9.96. La moda es de 8.02 y la media del promedio de calificaciones es de 8.1957.

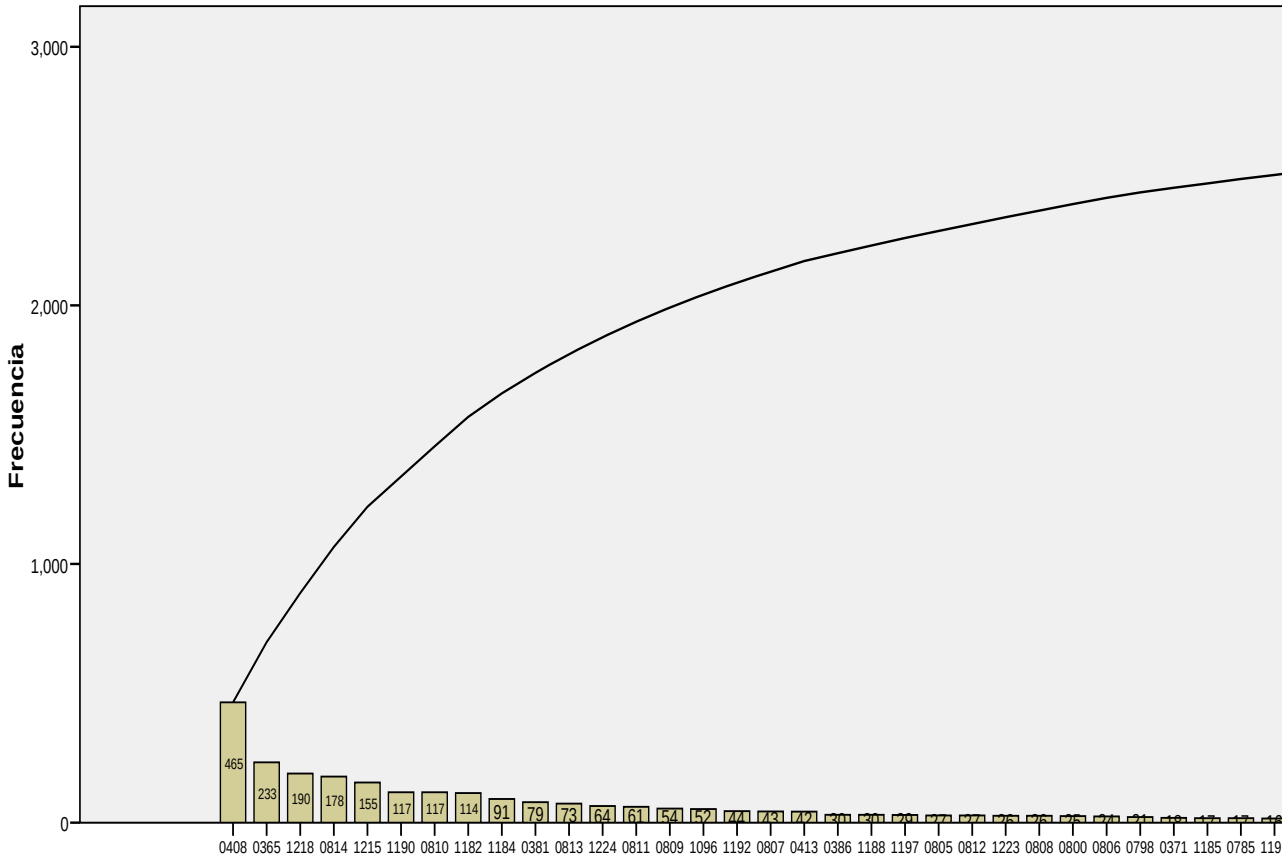


Figura 4.50. Gráfica de Pareto de los alumnos por planes de estudios DGAE que terminan todas sus materias.

En la *figura 4.50*, se muestra que el 80% de los alumnos que sí terminan sus materias son de las carreras 0408 (Ing. en Computación [1994]), 0365 (Ing. Civil [1994]), 1218 (Ing. Mecánico), 0814 (Ing. Eléctrica Electrónica – Energía Eléctrica [1994]), 1215 (Ing. Industrial), 1190 (Ing. en Computación), 0810 (Ing. Industrial – admón. y sistemas [1994]), 1182 (Ing. Civil), 1184 (Ing. Eléctrica Electrónica), 0381 (Ing. Petrolera [1994]), 0813 (Ing. Eléctrica Electrónica – Electrónica para Comunicaciones [1994]), 1224 (Ing. Mecatrónica), 0811 (Ing. Eléctrica Electrónica – Electrónica [1994]), 0809 (Ing. Industrial – Producción [1994]), 1096 (Ing. Eléctrica Electrónica – Biomédicas [1994]), 1192 (Ing. en Computación – Redes y Seguridad), 0807 (Ing. Mecánica – Ter y

Mej Amb [1994]), 0413 (Ing. Eléctrica Electrónica [1994]) y 0386 (Ing. Topográfica y Geodesta [1994]).

Un problema que se nota es que, normalmente en los planes de estudio 2006, cuando un alumno escoge un módulo de salida para la carrera de ingeniería en computación (clave plan DGAE 1190) éste cambia según el módulo, es decir, puede ser desde el plan 1191 (Ing. de hardware) hasta el 1196 (biomédicas). Normalmente a los alumnos se les registra su cambio a partir del semestre que cursaron por lo que en sus registros aparece que las cuentas tienen dos o incluso más planes de estudio. Se considera normal, ya que se puede dar seguimiento a esos alumnos para saber si cambiaron de plan de estudios o en qué semestre se cambiaron de plan de estudios. Pero existe el problema de que en algunas cuentas se actualizó el cambio de plan o la elección del módulo de salida para todo su historial, lo cual impide dar seguimiento al historial del alumno en cuanto a si cambió de plan de estudios y cuándo fue éste (esto se puede observar en la tabla *alumnodesercion* en la columna *duplicidad*, si ésta es cero, es porque sólo tiene registrado un solo plan de estudios; si es uno, significa que tiene dos o más planes de estudios registrados). Para lidiar con el problema, sólo se obtienen los números de cuenta distintos, sin que se repitan con el plan de estudios con el que terminan, desertan o con el que están cursando.

Por ejemplo, de una gráfica se observó lo siguiente (*figura 4.51*):

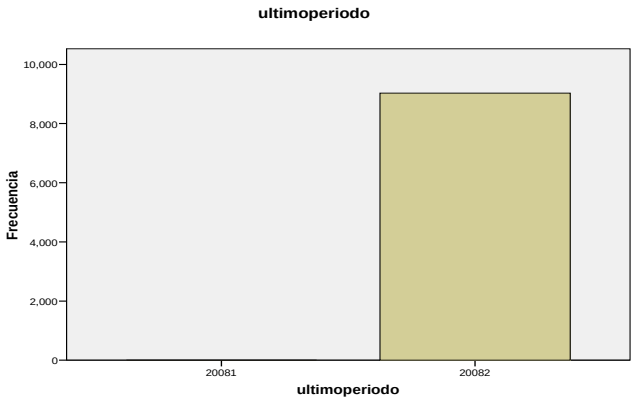


Figura 4.51. Se observa un valor anómalo (2008-1).

Entonces se procede a investigar el porqué. Se hace una consulta en los datos para saber qué número(s) de cuenta(s) tiene(n) ese semestre:

```
SELECT * FROM alumnodesercion4
WHERE terminomaterias='NO'
AND deserto='NO'
AND ultimoperiodo='20081';
```

Se tiene un solo resultado:

cuenta	Causa_ingreso	Pln_dgae	creditos	periodos	ultimoperiodo	deserto	terminomaterias	duplicidad	Indice
*****	72	1218	415	14	20081	NO	NO	1	****

Se obtiene el número de cuenta. Normalmente este resultado estaría clasificado como un alumno que desertó. Volviendo a buscar éste número de cuenta en la tabla *alumnodesercion* se obtienen dos resultados del mismo número de cuenta:

```
SELECT * FROM alumnodesercion4
WHERE cuenta='0980*****';
```

cuenta	Causa_ingreso	Pln_dgae	creditos	periodos	ultimoperiodo	deserto	terminomaterias	duplicidad	Indice
098010488	72	1218	415	14	20081	NO	NO	1	9229
098010488	68	1224	415	14	20081	SI	NO	1	11661

Asimismo investigando en su historial y revisando el número de sus créditos:

```
SELECT h.cuenta,h.carrera,h.pln_dgae,h.causa_ingreso,h.asignatura AS 'clave asignatura',
a.nombre_completo,h.periodo,h.calificacion,h.aprobo,h.tipo
FROM historias h, asignatura a
WHERE h.asignatura=a.clave
AND cuenta='098010488'
ORDER BY h.periodo
```

Se observa que el alumno cursó una materia del plan 1218 y fue reprobada (calificación de *NP*). Esa es la única materia marcada con ese plan de estudios, por lo que ha sido un dato erróneo o *sucio* – en la jerga de la minería de datos. Pero si se obtienen sólo las materias que fueron aprobadas:

```
SELECT h.cuenta,h.carrera,h.pln_dgae,h.causa_ingreso,h.asignatura AS 'clave asignatura',
a.nombre_completo,a.creditos AS creditos_materia,h.periodo,h.calificacion,h.aprobo,h.tipo
FROM historias h, asignatura a
WHERE h.asignatura=a.clave
AND aprobo='SI'
AND cuenta='098010488'
ORDER BY h.periodo
```

En este último *query* sólo hay materias cursadas y aprobadas de la carrera 1224 que pide 421 créditos (la carrera 1218 pide 406 créditos). En este caso, el alumno cuenta con 415 créditos. Tomando en cuenta los criterios anteriores, el alumno sí desertó desde que se ha determinado que el alumno en cuestión pertenece en realidad al plan de estudios 1224. Por lo que el dato en donde se marca que éste alumno sí terminó sus materias y que no desertó para la carrera 1218, no se tomará en cuenta de tal forma que no pondere los modelos que se obtengan de la minería (el dato se borrará y se respaldará en un archivo de texto).

Se observa que es de gran utilidad analizar los datos que tienen características extraordinarias, anormales o poco usuales ya que lleva a la detección de posibles anomalías.

4.- El conjunto de alumnos que no terminan o que aún no han terminado sus materias y no han desertado. Esto puede dar un panorama de los alumnos que están estudiando (al semestre inmediato anterior) y así saber en qué semestres se encuentra el grueso de estos estudiantes.

El *query* o consulta es el siguiente:

```
SELECT * FROM alumnodesercion4
WHERE terminomaterias='NO'
AND deserto='NO';
```

Se procede a obtener las gráficas y las estadísticas:

De la *figura 4.52* se observa que el 80% de los alumnos se encuentran cursando en los semestres 1, 3, 5, 7 y 8.

De la *figura 4.53* se observa los planes de estudio que tienen al 80% de los alumnos cursando según su plan de estudios los cuales son: 1190 (Ing. en Computación), 1184 (Ing. Eléctrica Electrónica), 1182 (Ing. Civil), 1215 (Ing. Industrial), 1223 (Ing. Petrolera) y 1218 (Ing. Mecánica).

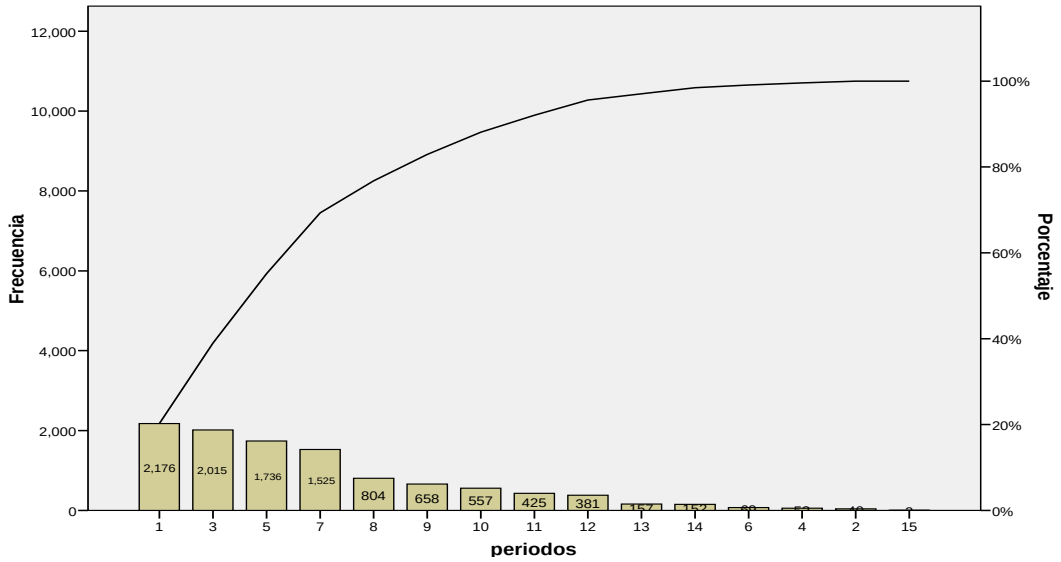


Figura 4.52. Diagrama de pareto para los alumnos que está cursando en los semestres ahí mostrados.

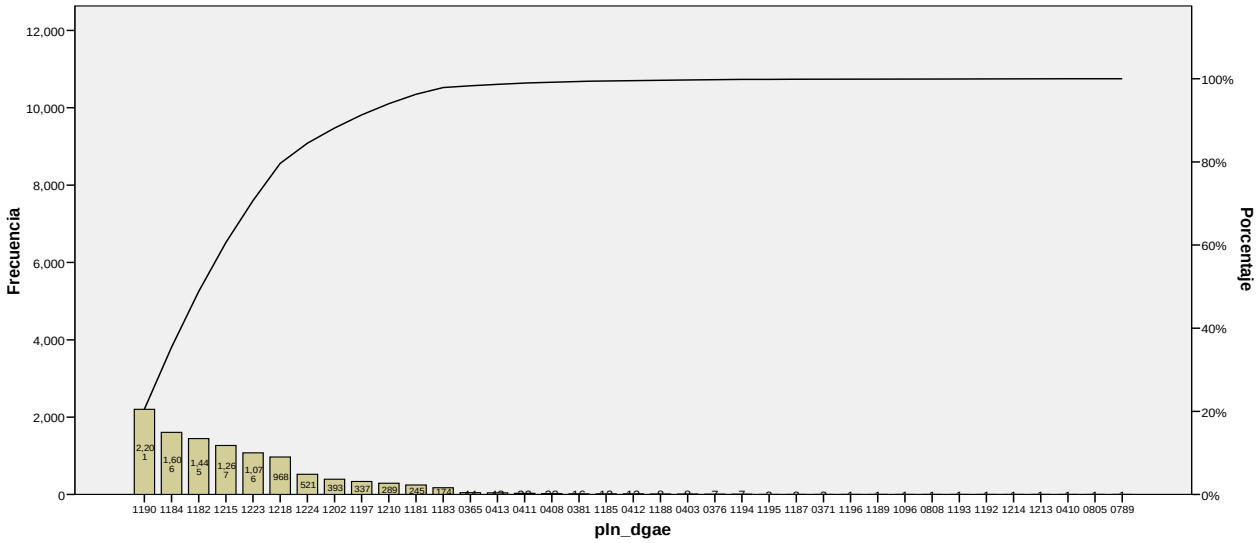


Figura 4.53. Diagrama de pareto de los planes de estudio de los alumnos que están cursando.

Se tiene también la gráfica de qué generaciones han cursado hasta el semestre inmediato anterior 2009-1 (*figura 4.54*):

En la gráfica anterior se observan generaciones de 1995, del 2000 y 2001. En este caso son alumnos que dejaron de venir y regresaron. De otra manera, ya no podrían seguir cursando. Esto se puede saber haciendo una consulta a la base de datos con la que se cuenta especificando que se muestre, por ejemplo, hasta la generación 2002:

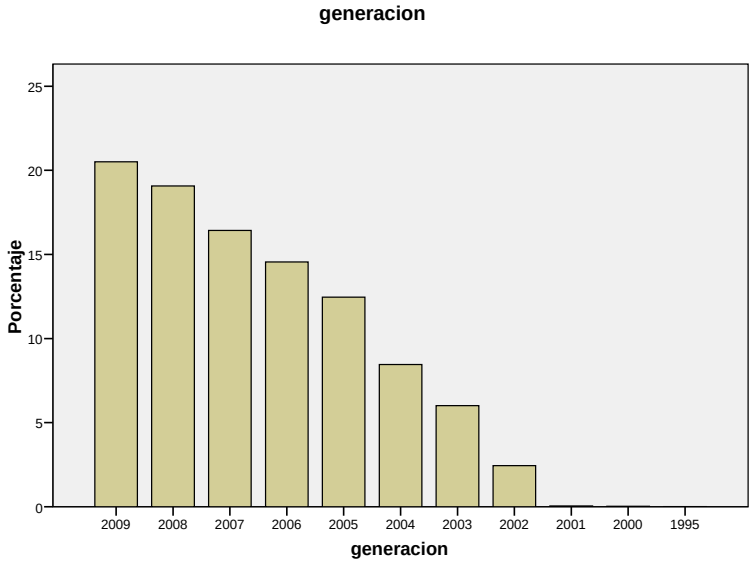


Figura 4.54. Generaciones cursando en el semestre inmediato anterior 2009-1.

```
SELECT * FROM alumnodesercion4
WHERE terminomaterias='NO'
AND deserto='NO'
AND generacion <= 2002
ORDER BY generacion,periodos;
```

Y parte de los resultados que se obtienen se muestran a continuación (*figura 4.55*):

Se observa que en el caso de la generación 1995, lleva apenas ocho semestres cursados, ha aprobado 30 materias y reprobado tres veces. Los tres casos de la generación 2000, llevan de 13 a 15 semestres cursados. De los que quedan de la generación 2001, llevan de 9 a 14 semestres cursados.

Ahora se obtiene información más detallada para el caso de la generación 1995 obteniendo su historial:

```
SELECT * FROM historias
WHERE cuenta='09125****'
ORDER BY periodo
```

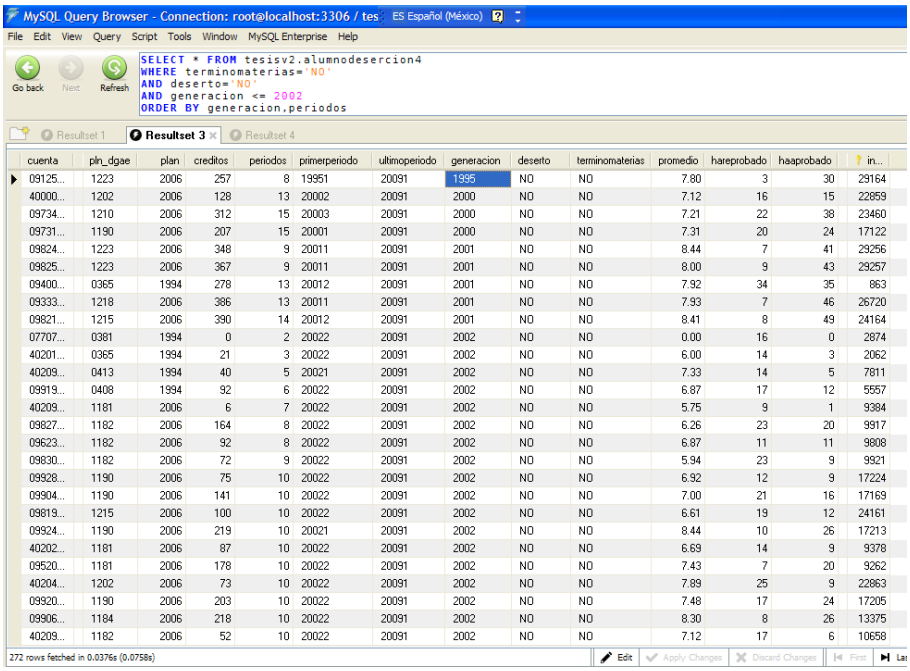


Figura 4.55. Generaciones cursando en el semestre inmediato anterior 2009-1.

Se observa de la figura 4.56, que entre los períodos 1999-2 y 2006-2, el alumno estuvo ausente. Y lo mismo sucede en los demás casos. Es por eso que aún tienen posibilidad, por reglamento, de seguir con sus estudios suponiendo que pidieron permiso de baja temporal.

Como se planteó anteriormente, se pueden obtener gráficas con la historia acumulada por semestre. Se obtuvieron tablas de *alumnodesercion* hasta los semestres 2007-1, 2007-2, 2008-1, 2008-2 y 2009-1. Nótese que para llegar a cada gráfica se tuvo que ejecutar los 12 procedimientos. Computacionalmente fue intenso.

Se obtuvieron 5 gráficas de deserciones por generación y van por orden de semestre ascendente (*figuras 4.57*, *4.58*, *4.59*, *4.60* y *4.61*) y otras 5 de terminación de materias por generación (*figuras 4.62*, *4.63*, *4.64*, *4.65* y *4.66*) en el mismo orden.

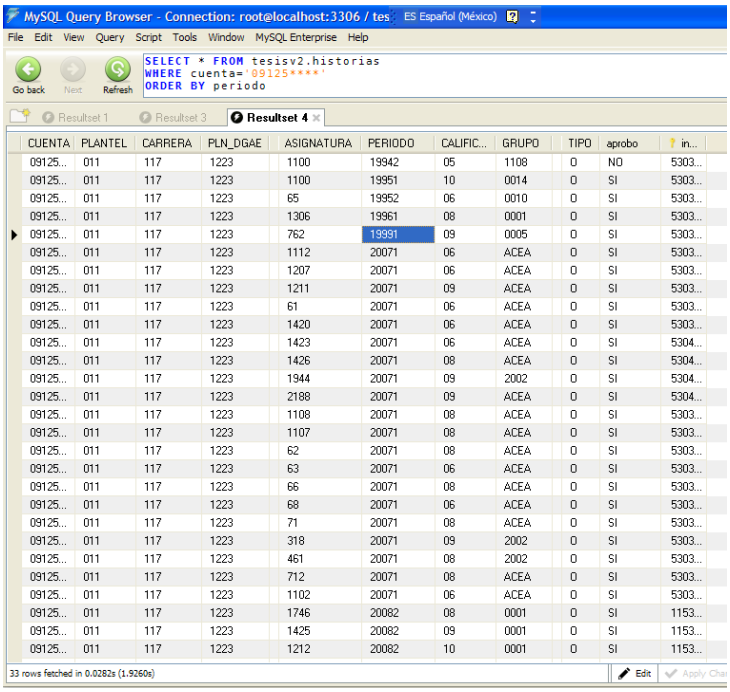


Figura 4.56. Historial del caso de la generación 1995.

Gráficas de deserciones:

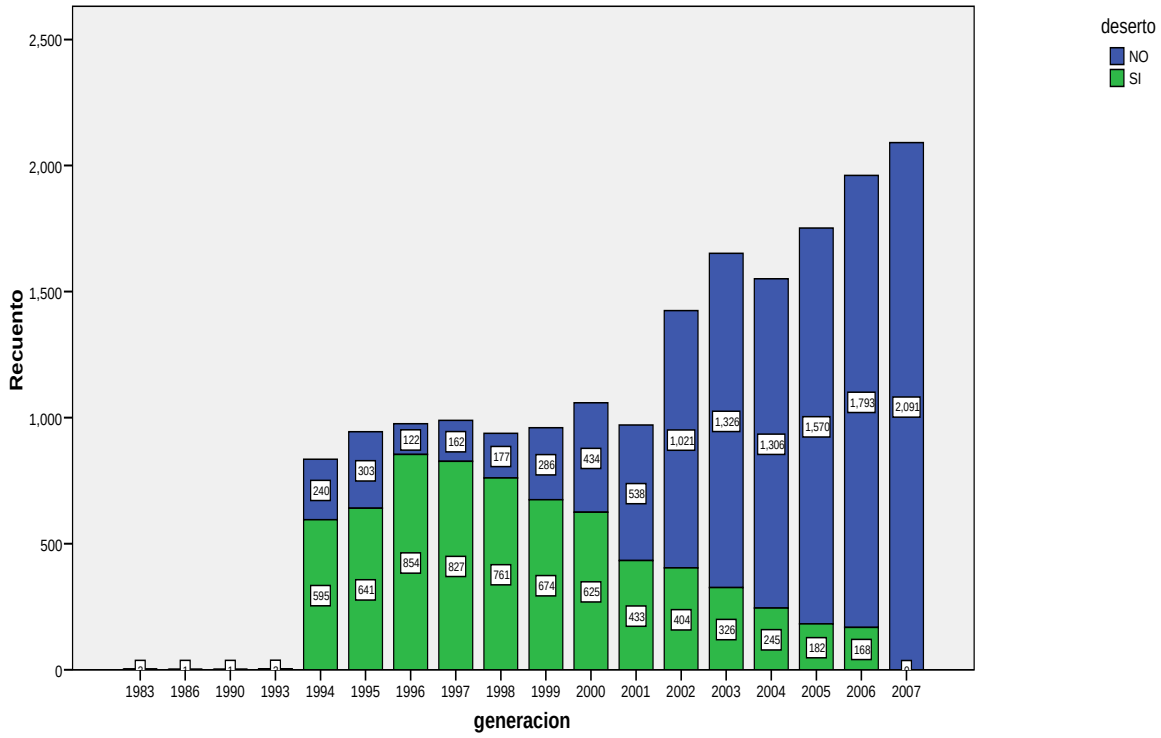


Figura 4.57 . Gráfica de deserción hasta el semestre 2007-1.

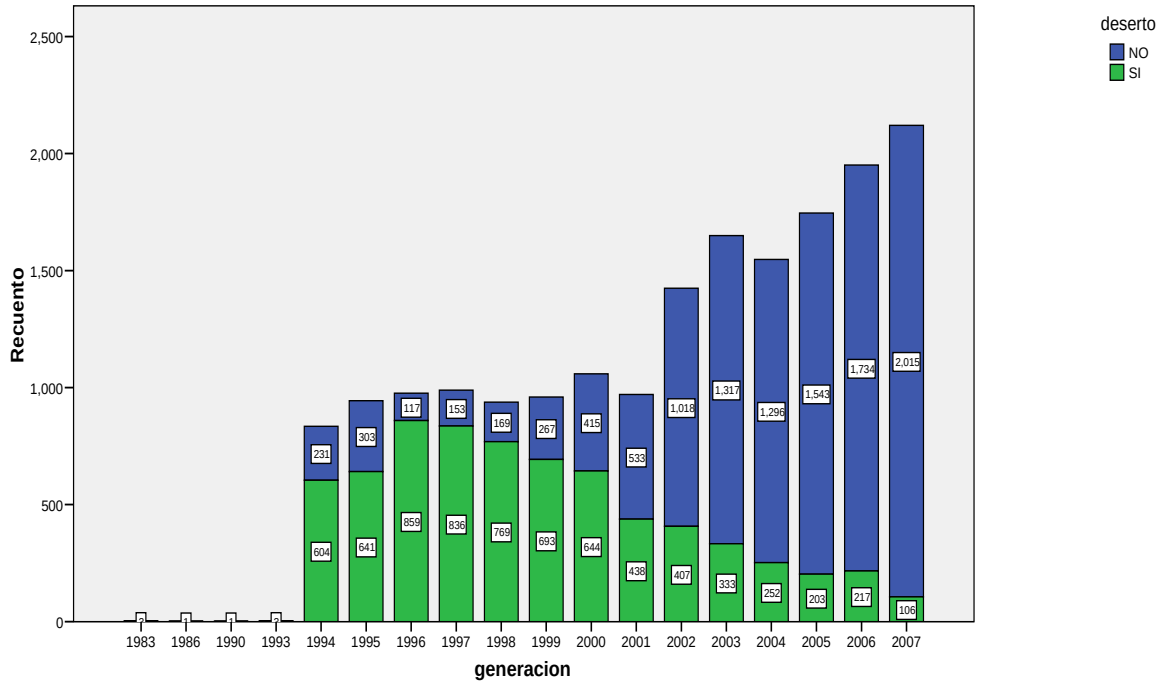


Figura 4.58. Gráfica de deserción hasta el semestre 2007-2.

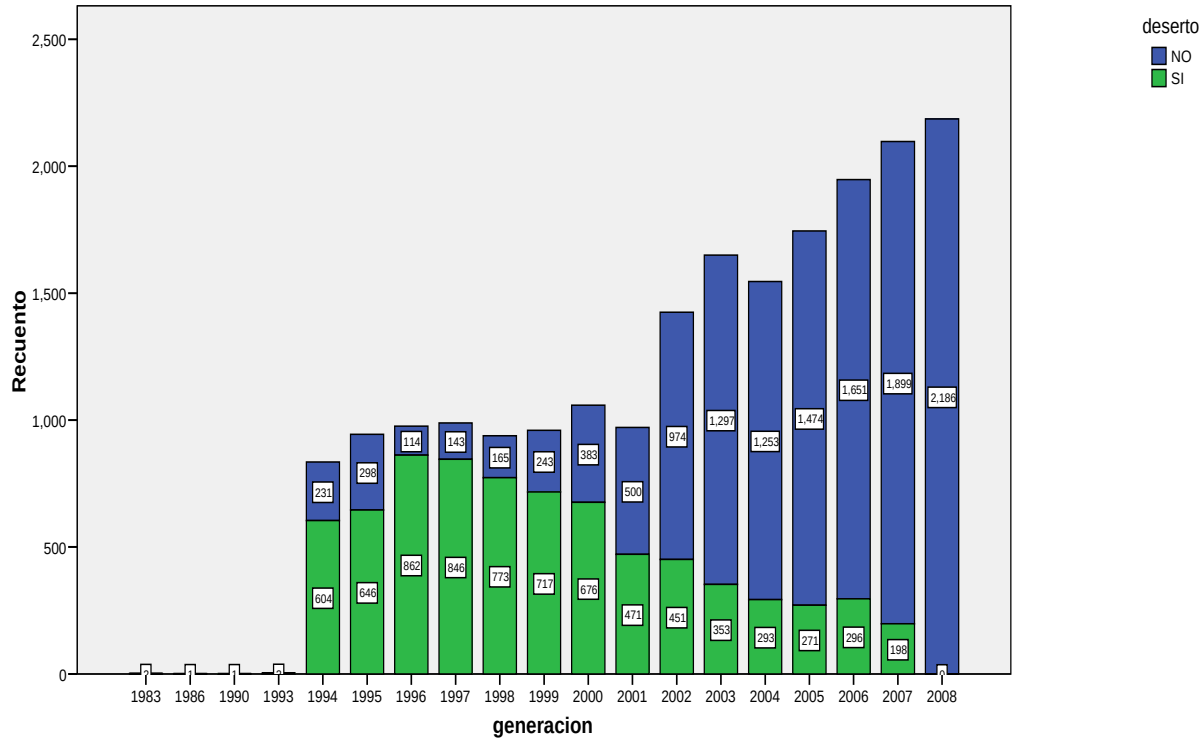


Figura 4.59. Gráfica de deserción hasta el semestre 2008-1.

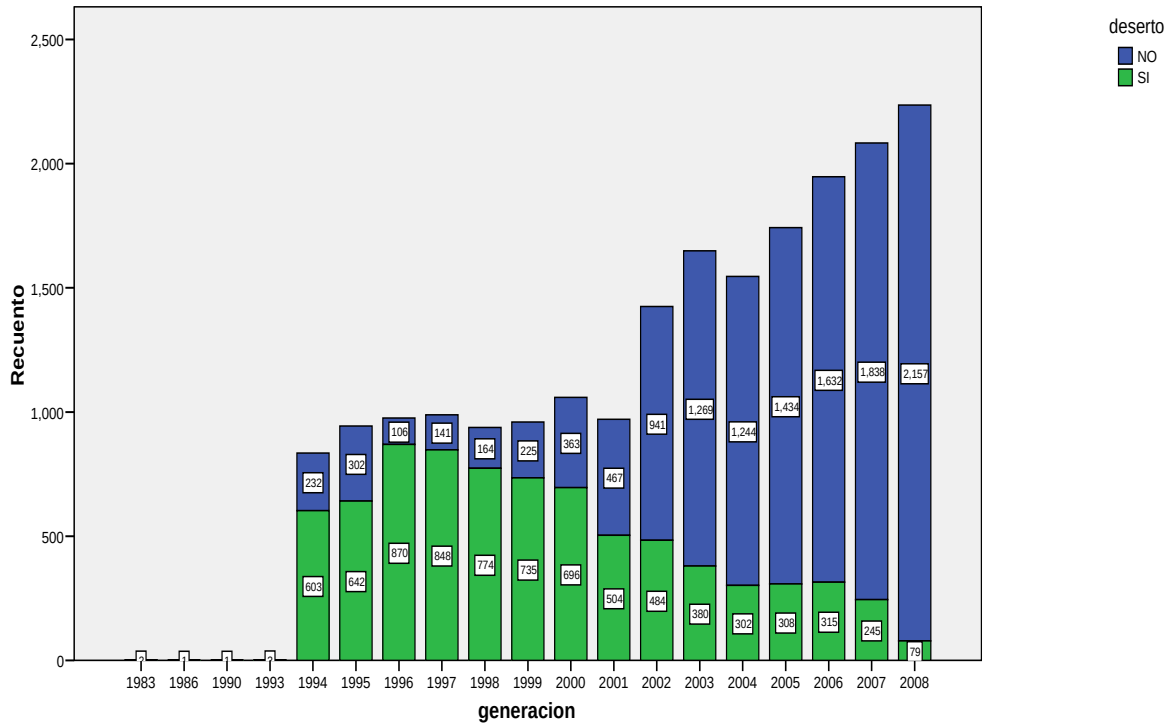


Figura 4.60. Gráfica de deserción hasta el semestre 2008-2.

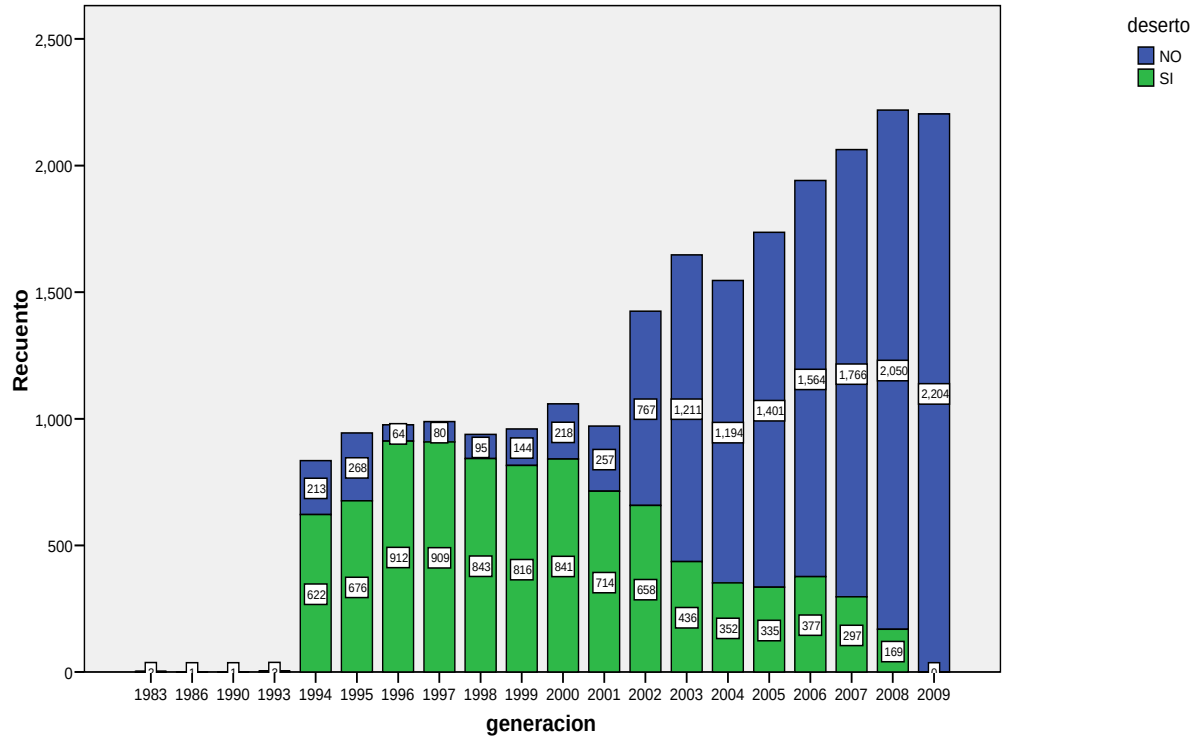


Figura 4.61. Gráfica de deserción hasta el semestre 2009-1.

Gráficas de terminación de materias:

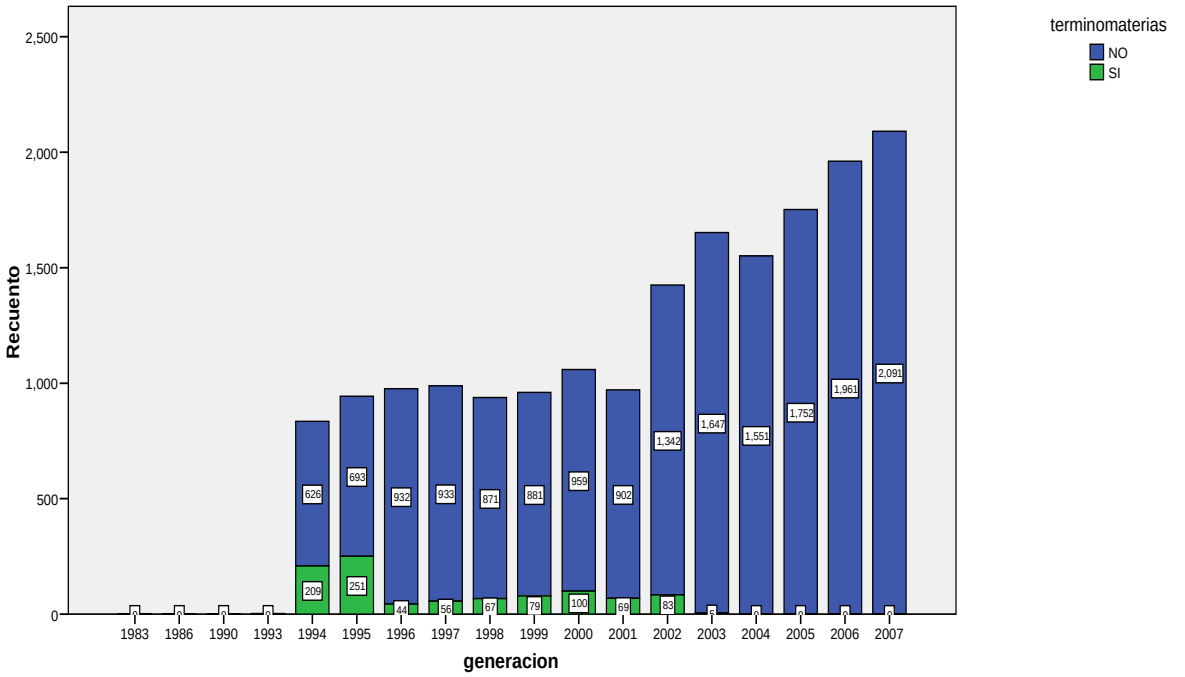


Figura 4.62. Gráfica de terminación de materias hasta el semestre 2007-1.

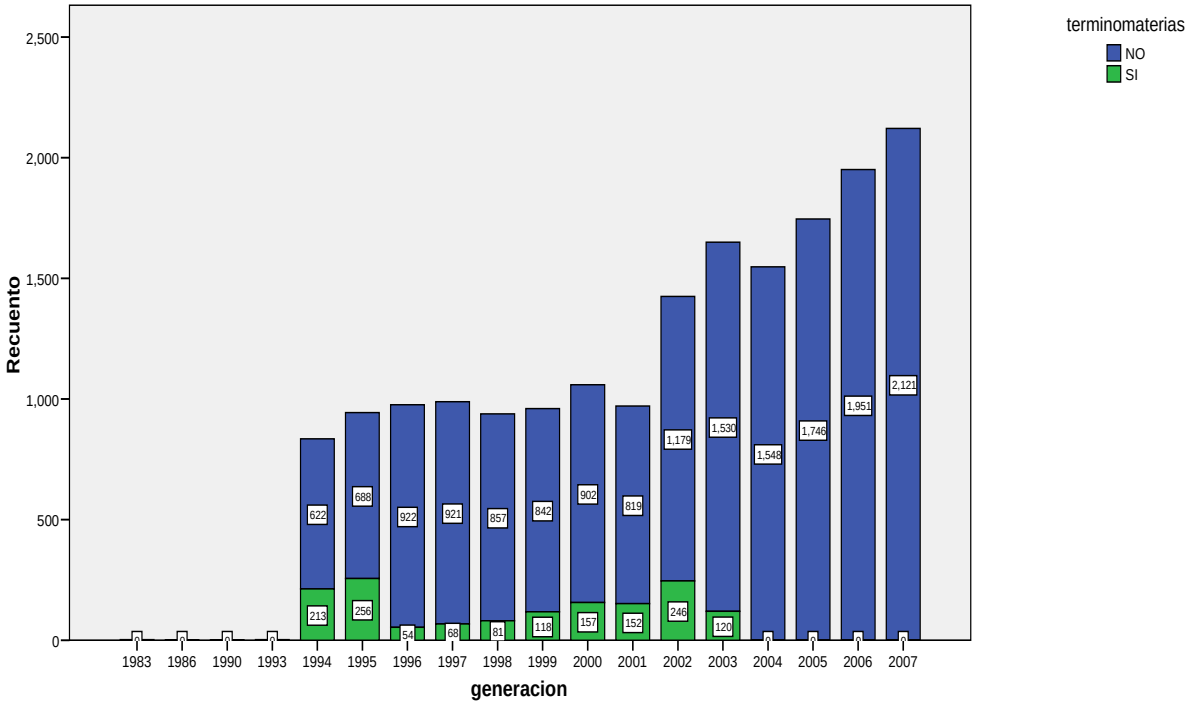


Figura 4.63. Gráfica de terminación de materias hasta el semestre 2007-2.

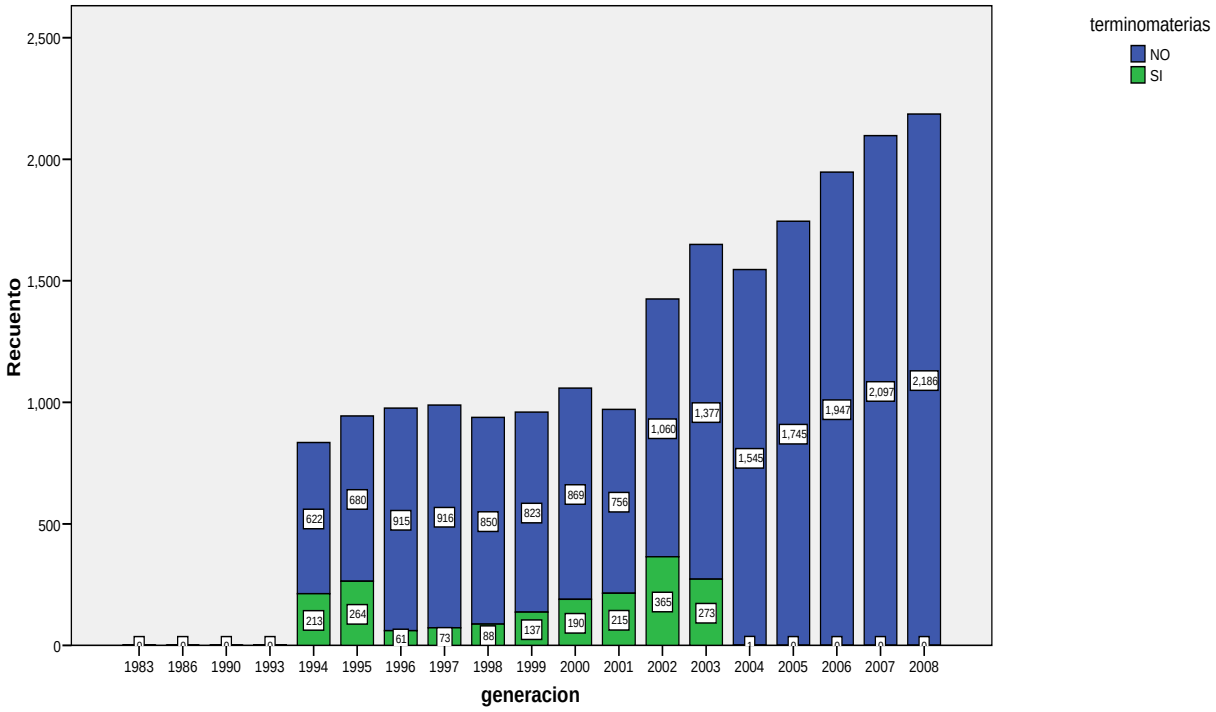


Figura 4.64. Gráfica de terminación de materias hasta el semestre 2008-1.

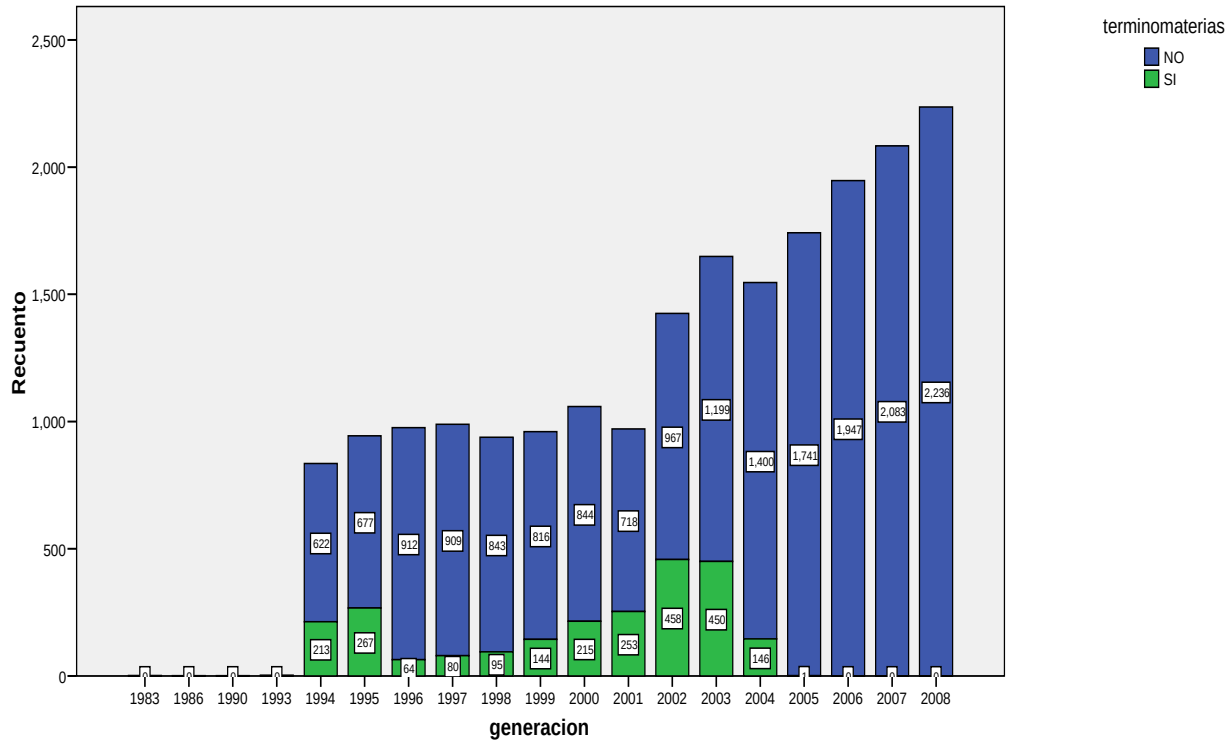


Figura 4.65. Gráfica de terminación de materias hasta el semestre 2008-2.

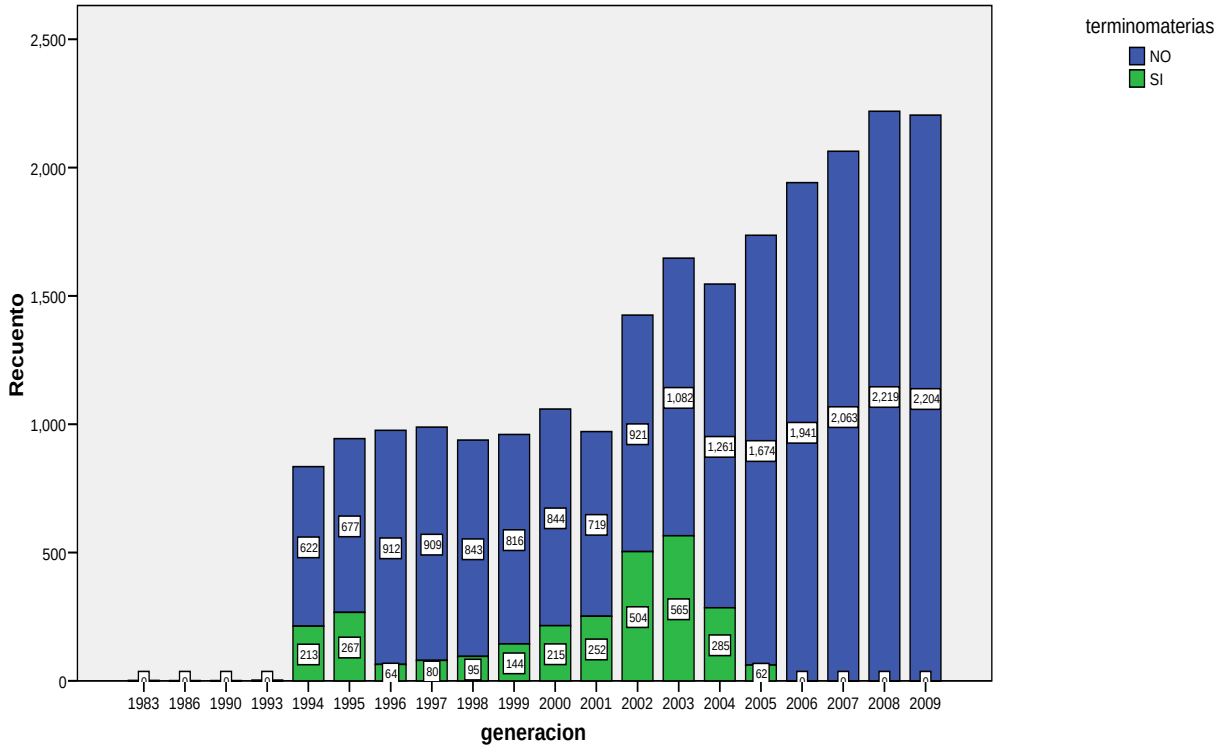


Figura 4.66. Gráfica de terminación de materias hasta el semestre 2009-1.

Con estas gráficas se puede ir observando durante 5 semestres cómo va cambiando el número de deserciones, el número de alumnos que van terminando todas sus materias por generación. También estas gráficas sirven para observar tendencias.

La minería de datos, además de encontrar patrones de comportamiento y tendencias, también sirve para la predicción de las mismas ya que al observar las gráficas bien se aplica asimismo para el entrenamiento de modelos y así poder predecir o anticiparse a lo próximo que va a suceder.

Se recalca mucho esta idea de poder predecir lo próximo. Si bien es cierto que predecir lo que sucederá a largo plazo es muy difícil de llevar a cabo, predecir algo a corto plazo es más factible. Sin embargo, sí es posible poder predecir cosas a largo plazo. La factibilidad dependerá de con cuántos datos históricos se cuentan y qué tan bien se entrene el modelo.

El éxito del buen aprendizaje del modelo no sólo depende de los datos, sino también de nuestro ingenio (en este caso, del ingeniero de datos o ingeniero de minería de datos) para limpiarlos y acomodarlos bien de tal forma que el modelo capte bien lo que precisamente se quiere que aprenda.

Además de contar con sólidos conocimientos en Bases de Datos, es bueno también tenerlos (o al menos buenas nociones) en programación de cualquier lenguaje ya que se comprende mejor cómo trabaja (o incluso hasta cómo piensa) la computadora. Con base en esto, se puede tomar una mejor decisión de cómo ir entrenando el o los modelos para su mejor aprendizaje.

Dado todo el análisis previo de estadística, se decidió aplicar los algoritmos para la clasificación donde se realizarán las tareas de predicción y las reglas de asociación. La primera se llevará a cabo por del algoritmo IBK y la segunda por medio de árboles de decisión.

Todo esto se irá explicando en el siguiente apartado en donde se habla y se explica el uso del programa para Minería de Datos que se ha estado utilizando: *Rapidminer*.

4.3.1. Encontrando reglas (árboles de decisión) y haciendo predicciones (algoritmo IBk).

Esta parte del estudio pretende, como su nombre lo indica, tratar de predecir por medio de reglas encontradas, a los alumnos que tienden a desertar. Primero se generaran reglas para la deserción enfocándonos a las generaciones con ayuda del algoritmo de árboles de decisión *Random Forest* para después aplicar el algoritmo de clasificación *IBk*.

Cabe resaltar que en la teoría se estipula que, después de la fase de Minería de Datos, se realizan las fases de Obtención de Patrones; después la de Evaluación, Interpretación, Visualización y, por último, la de Difusión y uso del conocimiento.

Una vez que se han analizado todas las gráficas, se decide qué consultas utilizar para el análisis de los datos generando las *vistas minables* (tablas o conjuntos de datos sobre los que se realizará el análisis – *Minería de Datos*) llegando a la obtención de los modelos. Se empieza con el análisis de la tabla *alumnodesercion4* (la cual contiene todos los datos más recientes) seleccionando todos los datos que contiene pero seleccionando sólo las columnas necesarias y por generación. En este caso, se seleccionan las columnas *cuenta*, *pln_dgae*, *plan*, *creditos*, *periodos*, *generacion*, *terminomaterias*, *promedio*, *hareprobado*, *haaprobado*, *deserto*.

Se hicieron consultas por generación. Se hizo una consulta para las generaciones 1994 y 1995, otra para las generaciones 1996, 1997 y 1998, 1998 y 1999 y otra para las generaciones 2000 y 2001. Después se obtienen consultas por generación desde la 2002 hasta la 2009. El porqué al principio se eligen de dos o tres generaciones, es porque viendo la gráfica en la *figura 4.47*, de las primeras generaciones, casi tienen el mismo número de deserciones además de que en las primeras generaciones se cuentan con menos datos por generación. A veces para un análisis es mejor tener un buen número de datos. Se hizo al principio estrictamente por generación pero no se obtenían tantas reglas interesantes por lo que se fue probando.

En el caso que se eligió tres generaciones y que la misma generación 1998 se usó en dos consultas distintas también demuestra la gran flexibilidad y versatilidad para utilizar el

algoritmo *Bosque Aleatorio* o *Random Forest* para encontrar patrones de comportamiento.

Se fueron ejecutando las consultas por generación, por ejemplo:

```
SELECT cuenta, pln_dgae, plan, creditos, periodos, generacion, deserto, terminomaterias,
promedio, hareprobado, haaprobado
FROM alumnodesercion4
WHERE generacion IN (1994, 1995)
```

Se tienen los datos como los mostrados en la *figura 4.67*. Una vez hecho esto se procede a exportar los datos obtenidos a un archivo en formato *csv* o separado por coma.

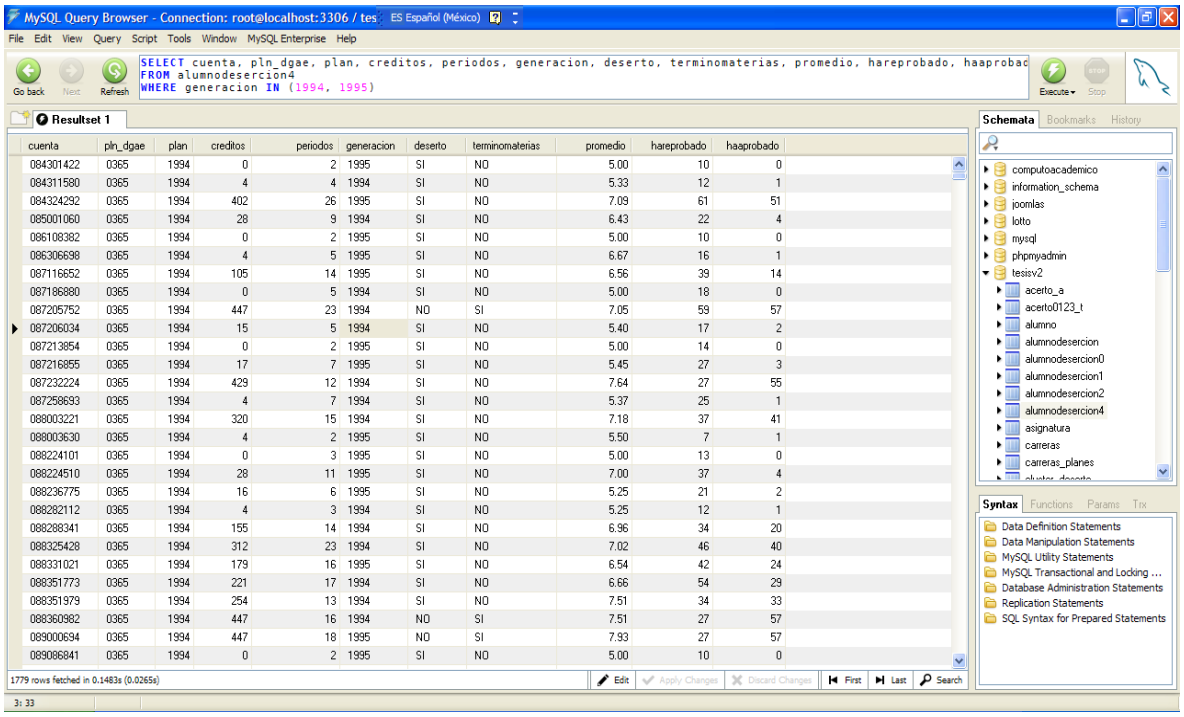


Figura 4.67. Visualizando todos los datos de la tabla *alumnodesercion* según las columnas seleccionadas.

Para exportarlos al formato en *.csv* se selecciona del menú *File*, *Export Resultset*, *Export As Excel File* como se muestra en la *figura* 4.68. Se elige una ruta y un nombre de archivo para guardar los datos. Posteriormente se procede a construir el árbol de procesos en *Rapidminer* para encontrar patrones de comportamiento en estas consultas.

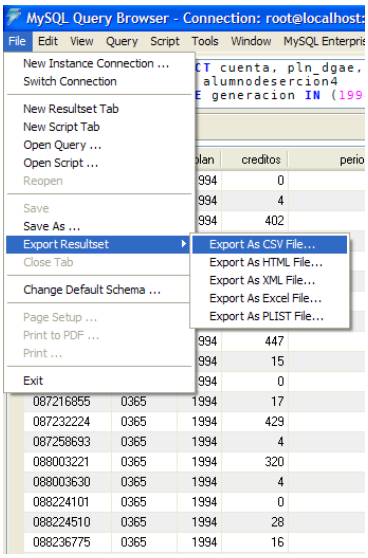


Figura 4.68. Exportando los datos a un archivo de *Excel*.

Entonces se ejecuta *Rapidminer*:

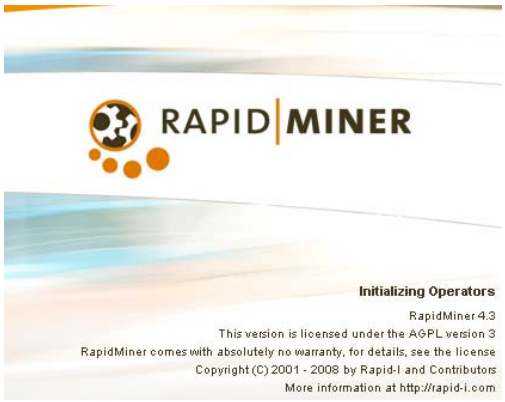


Figura 4.69. Arrancando *Rapidminer 4.3*.

En este caso se usa la versión 4.3. *Rapidminer* muestra el siguiente menú de bienvenida (*figura 4.70*):

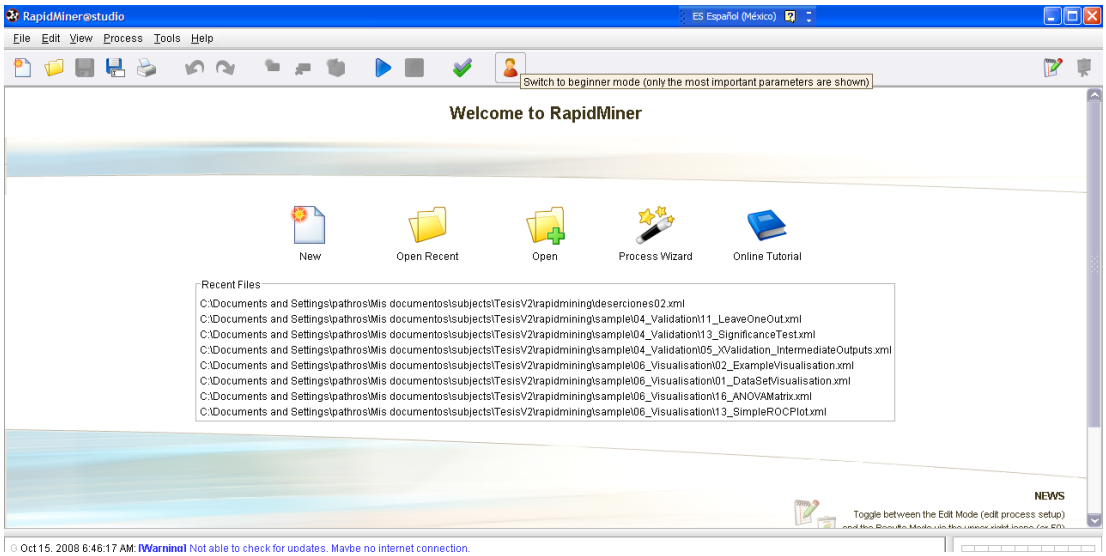


Figura 4.70. Pantalla de bienvenida de *Rapidminer*.

De la *figura 4.70*, en la parte de arriba, hay que dar clic en la parte de modo principiante / experto de tal forma que quede en modo experto. Esto es con el fin de que se puedan mostrar todos los parámetros. Dar clic en nuevo (o *new*).

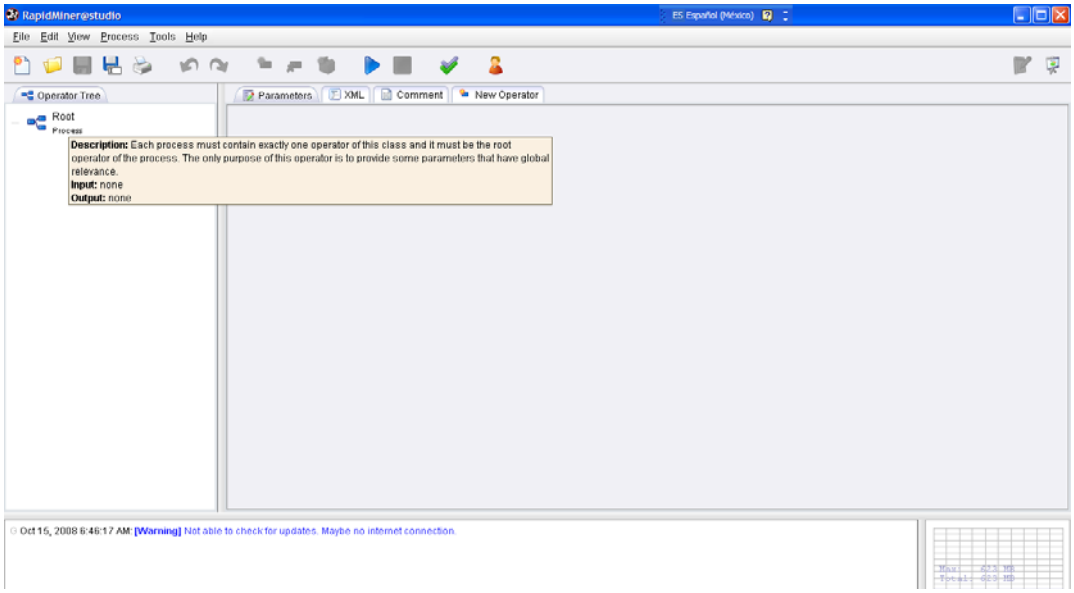


Figura 4.71. Ventana principal de *Rapidminer*.

Un experimento básico en Rapidminer, de acuerdo a lo mencionado en la referencia (28), consiste de los siguientes operadores:

1. La raíz del árbol de procesos o *root*.
2. El cargador de los datos (data loader)
3. El visualizador de datos (data visualizer)
4. El que valida los datos (data validator)
5. El creador del modelo (model creator)
6. El escritor del modelo (model writer)
7. El que aplica el modelo (model applier)
8. El evaluador del desempeño (performance evaluator)
9. El experimento final (Final Experiment).

Se probaron distintos algoritmos de los que contiene integrados *RapidMiner* y se lograron encontrar patrones interesantes usando el algoritmo del *Bosque Aleatorio* o *Random Forest*. Se obtuvieron los diagramas de los árboles los cuales sirven para visualizar de manera general las reglas encontradas.

El algoritmo *Random Forest* obtiene un número *N* (parámetro que puede ser modificado por el usuario) de árboles. Nos ha parecido un buen instrumento para obtener muchísimos patrones de comportamiento ya que, por ejemplo, con un *árbol de decisión* uno elige la variable etiqueta (sobre la cual predecir o encontrar patrones, por ejemplo, si desertó o no), este árbol sólo obtiene un árbol y sólo elige las primeras variables que encuentra. Es decir, obtiene un árbol de deserción según el promedio y el número de créditos obtenidos. ¿Y las demás variables dónde quedaron? ¿El resto de las variables no intervienen? Estas dudas se generan porque el árbol, por más cambio en los parámetros que se haga, seguirá saliendo el mismo árbol. Además, el árbol de decisión tarda más tiempo en ejecutarse conforme el número de datos vaya aumentando.

Random Forest obtiene más árboles, por ejemplo, se escogen 10. De estos árboles, 6 son los mismos, pero los cuatro restantes obtienen reglas de deserción según las distintas variables, por ejemplo, un árbol con variables de promedio y créditos, otro con variables de plan de estudios, materias aprobadas y reprobadas, etc. Lo cual hace posible que se pueda visualizar bien cómo las demás variables van influyendo en la columna etiqueta (*label column*), en este caso, la deserción.

Otra gran ventaja de usar este algoritmo es que su tiempo de ejecución es verdaderamente corto incluso si se elige un número de 50 árboles. Entre más árboles se tengan, más reglas se pueden encontrar y es que en algunos casos se obtienen árboles demasiado grandes o *frondosos*. Estos árboles sirven muy bien para la detección de casos específicos extraordinarios o anómalos. Pero también se pueden obtener árboles muy pequeños pero contundentes. Es decir, que aunque se encuentran pocas reglas y, a primera vista, muy generales, esas reglas encontradas bastan para entender, a manera de panorama, lo que sucede.

Primeramente se van a obtener patrones de comportamiento con *Random Forest* usando un árbol de procesos en *Rapidminer* sencillo, sin validación. El árbol de procesos típico que se usa en *Rapidminer* se usará en la parte de predicciones. En este caso sólo se necesitan patrones de comportamiento los cuales pueden comprobarse usando consultas SQL.

Se procede a crear el árbol de procesos en *Rapidminer*. En la figura anterior, la *figura 4.71*, se tiene el árbol de procesos. Hacer *clic* con el botón derecho del *mouse*. Del menú contextual seleccionar *New Operator*, *IO*, *Examples*, *CSVExampleSource*. *Figura 4.72*.

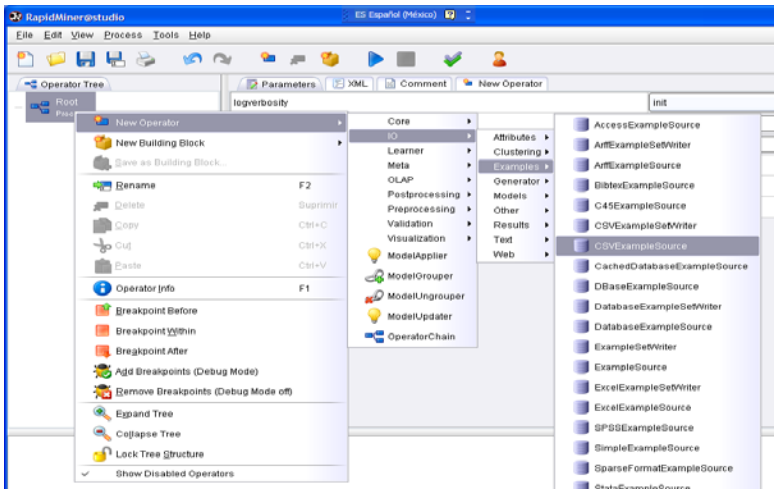


Figura 4.72. Seleccionando un archivo de fuente del tipo separado por coma o *.csv*.

De la misma manera haciendo *clic* con el botón derecho del *mouse* sobre el operador *root*, seleccionar *New Operator*, *Visualization*, *ExampleVisualizer*. Esto con el fin de poder visualizar los datos con datos estadísticos y poder verificar que los datos que se quieren cargar sean precisamente esos y que estén completos. *Figura 4.73*.

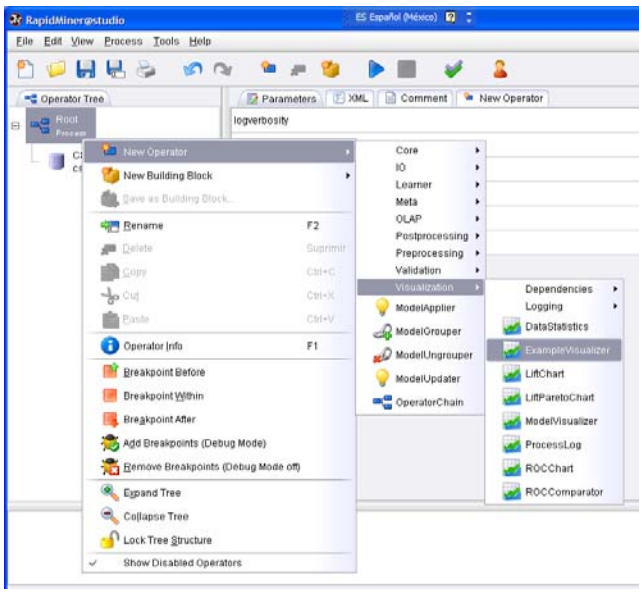


Figura 4.73. Seleccionando el visualizador de los datos cargados.

Igualmente del menú contextual se procede a elegir el algoritmo *Random Forest*. Del menú contextual seleccionar *New Operator*, *Learner*, *Supervised*, *Trees*, *RandomForest*. *Figura 4.74*.

Configurar los parámetros de cada uno de los operadores. Sólo señalar con el *mouse* cada operador. Señalando el operador *CSVExampleSource*, en la ventana del lado derecho se observan los parámetros. *Figura 4.75*.

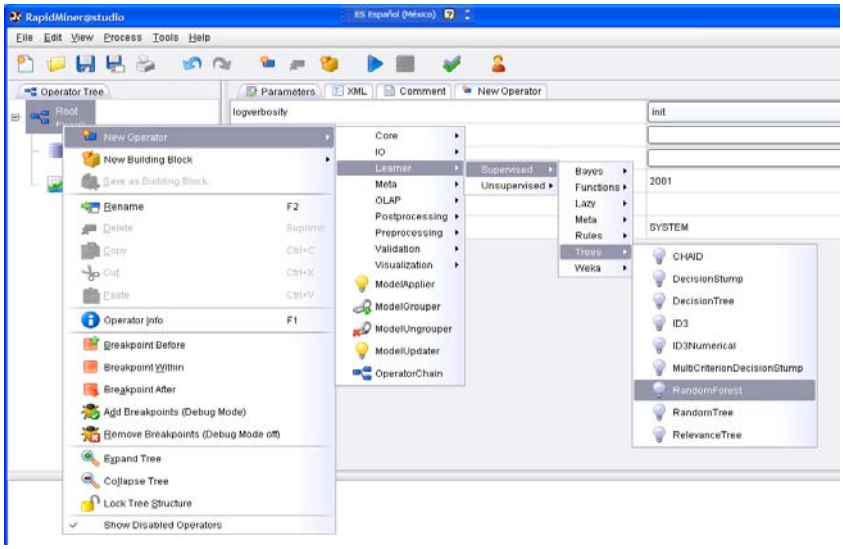


Figura 4.74. Seleccionando el algoritmo Random Forest o Bosque Aleatorio.

Se tiene que especificar la ruta del archivo *.csv* tecleando en el botón que tiene tres puntos suspensivos. De ahí se abre una nueva ventana en la que sólo se tiene que buscar el archivo y seleccionarlo. Después de esto, en la ventana de parámetros se mostrará el nombre del archivo así como su ruta. En el parámetro *label_name* teclear *deserto* que es la variable de la que se quiere saber y encontrar patrones. En *label_column* hay que teclear el número de la columna (de izquierda a derecha empezando desde 1) en donde se encuentra *deserto*. En este caso, teclear la columna 7. En el parámetro *ID_name* teclear la columna que señala al identificador, en este caso es el número de cuenta por lo que teclear *cuenta* y su correspondiente número de columna en el archivo: 1 (es la primera columna). El resto de los parámetros se dejan como estan. Por defalut.

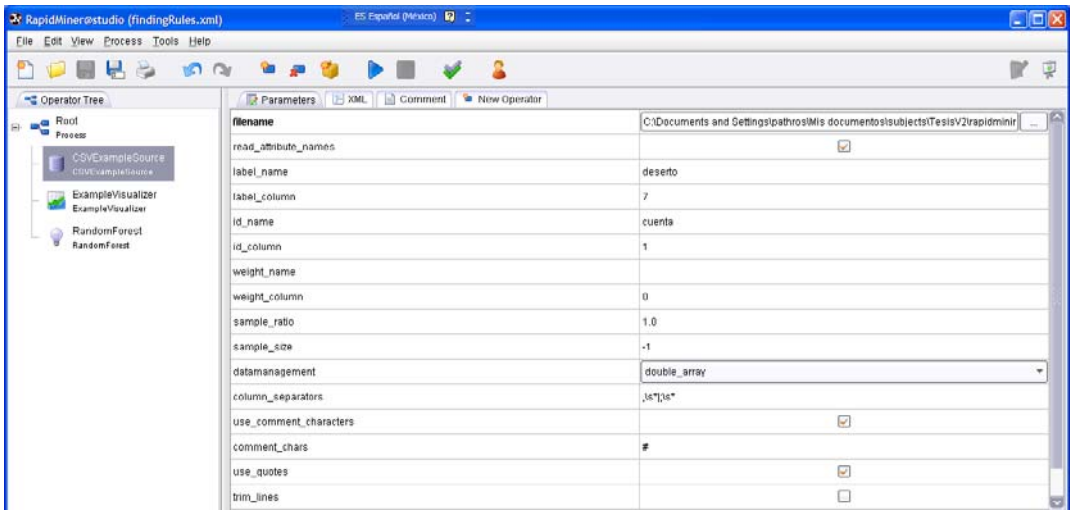


Figura 4.75. Modificando los parámetros de *CSVExampleVisualizer*.

Ahora dar *clic* sobre *RandomForest* para visualizar los parámetros y modificarlos de acuerdo a nuestras necesidades. Seleccionar la primera opción *keep_example_set*, en el número de árboles o *number_of_trees* teclear 50. En el siguiente parámetro criterio o *criterion* seleccionar *gini_index*. En este parámetro se selecciona la manera en la que el bosque aleatorio encuenta los patrones. Haciendo varias pruebas, este índice es el que mejor encuentra patrones y favorece el crecimiento máximo de los árboles, pero también ayuda a encontrar árboles pequeños pero contundentes. Se explica en la parte teórica el funcionamiento de este tipo de índice y de este bosque aleatorio. El resto de los parámetros se dejan como están. Por default.

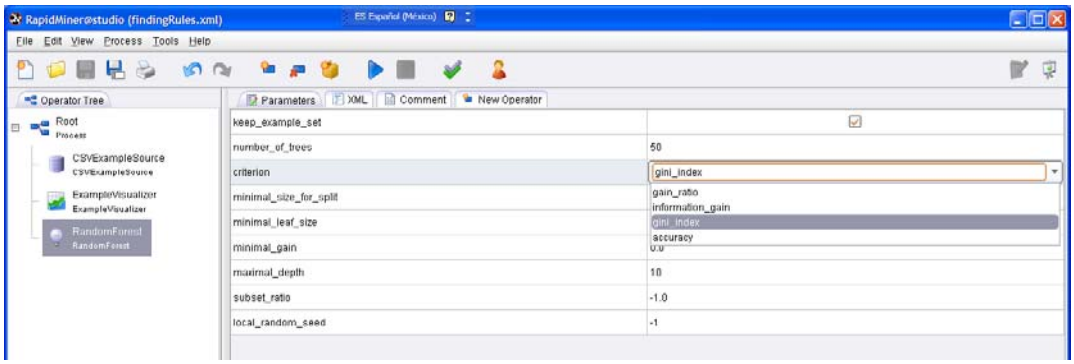


Figura 4.76. Modificando los parámetros de *RandomForest.*

Se ha terminado este árbol de procesos. Presionamos el botón de la palomita verde que se encuenta en el menú en la parte superior (marcado en la *figura 4.76*). Esto último sirve para comprobar que no existan errores en el árbol de procesos que se crea. En *Rapidminer* algunos parámetros requieren de operadores o nodos *hijos* o deben ser operadores *hijos*. Si los operadores no están bien ordenados, *Rapidminer* lo hará saber indicándonos el error.

Después de teclear la palomita se observa que todo está bien (*figura 4.77*).

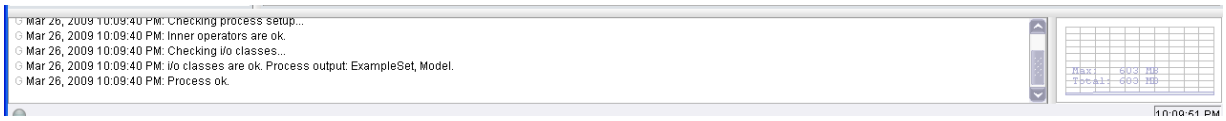


Figura 4.77. Verificando que el árbol de procesos esté bien.

Luego, de la *figura 4.76*, dar *clic* sobre la flechita azul de *play* para iniciar el proceso. A veces cuando se ha modificado el archivo, al momento de correr el proceso, *Rapidminer* pregunta si se desea guardar el proceso. Si es la primera vez, hay que especificar el nombre y la ruta del archivo en donde lo se quiere guardar. Si no, lo sobrescribe y simplemente continúa con el proceso. Véase la *figura 4.*78

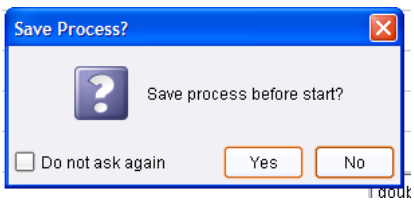


Figura 4.78. Se pregunta si se desea guardar el proceso o los cambios hechos en él.

Se observa en la ventana de abajo cómo va corriendo el proceso y esperamos, en este caso, algunos segundos hasta que termina de ejecutarse por completo (*figura 4.79*). También en la parte inferior derecha se observa el uso de la memoria de la computadora.

Figura 4.79. Observamos el processo corriendo.

Cuando termina, automáticamente *Rapidminer* cambia a la ventana de los resultados y se puede visualizar los árboles de decisión generados (*figura 4.80*). Si se quiere regresar a la ventana donde se muestra el árbol de procesos, basta con dar *clic* sobre el botón que dice *Change to the Edit Mode* el cual está en la parte superior derecha.

De entrada, en la misma *figura 4.80*, se observa uno de los árboles que se obtuvieron. Se tiene en total 50 árboles, ya que así se fijó el parámetro y se puede ir viendo cada uno de ellos seleccionando las pestañas de cada árbol. Cada árbol de hecho puede también visualizarse a manera de texto. Puede resultar difícil al principio entender cómo se va desglosando el árbol en texto, pero esta modalidad resulta ser más cómoda cuando se tienen árboles muy grandes.

También, arriba de las pestañas y del lado izquierdo que permiten visualizar un árbol individualmente, se tienen dos pestañas más: *SimpleVote* y *DataTable*. La primera corresponde a los árboles y la segunda a la visualización de los datos con los que se está trabajando en cuestión. Se pueden ver cuáles son los metadatos (información de los datos o en este caso, de las columnas), datos, así como poder realizar ciertos tipos de gráficas de los mismos.

Para cada archivo de generación se hizo el mismo procedimiento descrito anteriormente, se obtuvieron las reglas, se seleccionaron las más interesantes y concisas, se verificaron con consultas *SQL* e incluso de algunas reglas se investigó el porqué de esas reglas usando asimismo más consultas *SQL*. Estas reglas encontradas ayudan en buena medida a la explicación de las gráficas de las figuras de la 4.57 a la 4.66. Las reglas más

relevantes obtenidas fueron las siguientes (han sido escogidas las más interesantes y concisas):

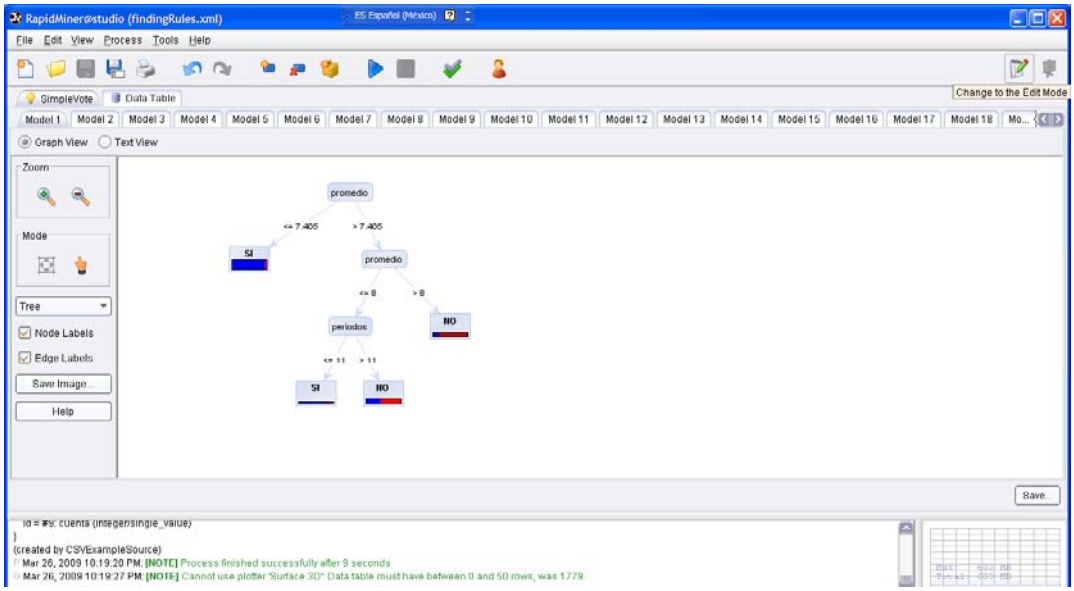
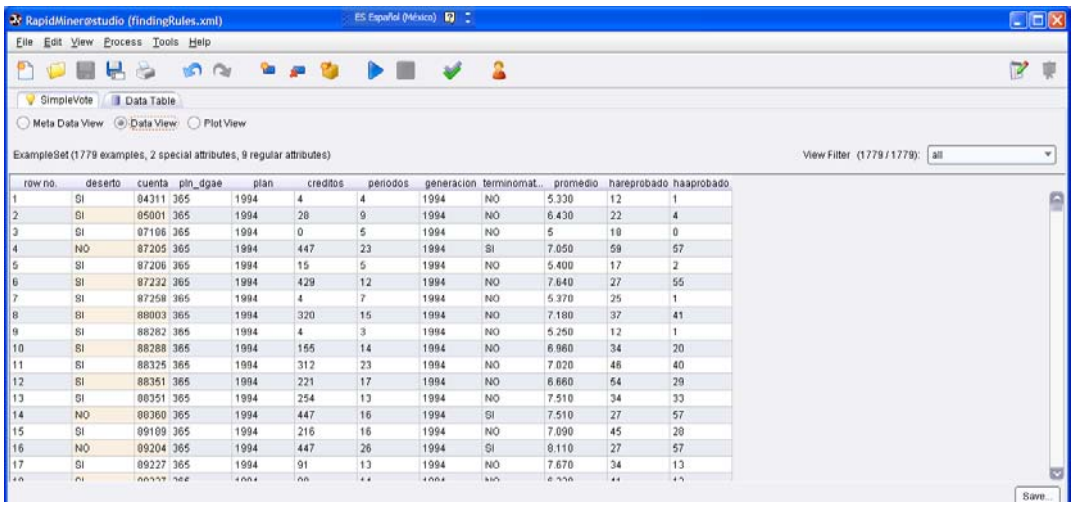


Figura 4.80. Se muestran los resultados al terminar de ejecutarse el proceso.



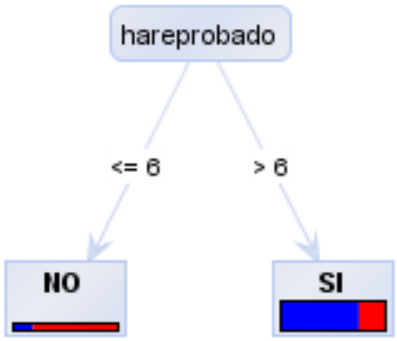
row no.	deserto	cuenta	pln_dgae	plan	creditos	periodos	generacion	terminomat...	promedio	hareprobado	haaprobado
1	SI	84311	365	1994	4	4	1994	NO	5.330	12	1
2	SI	85001	365	1994	28	9	1994	NO	6.430	22	4
3	SI	87106	365	1994	0	5	1994	NO	5	10	0
4	NO	87205	365	1994	447	23	1994	SI	7.050	59	57
5	SI	87206	365	1994	15	5	1994	NO	5.400	17	2
6	SI	87232	365	1994	429	12	1994	NO	7.640	27	55
7	SI	87258	365	1994	4	7	1994	NO	5.370	25	1
8	SI	88003	365	1994	320	15	1994	NO	7.180	37	41
9	SI	88282	365	1994	4	3	1994	NO	5.250	12	1
10	SI	88288	365	1994	155	14	1994	NO	6.960	34	20
11	SI	88325	365	1994	312	23	1994	NO	7.020	46	40
12	SI	88351	365	1994	221	17	1994	NO	6.660	54	29
13	SI	88351	365	1994	254	13	1994	NO	7.510	34	33
14	NO	88360	365	1994	447	16	1994	SI	7.510	27	57
15	SI	89189	365	1994	216	16	1994	NO	7.090	45	28
16	NO	89204	365	1994	447	26	1994	SI	8.110	27	57
17	SI	89227	365	1994	91	13	1994	NO	7.670	34	13

Figura 4.81. Visualizando los datos con los que se está trabajando.

Generaciones 1994 y 1995:

- De los que han reprobado hasta en 6 ocasiones, de 95 casos 76 no desertaron y 75 terminaron sus materias, los demás desertaron, algunos (en este caso, 5 alumnos) incluso faltándoles 1 materia ó 2 (*figura 4.82*).

- De los que han reprobado más de 6 veces, de 1684 casos, 1279 sí desertaron y el resto sí terminó sus estudios.



Árbol 1.

Y las consultas para corroborar la información obtenida en los árboles son:

SELECT * FROM alumnodesercion4
WHERE generacion IN (1994,1995)
AND hareprobado<=6
ORDER BY deserto,terminomaterias
SELECT * FROM alumnodesercion4
WHERE generacion IN (1994,1995)
AND hareprobado>6
ORDER BY deserto,terminomaterias

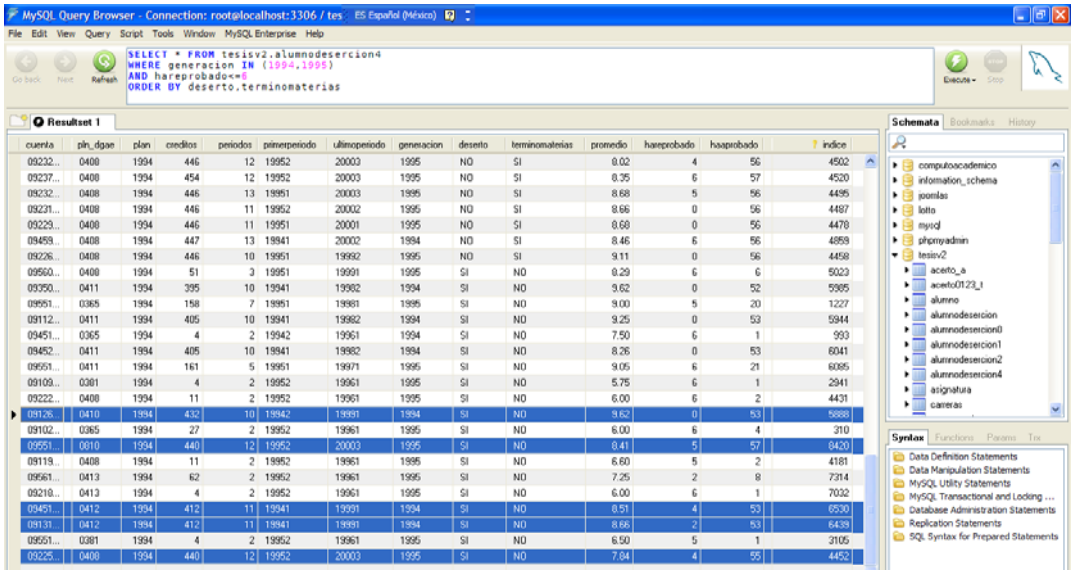
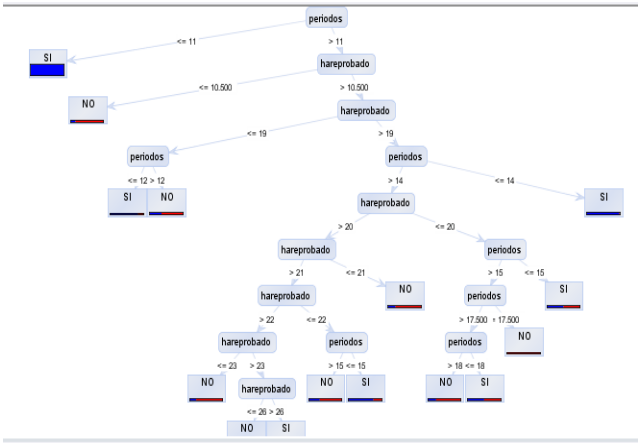


Figura 4.82. Corroborando y ampliando la explicación de una regla encontrada por el árbol 1.

En estas generaciones destaca que si un alumno reprueba 6 o menos veces, termina todas sus materias.

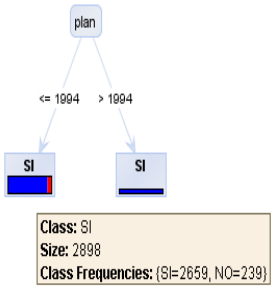
Generaciones 1996, 1997 y 1998:

- De 1942 casos, 1934 desertaron habiendo cursado hasta 11 semestres. Y hasta 14 semestres cursados 180 casos de 183.



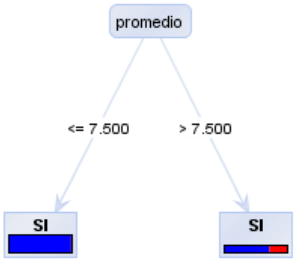
Árbol 2.

- De estas generaciones, los que se cambiaron de plan (al 2006), todos desertaron. (5 casos).



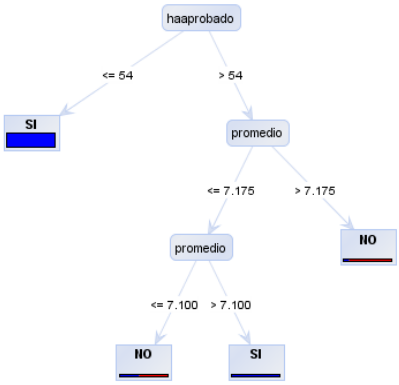
Árbol 3.

- De 2196 casos, 2160 desertaron con un promedio menor a 7.5.

- 

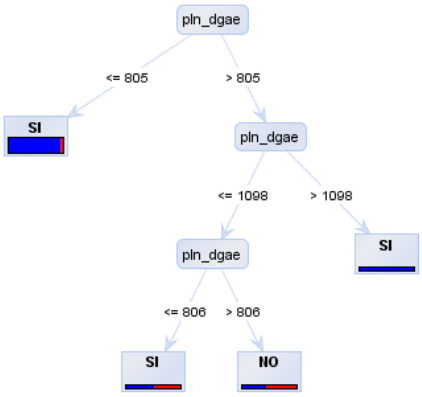
Árbol 4.

- Los que no desertaron tuvieron un promedio mayor a 7.175 (204 de 231 casos).

- 

Árbol 5.

- De las carreras en las que hubo menor deserción se ubican en las de ing. mecánica (módulos mejor ambiental, biomédicas y mecatrónica), ing. industrial (producción y administración y sistemas) e ingeniería eléctrica-electrónica.

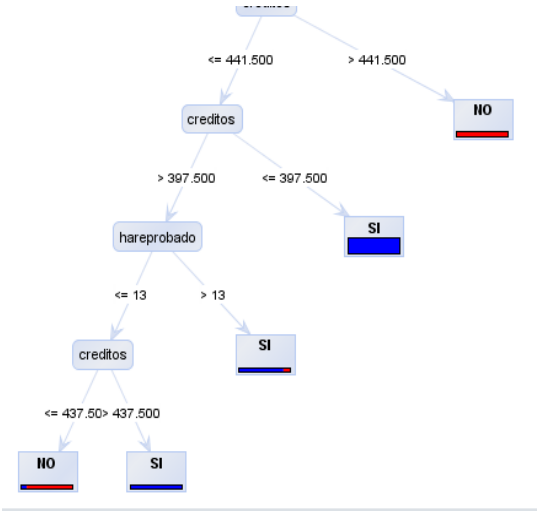
- 

Árbol 6.

Para estas generaciones, hubo mucha deserción. Se destaca que los que terminaron todas sus materias lo hicieron con un promedio mayor a 7.175.

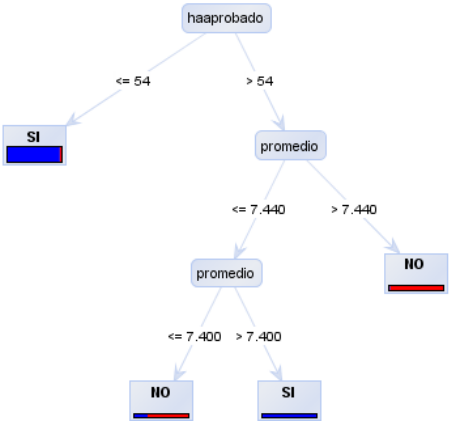
Generaciones 1998 y 1999:

- En las generaciones **98 y 99**, de 1574 casos, 1573 desertaron con 397 créditos o menos.



Árbol 7.

- Los que han aprobado más de 54 materias (el mínimo para algunas carreras) y con promedio mayor a 7.44 sí terminaron todas sus materias – 179 de 186 casos. Los que desertaron, tan sólo les faltaban unos cuantos créditos para terminar (mínimo tienen 431 créditos) y su promedio era de entre 7.5 y 8.4. Todos, en este caso, del plan 1994.



Árbol 8.

SELECT * FROM alumnodesercion4

WHERE generacion IN (1998,1999)

AND haaprobado>54

AND promedio>7.440

ORDER BY deserto,terminomaterias

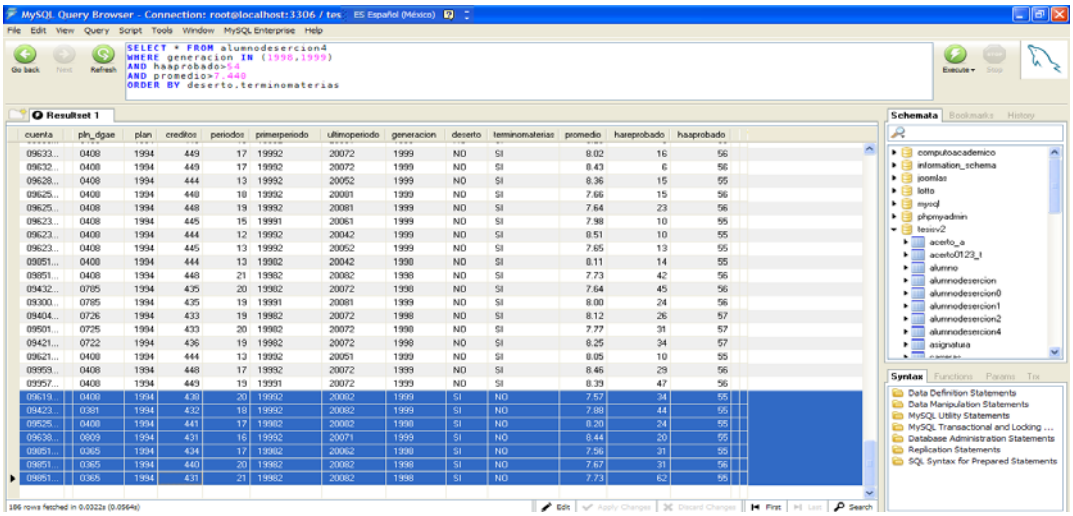
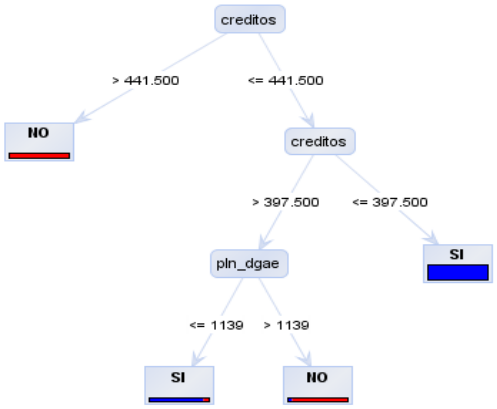


Figura 4.83. Corroborando y ampliando la explicación de una regla encontrada por el árbol 8.

- De 24 casos, 22 terminaron sus materias al cambiarse de plan (al 2006), los 2 que desertaron les faltaban pocos créditos para terminar pero ya tenían 19 y 20 semestres cursados con promedio de 7 y 8.



-

Árbol 9.

SELECT * FROM alumnodesercion4

WHERE generacion IN (1998,1999)

AND creditos BETWEEN 398 AND 441

AND pln_dgae>1139

ORDER BY deserto,terminomaterias

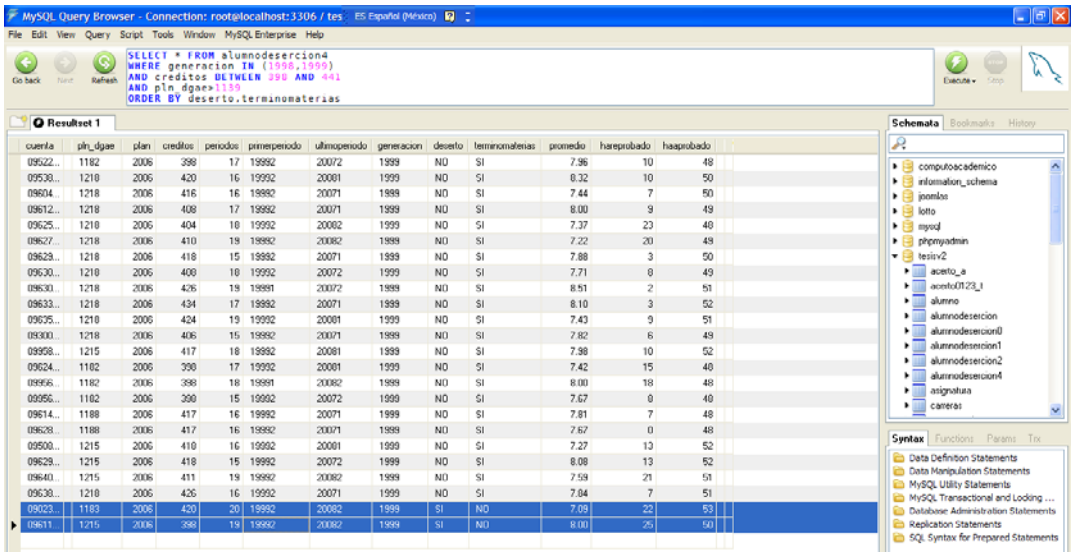
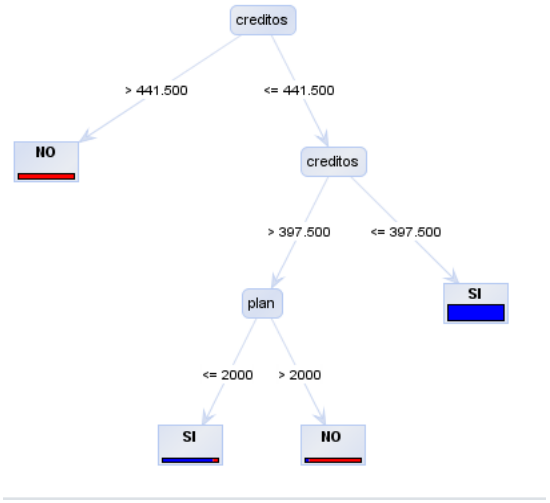


Figura 4.84. Corroborando y ampliando la explicación de una regla encontrada por el árbol 9.

En el siguiente árbol de decisión se observa en la parte de abajo de los que tenían más de 397 créditos, los que cambiaron de plan, no desertaron logrando terminar sus materias, 22 de 24 casos. Y los que permanecieron en el viejo plan, de 92 casos, 84 desertaron pero estando cerca de terminar sus materias, el resto sí terminó.



Árbol 10.

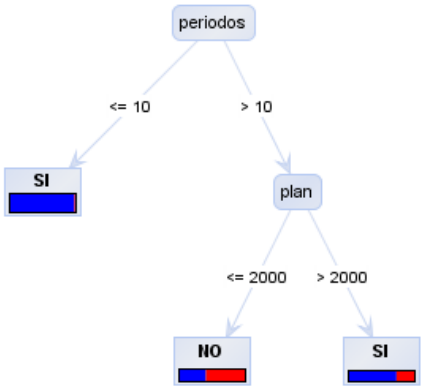
SELECT * FROM alumnodesercion4
WHERE generacion IN (1998,1999)
AND creditos BETWEEN 398 AND 441
AND plan<=2000
ORDER BY deserto,terminomaterias

En estas generaciones se tiene que los que se cambiaron al plan de estudios 2006, normalmente terminaron todas sus materias.

Generaciones 2000 y 2001:

- De los que cursaron menos de 10 semestres, de 998 casos, 985 desertaron, los 13 restantes no desertaron de los cuales 2 siguen cursando.
- De los que han cursado más de 10 semestres y estando en el plan 1994, de 593 casos, 346 no desertaron, 345 ya terminaron sus materias.
- Los que se cambiaron al plan 2006, de 439 casos, 323 desertaron, 5 siguen cursando y los restantes 111 sí terminaron sus materias.

Estas últimas tres reglas se observan en el *árbol 11* y las respectivas consultas que se hacen para corroborar y entender mejor las reglas se muestran a continuación:



Árbol 11.

SELECT * FROM alumnodesercion4
WHERE generacion IN (2000,2001)
AND periodos<=10
ORDER BY deserto,terminomaterias

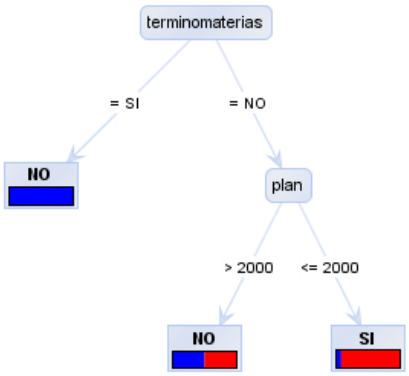
SELECT * FROM alumnodesercion4
WHERE generacion IN (2000,2001)
AND periodos>10
AND plan>2000
ORDER BY deserto,terminomaterias

SELECT * FROM alumnodesercion4
WHERE generacion IN (2000,2001)
AND periodos>10
AND plan<=2000
ORDER BY deserto,terminomaterias

En estas generaciones sucede lo contrario. Hubo más alumnos que terminaros todas sus materias en el plan 94 que en el plan 2006.

Generación 2002:

- De los que no han acabado todas sus materias aún, de los que están en el plan 2006, poco más de la mitad no ha desertado (217 contra 209 casos que sí desertaron).
- De los que siguieron cursando el plan anterior (1994), de un total de 495 sólo 46 no han desertado.
- De los 263 que siguen cursando, sólo 45 rebasan los 390 créditos, y la mayoría ya lleva cursados 13 ó 14 semestres, sabiendo que 15 semestres es el tiempo límite. Después deben presentar solamente extraordinarios.



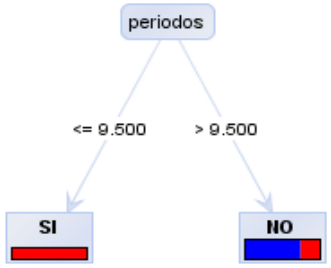
Árbol 12.

SELECT * FROM alumnodesercion4
WHERE generacion=2002
AND plan>2000
AND terminomaterias='NO'
AND deserto='NO'
ORDER BY deserto,terminomaterias

SELECT * FROM alumnodesercion4
WHERE generacion=2002
AND terminomaterias='NO'
AND deserto='NO'
ORDER BY deserto,terminomaterias,creditos

- De 417 casos, 419 desertaron habiendo cursado hasta 9 semestres, la mayoría con menos de 15 créditos y a lo mucho un caso con 285 créditos. Pero los 8 que no han desertado tienen pocos créditos, a lo mucho 164 créditos y bajo promedio.

- Por el otro lado, de los que llevan cursado más de 9 semestres, de 1008 casos, 759 casos no han desertado, de los cuales, 504 ya terminaron sus materias. Y de los que no han desertado, 50 alumnos tienen altas probabilidades de terminar dado que rebasan los 386 créditos y en algunas carreras sólo piden 400 créditos para terminar.



Árbol 13.

SELECT * FROM alumnodesercion4
WHERE generacion=2002
AND periodos>9.5
ORDER BY deserto,terminomaterias,creditos

SELECT * FROM alumnodesercion4
WHERE generacion=2002
AND periodos<=9.5
ORDER BY deserto,terminomaterias,creditos

Para esta generación los que llevan cursados más de 9 semestres, normalmente terminan todas sus materias.

Generación 2003:

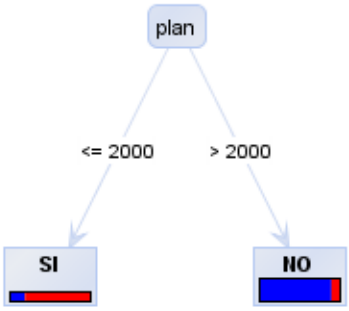
- De los 565 que terminaron sus materias, 6 terminaron en 9 semestres. Y el resto terminó en un rango de 10 a 14 semestres.

SELECT * FROM alumnodesercion4
WHERE generacion=2003
AND terminomaterias='SI'
ORDER BY deserto,terminomaterias,periodos

En esta generación, en donde se tiene el mayor número de alumnos que sí terminan todas sus materias, la mayoría terminó habiendo cursado entre 10 y 14 semestres.

Generación 2004:

- De 285 casos que están en el plan 1994, 229 desertaron. De los 56 que no desertaron, 16 terminaron sus materias. 13 rebasan los 300 créditos por lo que se les ve posibilidades de terminar.
- De los 1261 que cursan en el plan nuevo, 123 desertaron, de los cuales 13 tenían más de 400 créditos. De los 1138 que no desertaron, 269 han terminado sus materias.



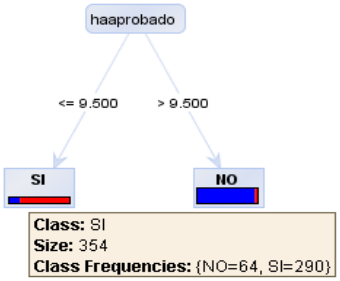
Árbol 14.

SELECT * FROM alumnodesercion4
WHERE generacion=2004
AND plan<=2000
ORDER BY deserto,terminomaterias,creditos

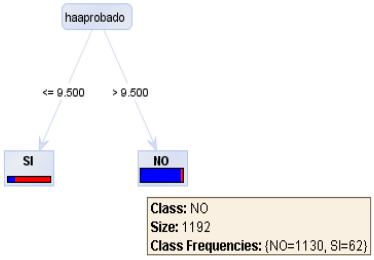
SELECT * FROM alumnodesercion4
WHERE generacion=2004
AND plan>2000
ORDER BY deserto,terminomaterias,creditos

SELECT * FROM alumnodesercion4
WHERE generacion=2004
AND plan>2000
AND terminomaterias='SI'
ORDER BY deserto,terminomaterias,creditos

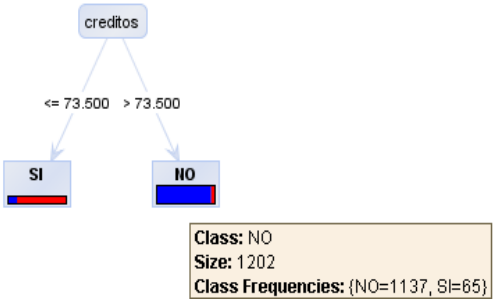
- De los 352 alumnos que han desertado, 290 desertaron habiendo aprobado menos de 9 materias; 267 desertaron habiendo cursado menos de 9 semestres; 287 desertaron teniendo menos de 74 créditos.



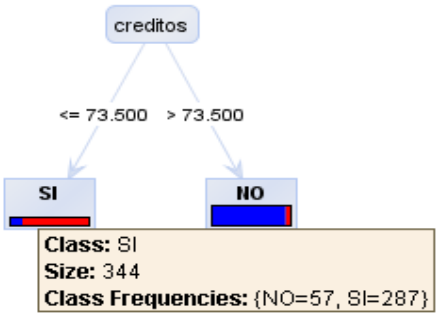
Árbol 15a.



Árbol 15b.



Árbol 16a.

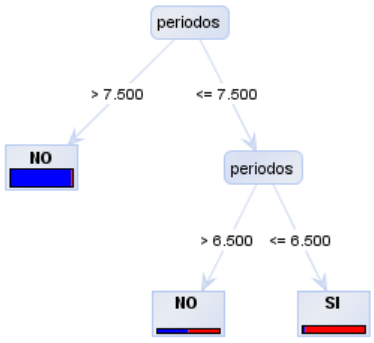


Árbol 16b.

En esta generación han desertado más alumnos en el plan anterior (1994) que los que están en el plan 2006.

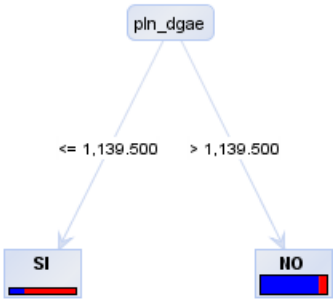
Generación 2005:

- De los 335 casos de deserción hasta ahora, 41 desertaron habiendo cursado 7 semestres y 270 menos de 7 semestres. 281 aprobaron hasta 7 materias.



Árbol 17.

- Si bien han desertado más en el plan nuevo (176 contra 159 del viejo plan), la proporción de deserción es menor en el nuevo plan.



Árbol 18.

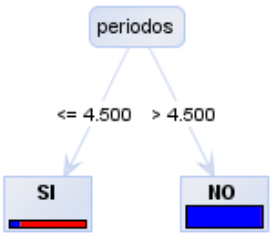
Generación 2006:

- De 377 alumnos que desertaron, 372 lo hicieron habiendo cursado menos de 6 semestres. 322 aprobaron hasta 7 materias, 321 con menos de 61 créditos. 53 desertaron habiendo reprobado entre 15 y 21 materias.
-

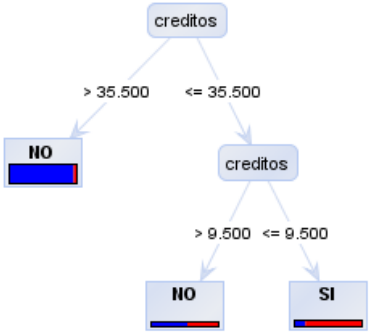
Árboles 19a, 19b, 19c y 19d.

Generación 2007:

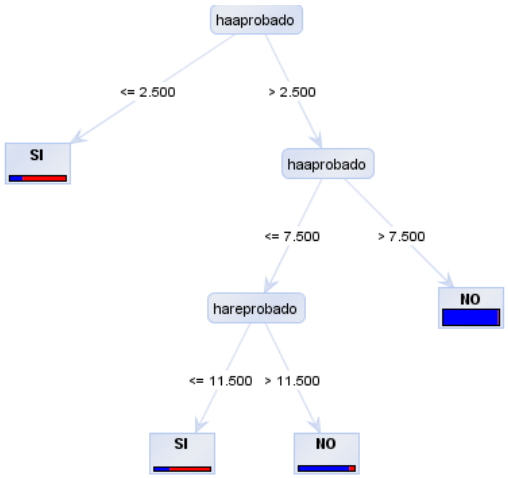
- De los 297 casos de deserción, 290 han desertado habiendo cursado hasta 4 semestres, 153 teniendo 9 o menos créditos, 76 teniendo entre 10 y 35 créditos, 90 aprobaron 7 o menos materias. 190 desertaron habiendo aprobado 2 o menos materias.



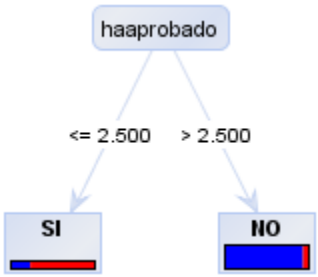
Árbol 20a.



Árbol 20b.



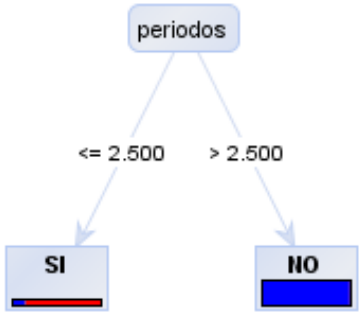
Árbol 20c.



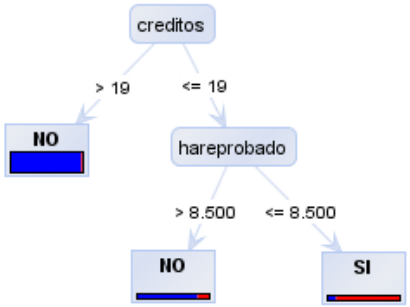
Árbol 20d.

Generación 2008:

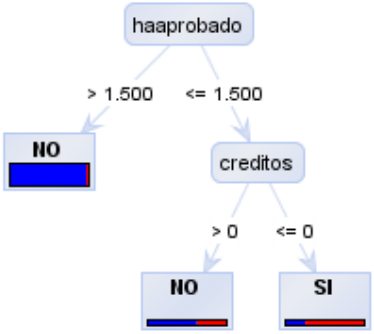
- De los 169 casos de deserción, 161 desertó con uno o dos semestres cursados.
- Se tienen 102 casos de alumnos que tienen 0 créditos de los cuales 75 ya desertaron. 124 casos han aprobado sólo una materia habiendo reprobado varias materias ya, de los cuales 45 ya desertaron.
- 114 desertaron habiendo reprobado 8 materias o menos.



Árbol 21.



Árbol 22.



Árbol 23.

Y las consultas para corroborar son:

```
SELECT * FROM alumnodesercion4
WHERE generacion=2008
AND haaprobado<=1.5
AND creditos>0
ORDER BY deserto
```

```
SELECT * FROM alumnodesercion4
WHERE generacion=2008
AND haaprobado<=1.5
AND creditos=0
ORDER BY deserto
```

Para la generación 2009 no se encontraron reglas relevantes y además son pocas. Esto se debe a que apenas se empiezan a acumular datos para esta generación.

Concluyendo a grandes razgos de los patrones encontrados, se observa que los alumnos que cambiaron de plan de estudios les fue mejor que a aquellos que se quedaron en el viejo plan. De los que desertaron varios reprobaron muchas materias (generalmente más de 6) y aprobaron pocas materias tales como las materias del anexo: cultura y comunicación, física experimental, álgebra, geometría analítica, cálculo I, computadoras y programación, química, computación para ingenieros, análisis gráfico y comunicación oral y escrita.

En otros casos, varios desertaron faltándoles 1 o pocas materias. Para algunos casos se observa porque alcanzaron el límite (o ya casi) permitido de tiempo para acabar la carrera.

Al observar los árboles generados, con frecuencia se necesita hacer consultas a la base de datos traduciendo las reglas encontradas a SQL. Observando los registros de estas consultas, complementan la explicación.

Después de observar las gráficas de las figuras de la 4.47 a la 4.51, se observa cómo van cambiando las tendencias. Esto sugiere utilizar algún algoritmo de clasificación con el

fin de poder predecir las próximas tendencias, es decir, hacer predicciones de cuántos alumnos van a desertar y cuántos terminarán sus materias por generación. Se empieza a hacer pruebas con redes neuronales, *perceptrón* y el clasificador *IBk*. De los 2 que usamos, este último fue el que dio los mejores resultados además de ser el que menos tiempo de ejecución necesita.

Para llevar a cabo las predicciones, en *Rapidminer* hacemos un nuevo árbol de procesos.

Hacer clic con el botón derecho del *mouse* sobre el operador *root* y del menú desplegado hay que ir deslizando el mismo por *new operador*, *IO*, *examples* y finalmente dar clic en *ExampleSource* (*figura 4.85*).

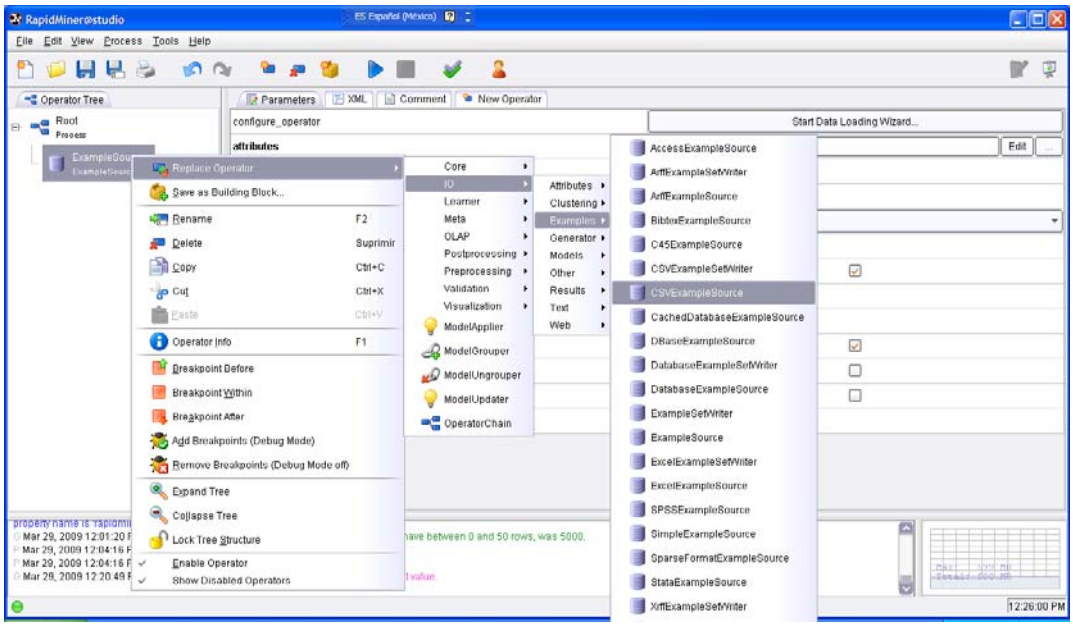


Figura 4.85. Seleccionando el tipo de archivo *separado por coma* (*.csv*) como archivo de datos (operador 2).

En seguida se mostrará este operador, el cual carga un archivo en formato separado por coma y dar un clic sobre éste operador para poder visualizar los parámetros del lado derecho.

Como primer parámetro, se tiene el *filename* el cual pide seleccionar el archivo que se va a utilizar en el análisis. Hay que dar clic en el botón que tiene puntos suspensivos (…) y buscar el archivo y seleccionarlo. *Figura 4.86*.

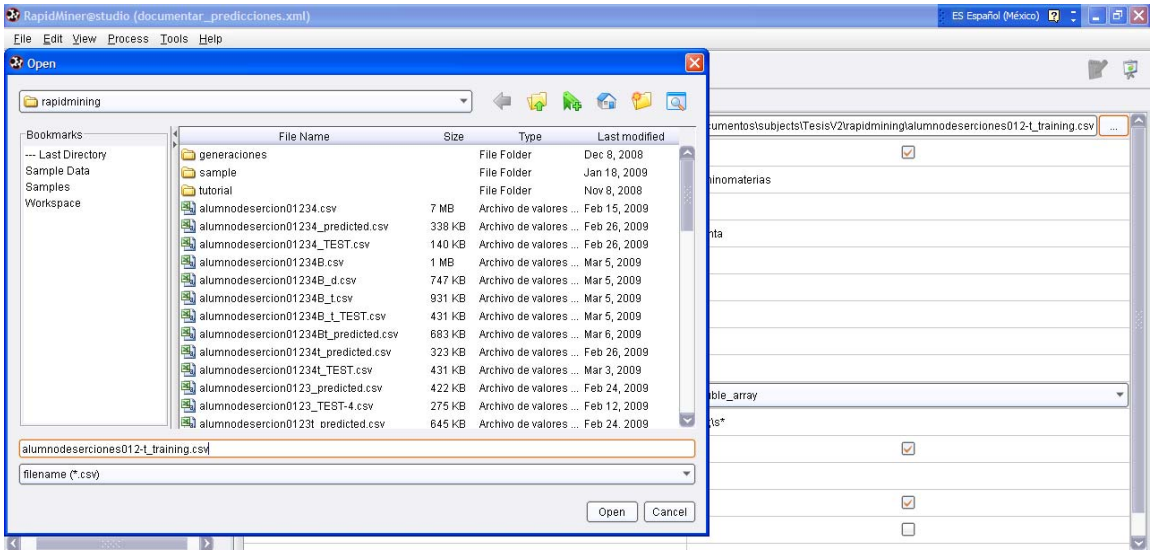


Figura 4.86. Seleccionando el archivo.

Hay que llenar los demás parámetros que se tienen abajo. En el parámetro *label_name* teclear *terminomaterias* el cual es la columna o variable sobre la que se desea predecir. *Label_column* se refiere a qué variable corresponde en el archivo de datos; en este caso le corresponde la columna 7. El parámetro *id_name* e *id_column* piden el nombre de la columna que funge como identificador y el número de la columna respectivamente. En este caso se teclea cuenta y como número de columna es 1. Véase la *figura 4.87*.

Para visualizar estos datos, se puede dar clic en *play* para ejecutar el proceso. A estas alturas el proceso es muy breve por lo que lo único que va a hacer *Rapidminer* en este caso será la de leer el archivo y mostrarlo. Nota: Si el proceso no se ha guardado, cada vez que se corra un proceso en *Rapidminer*, éste preguntará si se desea guardar el árbol de procesos o no. Esto es opcional. Si el árbol de procesos ya es demasiado largo o grande, entonces sí se recomienda ampliamente guardar el proceso, si no, al cerrar *rapidminer* y al volverlo abrir, habrá que construir de nuevo todo el proceso.

Para cambiar de vista entre los resultados y el árbol de procesos basta con dar clic sobre los botones que están en la parte superior derecha. Véase la *figura 4.88*.

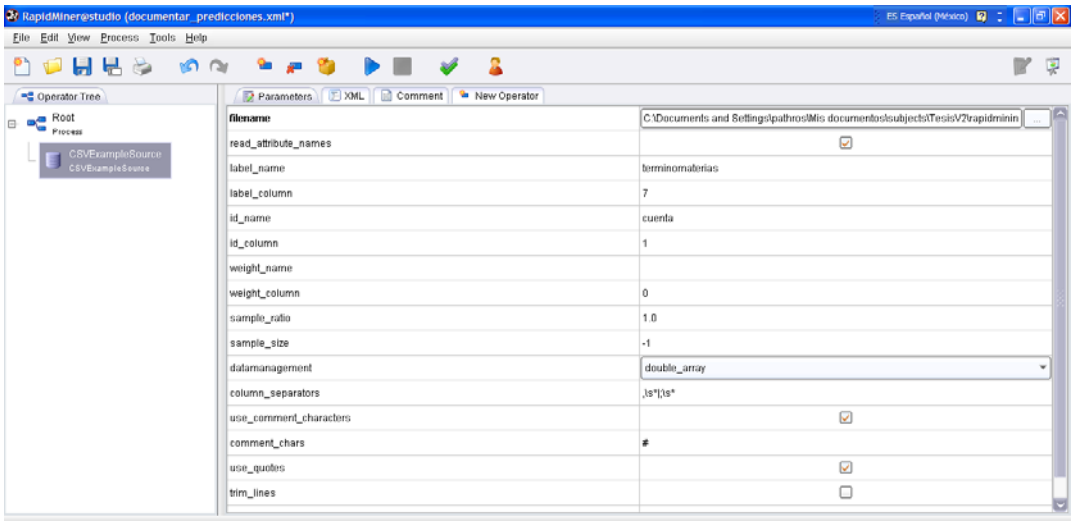
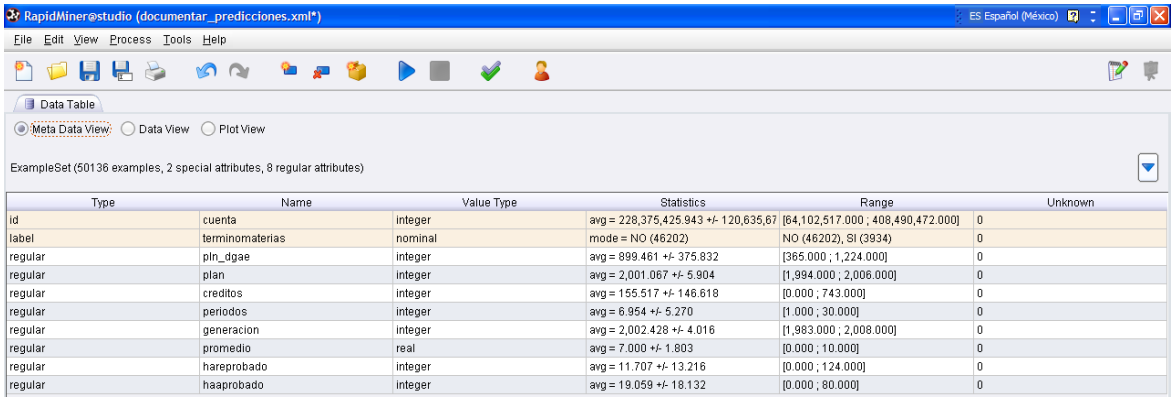


Figura 4.87. Corriendo el proceso para visualizar los datos del archivo fuente seleccionado.



Type	Name	Value Type	Statistics	Range	Unknown
id	cuenta	integer	avg = 228,375,425.943 +/- 120,635,67	[64,102,517.000 ; 408,490,472.000]	0
label	terminomaterias	nominal	mode = NO (46202)	NO (46202), SI (3934)	0
regular	pln_dgae	integer	avg = 899.461 +/- 375.832	[365.000 ; 1,224.000]	0
regular	plan	integer	avg = 2,001.067 +/- 5.904	[1,994.000 ; 2,006.000]	0
regular	creditos	integer	avg = 155.517 +/- 146.618	[0.000 ; 743.000]	0
regular	periodos	integer	avg = 6.954 +/- 5.270	[1.000 ; 30.000]	0
regular	generacion	integer	avg = 2,002.428 +/- 4.016	[1,983.000 ; 2,008.000]	0
regular	promedio	real	avg = 7.000 +/- 1.803	[0.000 ; 10.000]	0
regular	hareprobado	integer	avg = 11.707 +/- 13.216	[0.000 ; 124.000]	0
regular	haaprobado	integer	avg = 19.059 +/- 18.132	[0.000 ; 80.000]	0

Figura 4.88. Visualizando los resultados obtenidos por el proceso.

En esta vista de la *figura 4.88* se pueden ir viendo los metadatos, los datos y existe asimismo la posibilidad de graficar los datos. Para que se pueda graficar todos los datos sin problemas, habrá que agregar un nuevo operador llamado *ExampleVisualizer* el cual ayuda a visualizar los datos importados.

Para visualizar los datos que se han cargado, de las opciones mostradas, seleccionar *Data View* (*figura 4.89*).

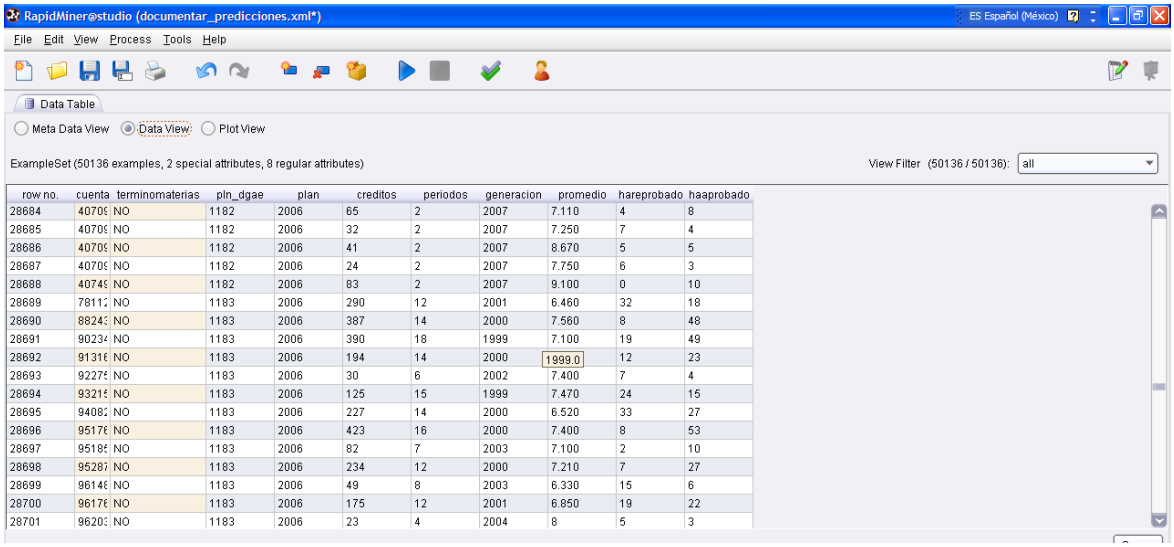


Figura 4.89. Visualizando los datos del archivo fuente.

Para agregar el operador *ExampleVisualizer* hay que dar clic con el botón derecho del *mouse* sobre el operador *root* para visualizar el menú contextual, de ahí, seleccionar *New Operator*, *Visualization*, *ExampleVisualizer*; de tal forma que se cuenta con este operador en el árbol de procesos. *Figura 4.90*.

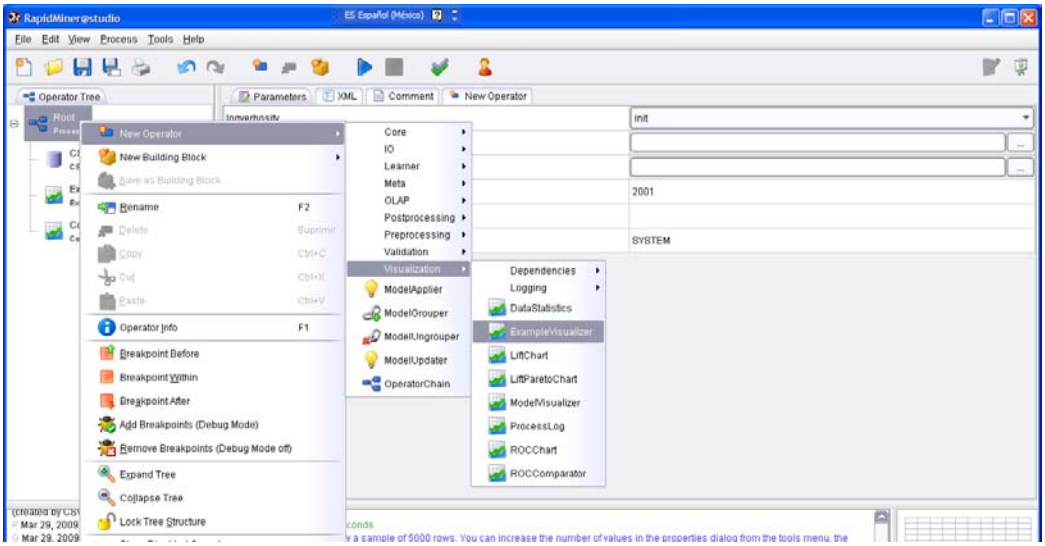


Figura 4.90. Seleccionando el operador *ExampleVisualizer* (operador 3).

Ahora sobre este operador, en el árbol de procesos, dar doble clic con el botón derecho del *mouse* para colocar ahí una pausa de tal forma que se puedan ver los datos antes de que el proceso siga con el siguiente operador. La pausa se muestra como un cuadrito rojo con una flecha hacia abajo. Véase la *figura 4.91*.

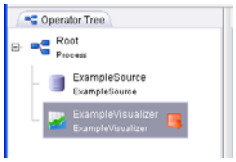


Figura 4.91. Colocando una pausa al proceso.

Asimismo se puede agregar otro operador que genere una matriz de correlación con el fin de poder visualizar las correlaciones entre las variables con las que estamos trabajando. Para ello, sobre el operador *root* dar clic con el botón derecho del *mouse* y del menú contextual seleccionar *New Operator*, *Visualization*, *Dependencies*, *CorrelationMatrix* (*figura 4.92*). Dejamos como están los parámetros de este operador por default. Es decir, sólo está seleccionada la casilla *normalize_weights*.

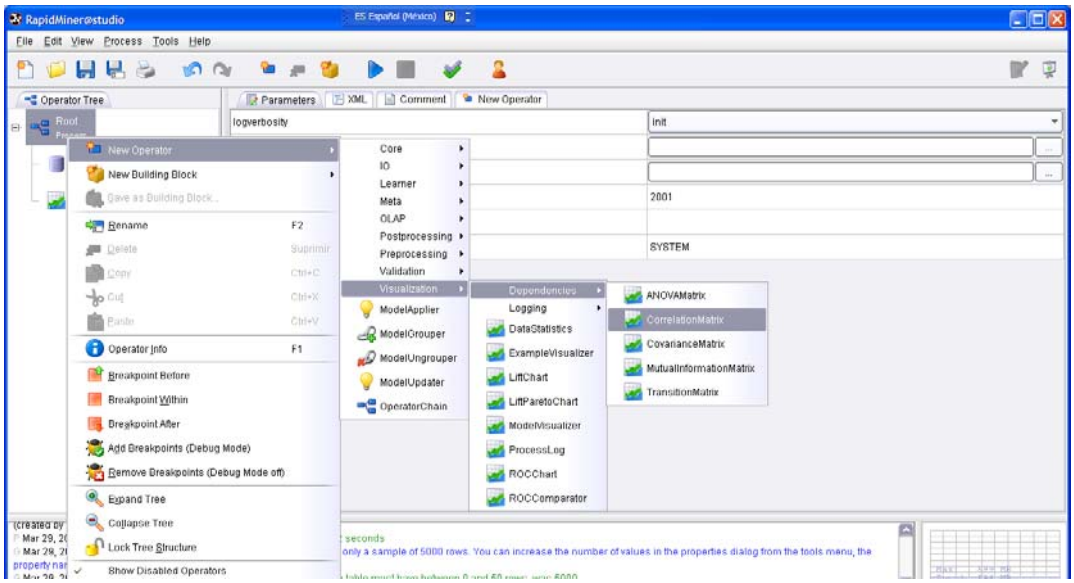


Figura 4.92. Seleccionando el operador que genera una matriz de correlación de todas las variables.

Se vuelve a teclear con el botón derecho del *mouse* sobre el operador *root* para seleccionar un tipo de validación con el fin de obtener la confiabilidad del modelo (es decir, qué tan acertado es y qué tan bien aprende de los datos). En este caso se usará la validación cruzada o *XValidation*. Del menú seleccionar *New Operator*, *Validation* y *XValidation*. Véase la *figura 4.93*.

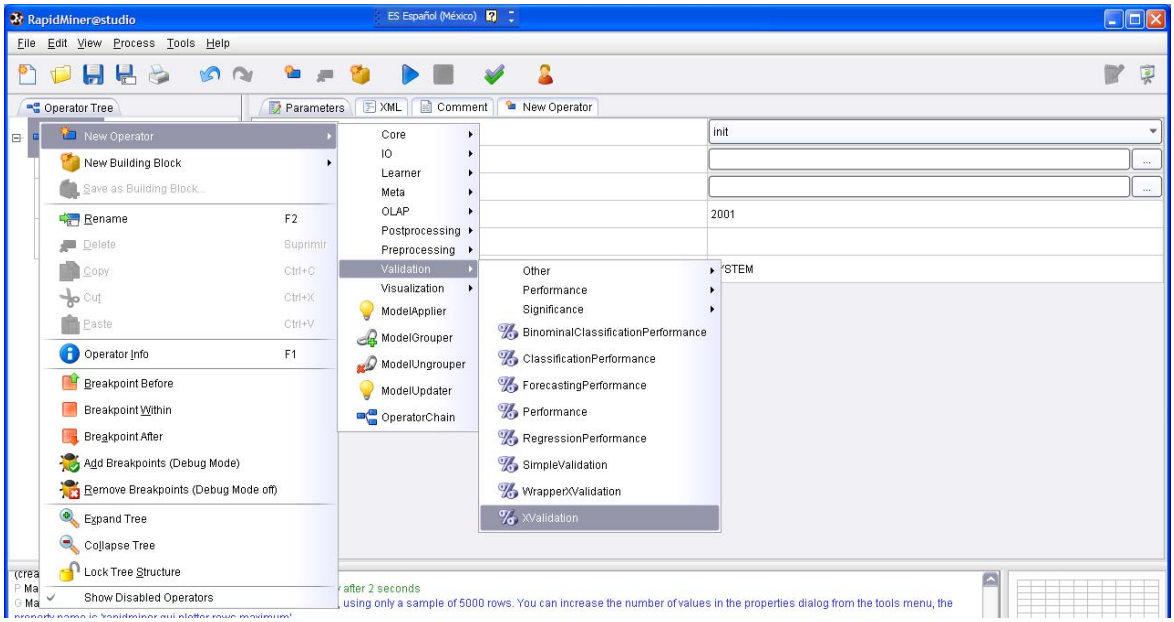


Figura 4.93. Seleccionando la validación cruzada o *XValidation* (operador 4).

Dar clic para señalar el operador *XValidation* para poder visualizar los parámetros del lado derecho. Se visualizarán unas opciones, hay que marcar las primeras tres que son *keep_example_set*, *create_complete_model* y *average_performances_only*. Los cuales sirven para visualizar los datos, crear del modelo y mostrar los promedios de desempeño. Véase la *figura 4.94*. Se observan otros parámetros como el *leave_one_out* el cual es un tipo de validación cruzada de *n* pliegues en donde *n* es el número de instancias en el conjunto de datos. La gran desventaja de usar esta validación, aparte del gran desgaste computacional es que no se puede estratificar lo cual provoca que se den tasas de error altas; por lo que esta validación no se usará. Se tiene otro parámetro que pide el número de validaciones o *number of validations* el cual es el número de veces que se repite el experimento para ir evaluando su desempeño.

El valor por default es 10 veces, lo mínimo es 2 veces. Abajo está otro parámetro en donde se puede seleccionar el tipo de muestreo y finalmente se tiene *local_random_seed* el cual es un generador de números aleatorios, se deja así como está. Por default el valor es -1.

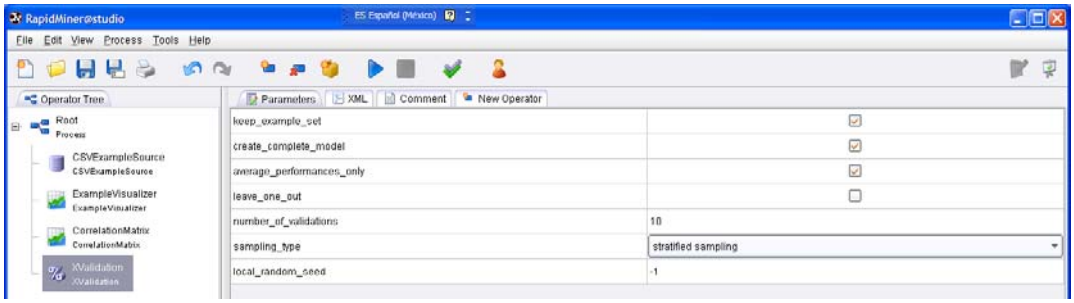


Figura 4.94. Parámetros del operador de validación cruzada *XValidation*.

Sobre el operador *XValidation* dar clic con el botón derecho del *mouse* para agregar un operador cadena, el cual sirve para unir varios operadores a un operador. Se empieza con el operador cadena u *OperatorChain* del menú contextual: *New Operator*, *OperatorChain* (*figura 4.95*).

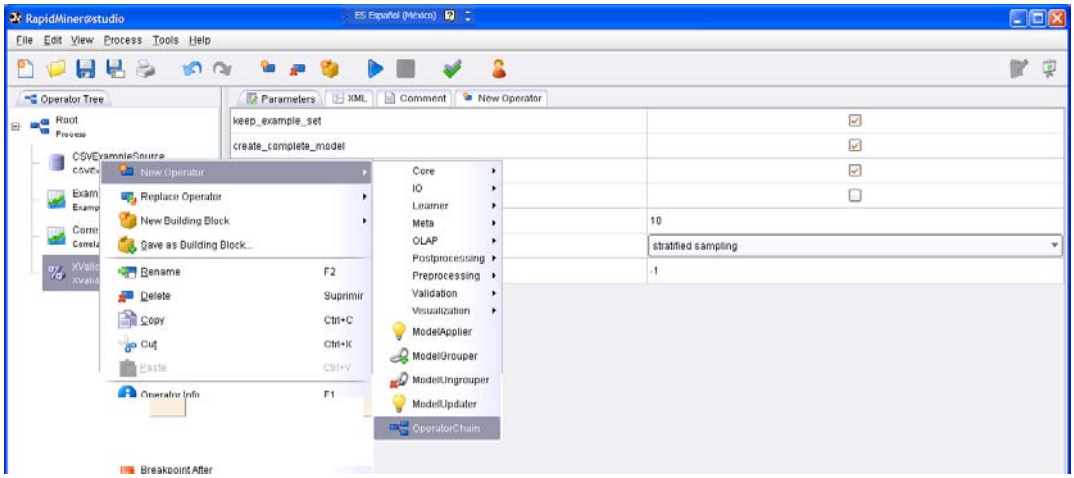


Figura 4.95. Seleccionando un operador *cadena* para encadenar o unir más procesos con otros dentro del árbol.

Sobre el operador *cadena* dar clic con el botón derecho del *mouse* para seleccionar el algoritmo que clasificará los registros que lea, en este caso será el algoritmo de clasificación *IBk*. En el menú seleccionar *New Operator*, *Learner*, *Supervised*, *Weka*, *Lazy* y *W-IBk*. Véase la *figura 4.96*.

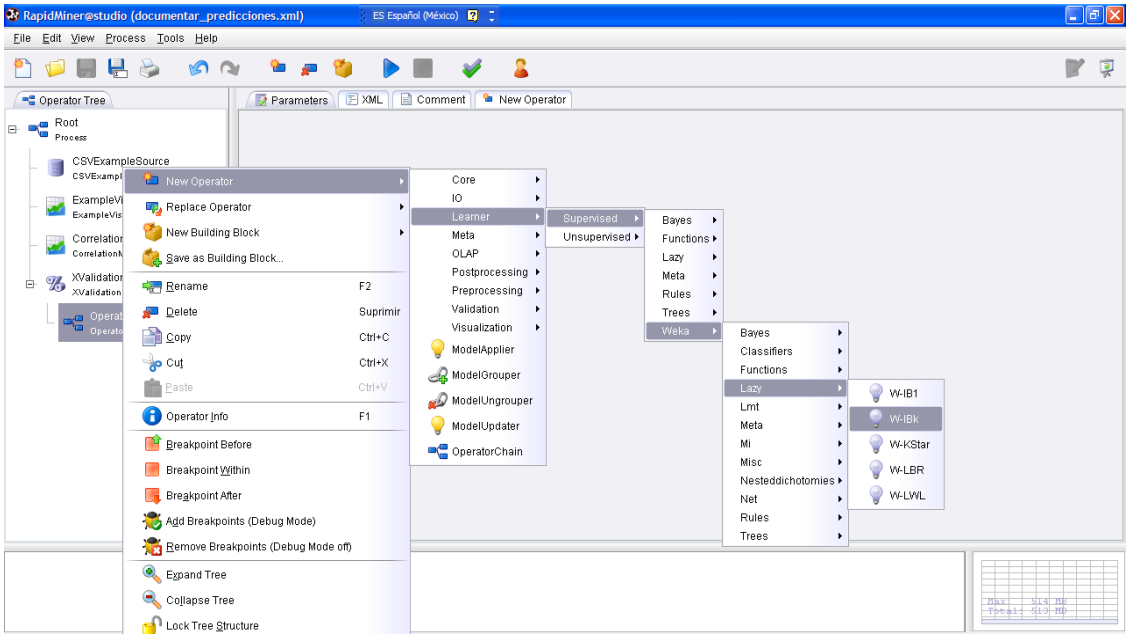


Figura 4.96. Seleccionando el algoritmo de clasificación del vecino *k* más cercano *IBk*.

Dar clic sobre este nuevo operador (*IBk*) para poder visualizar los parámetros del lado derecho. Sólo seleccionar la casilla *keep_example_set* y lo demás, si así se desea, se deja así ya que son los valores por default. Véase la *figura 4.97*.

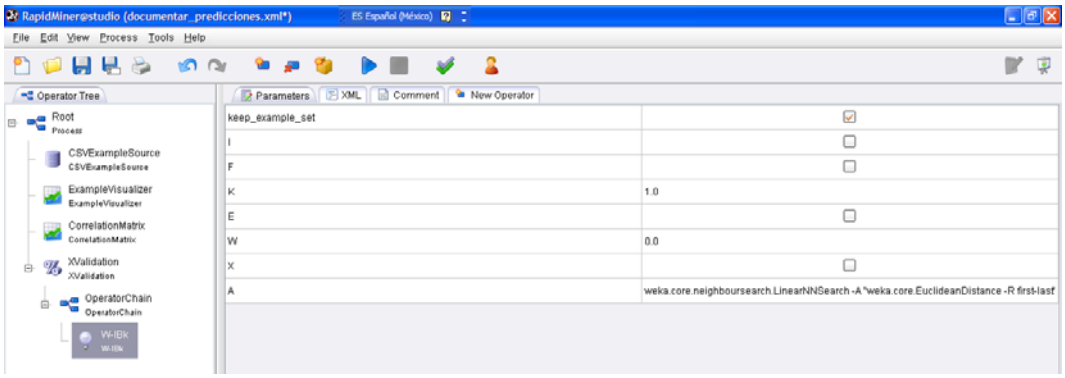


Figura 4.97. Parámetros del operador *IBk* (operador 5).

Se puede también agregar un operador que guarde el modelo generado. Esto es útil cuando se quiere volver a aplicar el modelo a otros datos (por ejemplo, de prueba). Dar clic con el botón derecho del *mouse* sobre el operador cadena u *OperatorChain* y en el menú contextual seleccionar *New Operator*, *IO*, *Models*, *ModelWriter*. Véase la *figura 4.98*.

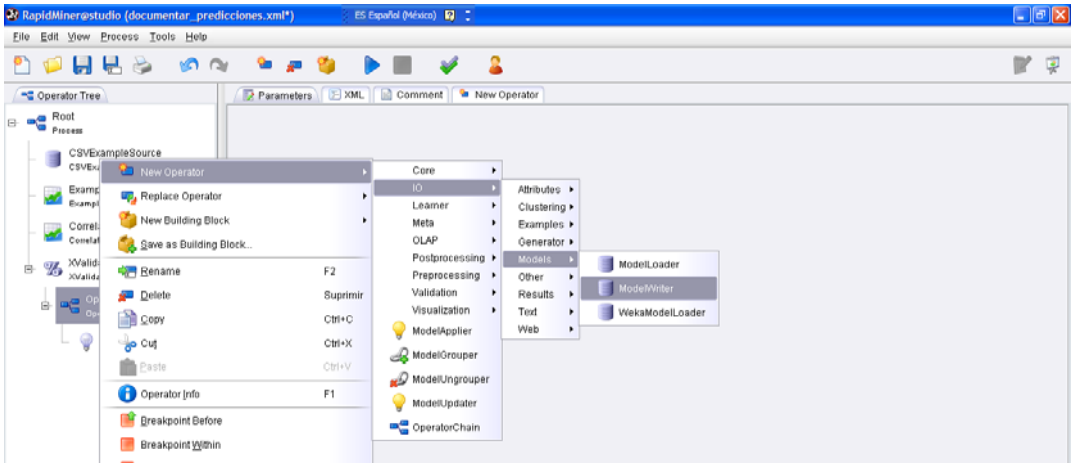


Figura 4.98. Seleccionando el operador que guarda el modelo generado: *ModelWriter*.

Dando clic en este operador, *ModelWriter*, se pedirá un nombre de archivo para guardar el modelo. Lo mismo que se hizo en el operador que carga los datos en formato *.csv*, se da clic sobre el botón de los puntos suspensivos para teclear el nombre del archivo que guardará el modelo (cuya extensión es *.mod*), y una vez seleccionado el nombre y la ruta en dónde se va a guardar, dar clic en *abrir*. No hay problema si el archivo no existe, ya que éste se creará automáticamente si no existe. Si el archivo a guardar ya existe, se sobrescribirá. Véase la *figura 4.99*.

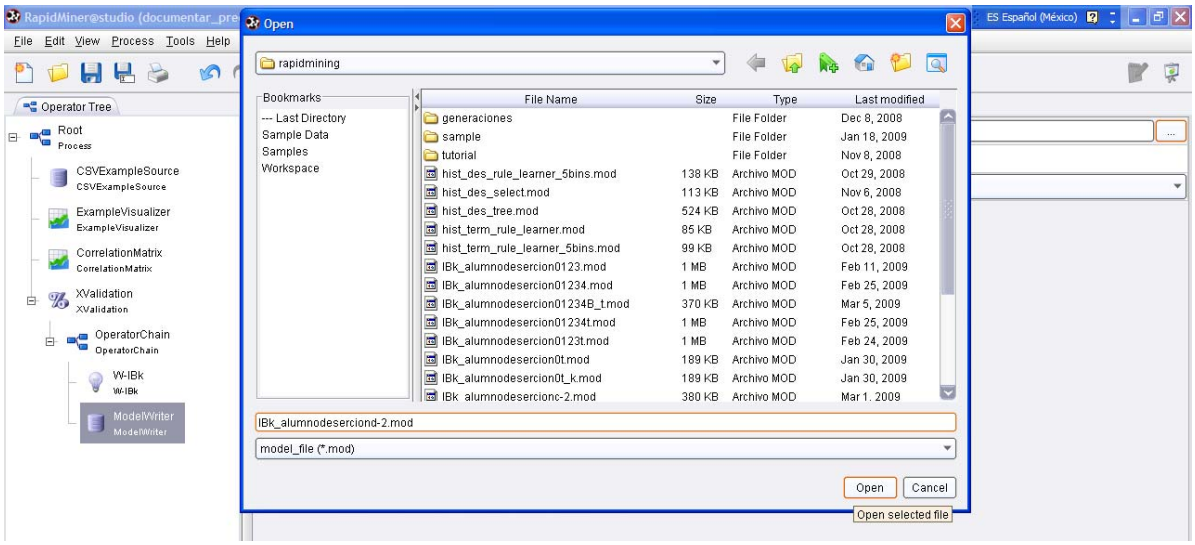


Figura 4.99. Seleccionando nombre y la ruta en dónde guardar el archivo del modelo o los modelos generados (operador 6).

Sobre el operador *XValidation* dar clic con el botón derecho del *mouse* para seleccionar otro operador *cadena* u *OperatorChain* con el fin de unir los procesos. Esto se hace porque en *Rapidminer* los operadores están limitados a tener pocos nodos hijos,

entonces con este operador cadena se pueden unir más operadores. Véase la *figura 4.100*

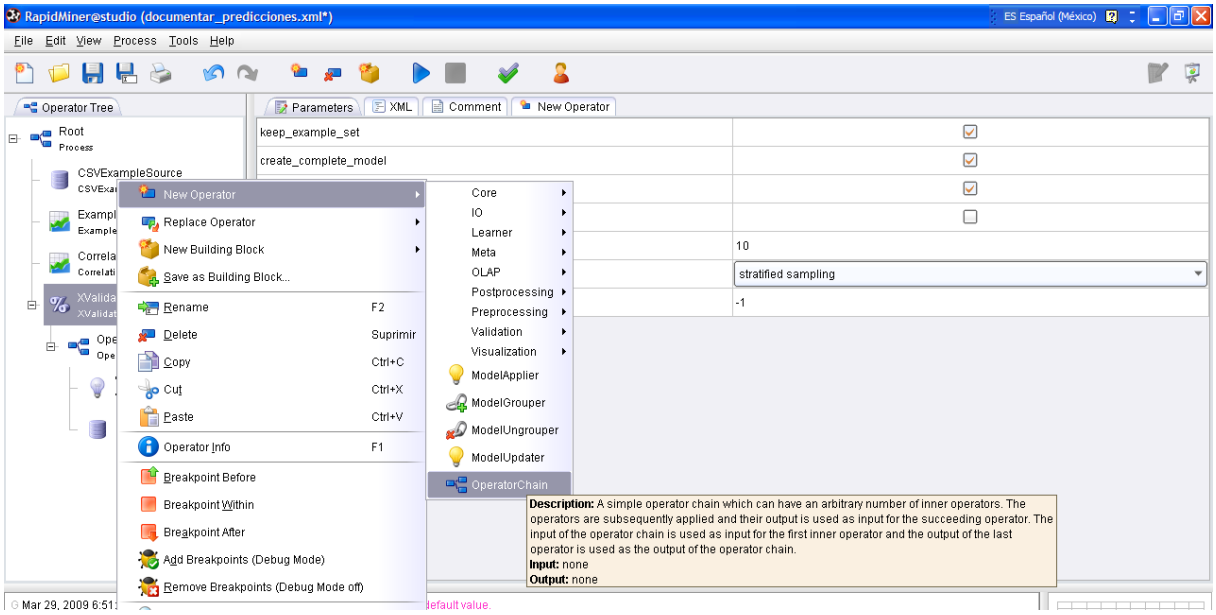


Figura 4.100. Seleccionando otro operador cadena u *OperatorChain*.

Ahora dar clic con el botón derecho del *mouse* sobre el nuevo operador *OperatorChain* para seleccionar un operador que aplique el modelo o *ModelApplier*. En el menú seleccionar *New Operator*, *ModelApplier*. Véase la *figura 4.101*.

Dar clic sobre *ModelApplier* para poder visualizar los parámetros del lado derecho. En este caso, seleccionar las casillas *keep_model* y *create_view* para que siga mostrando el modelo. Véase la *figura 4*.102.

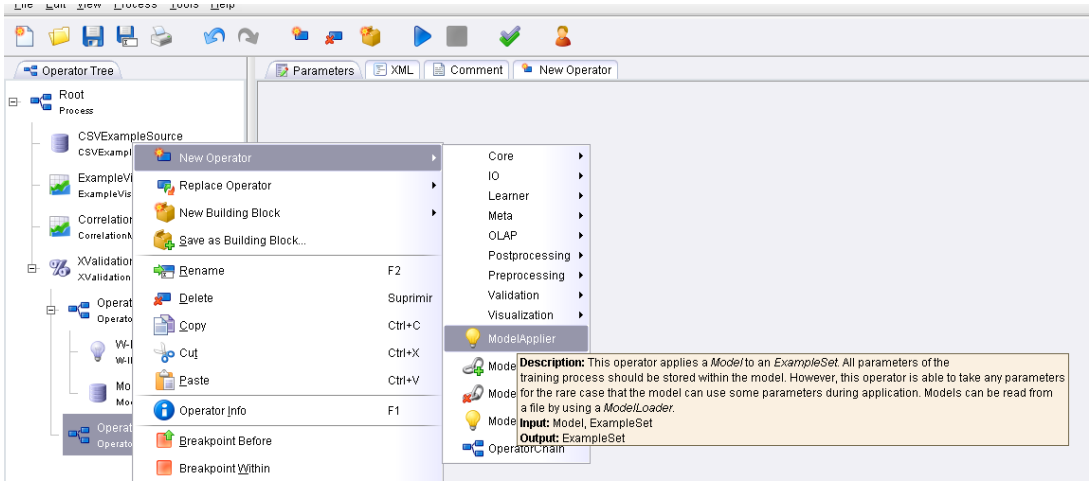


Figura 4.101. Seleccionando el aplicador del modelo o *ModelApplier* (operador 7).

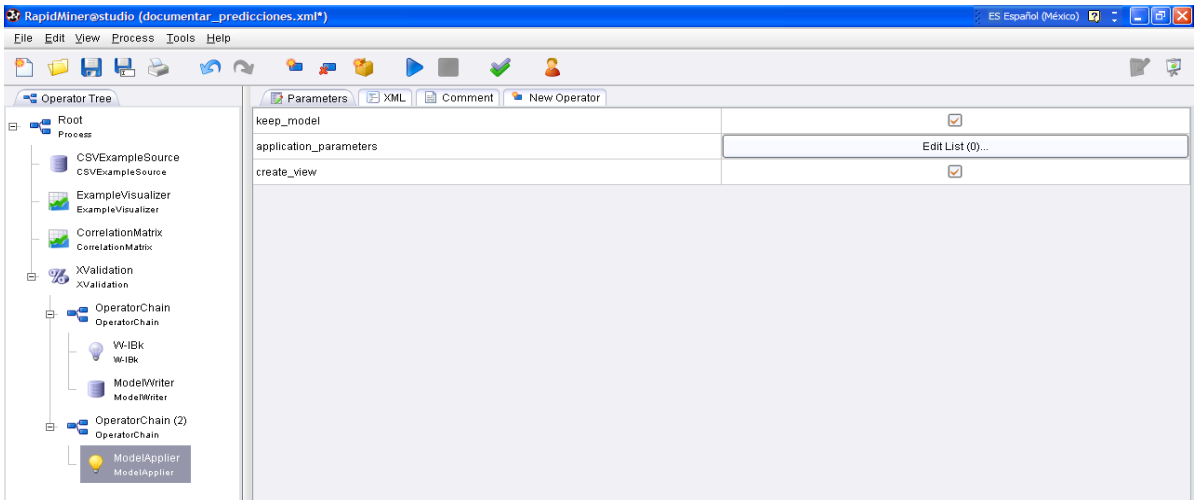


Figura 4.102. Parámetros del operador *ModelApplier* o aplicador del modelo.

Ahora se va a agregar un operador que evalúe el desempeño del modelo o *performance*. Se da clic sobre el operador cadena con el botón derecho del *mouse* y se selecciona *New Operator*, *Validation*, *ClassificationPerformance*. Véase la *figura 4.103.*

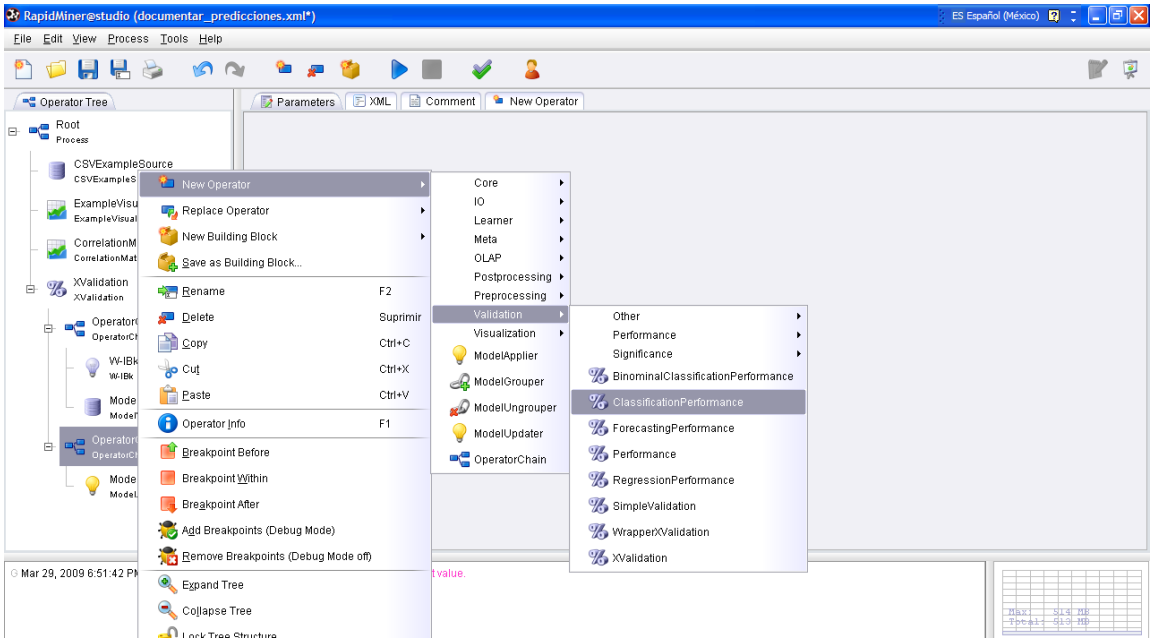


Figura 4.103. Seleccionando el operador Classification*Performance* que evalua el desempeño del modelo o de los modelos generados (operador 8).

Después dar clic sobre el operador *performance* para poder visualizar los parámetros del lado derecho y marcar las casillas *keep_example_set* para poder seguir viendo los datos con los que se está trabajando; y otros parámetros de evaluación que se consideren

pertinentes tales como *accuracy* (exactitud), *classification error*, etc. En el parámetro *main_criterion* se elige *accuracy*. Véase la *figura 4.104*.

Una vez listo el proceso, se procede a *validar* el proceso haciendo clic sobre el botón de validación (en forma de palomita verde). Esto sirve para verificar que el árbol de proceso no tenga errores (*figura 4.105*).

Después se da *clic* en el botón de *play* para iniciar el proceso y a continuación pregunta si se desea guardar el proceso, en este caso se puede decir que sí o no. Después de esto, el proceso comienza a ejecutarse. La marcha del proceso se va mostrando en la parte inferior; es decir, se va mostrando en qué parte del árbol de procesos se va ejecutando el proceso. Véase la *figura 4.106*.

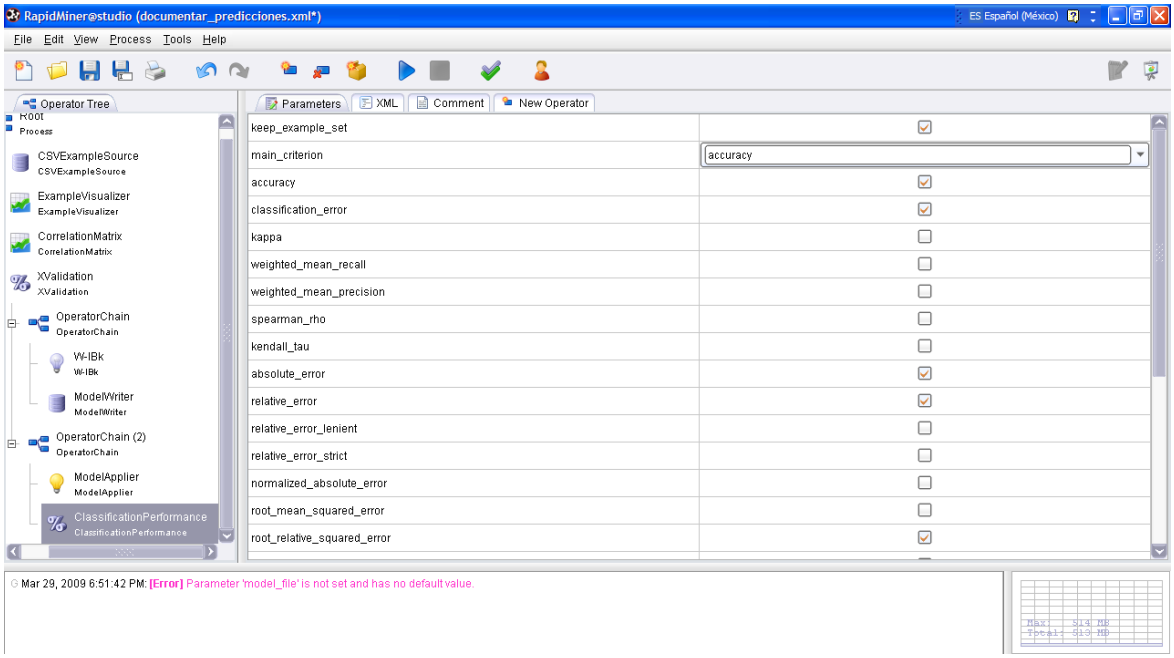


Figura 4.104. Parámetros del operador que evalúa el desempeño o *performance*.

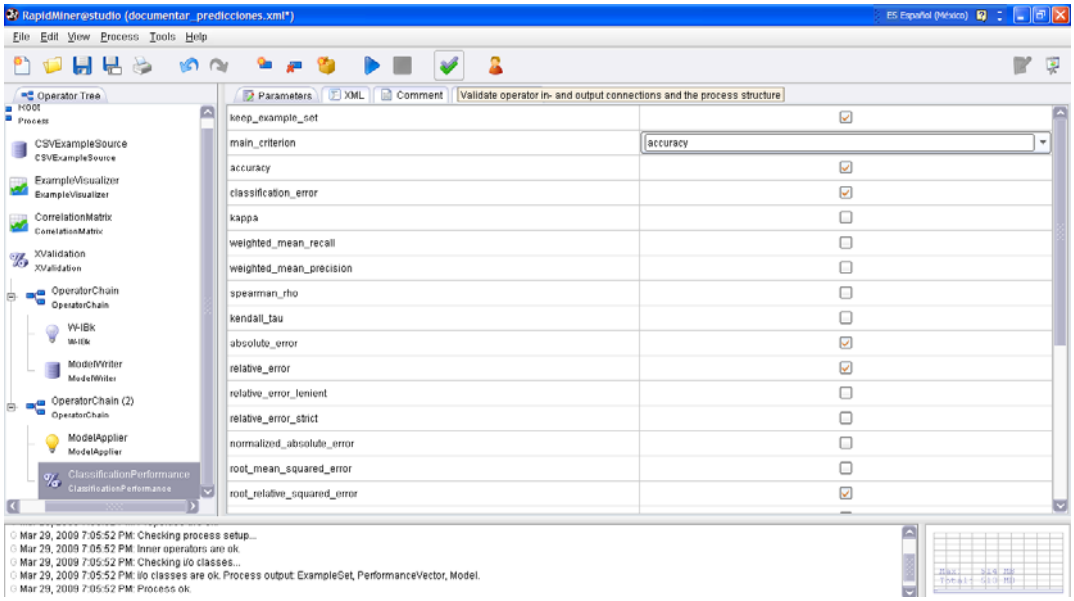


Figura 4.105. Verificando que el árbol de procesos esté bien armado o que no tenga errores haciendo *clic* sobre la palomita verde.

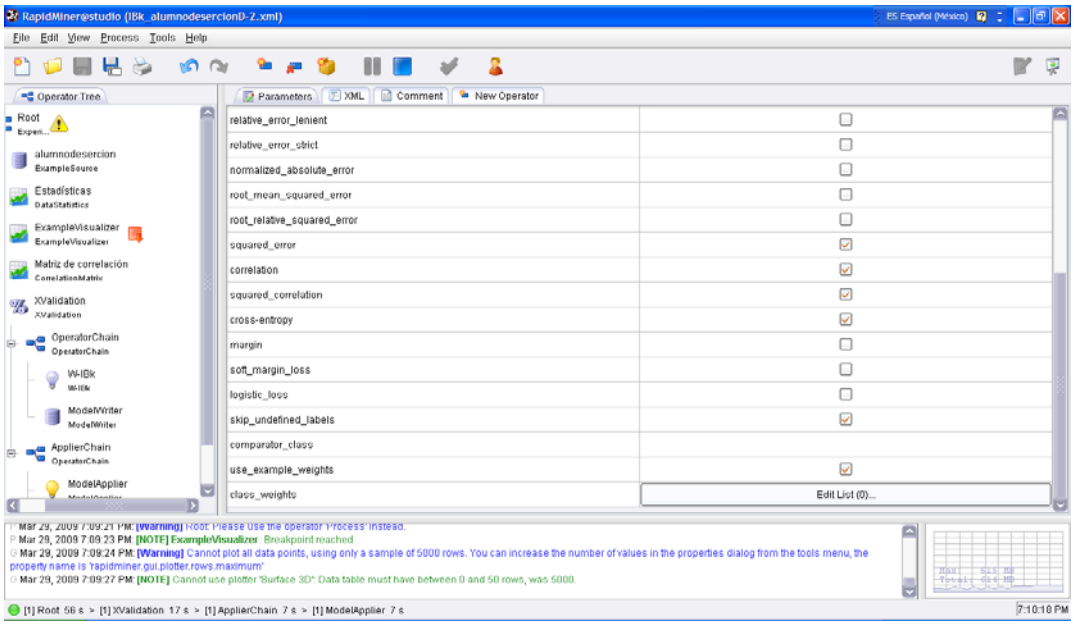


Figura 4.106. Corriendo el proceso.

El proceso hace una pausa, como se había mencionado, sirve para poder ver los datos cargados antes de que siga el análisis. Esto ayuda para ver si los datos han sido cargados correctamente y hacer unas gráficas. Véanse las *figuras 4.107* y 4.*108*.

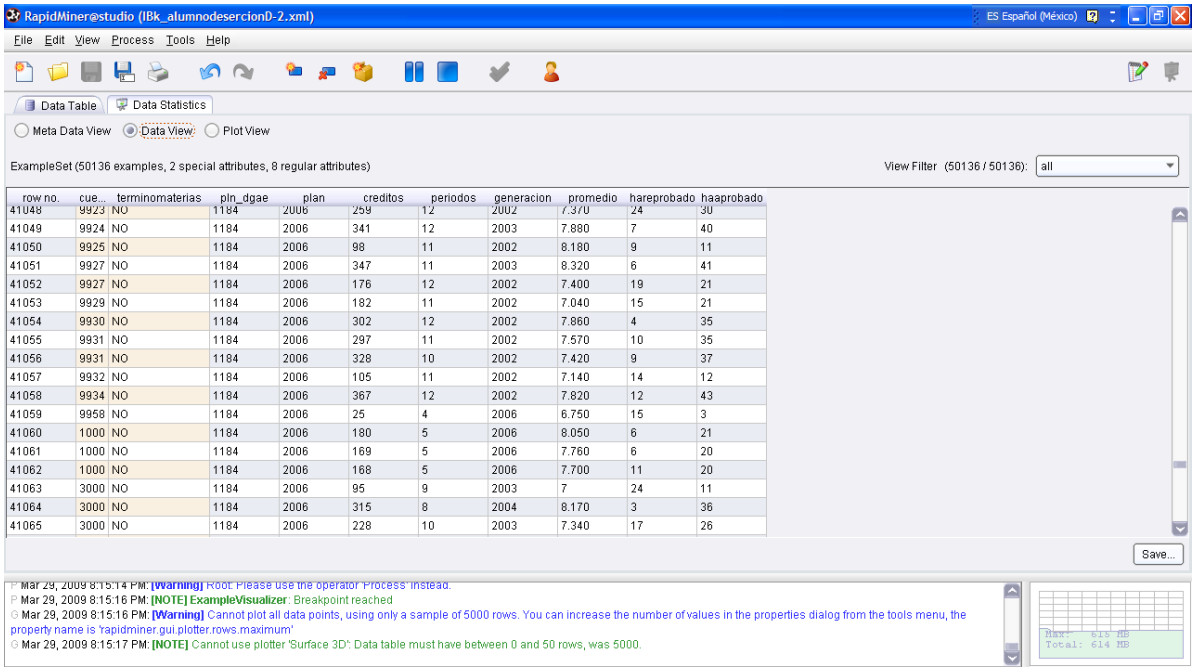


Figura 4.107. Visualizando los datos cargados.

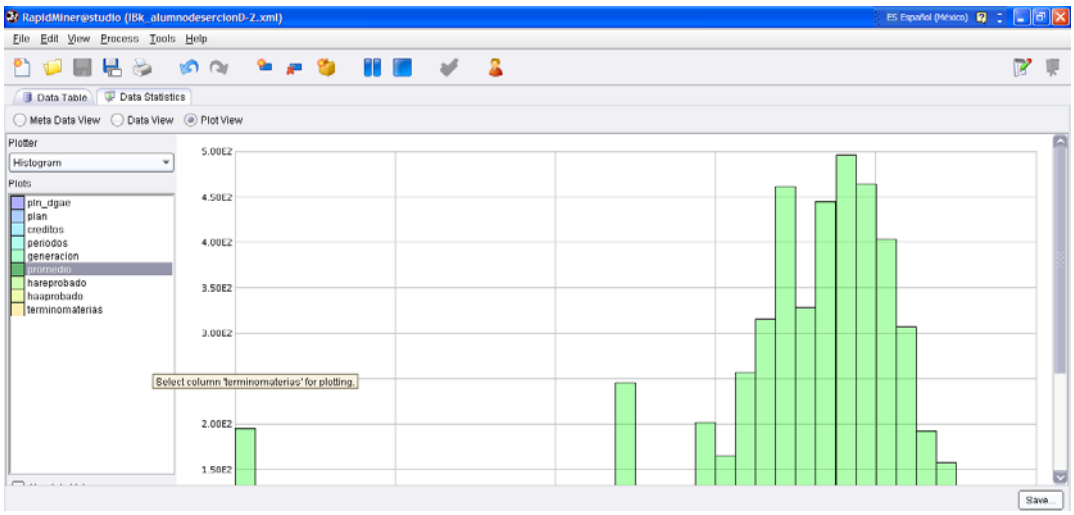


Figura 4.108. Realizando gráficas de los datos

Para continuar, hay que dar clic sobre el botón de pausa o sobre el botón de detener si no se quiere continuar con el proceso. De hecho, las pausas se pueden colocar en cualquier operador del árbol de procesos. Para quitar las pausas hacer nuevamente doble clic sobre las mismas.

Cuando el proceso termina, *Rapidminer* automáticamente muestra la ventana de resultados. Véase la *figura 4.109*.

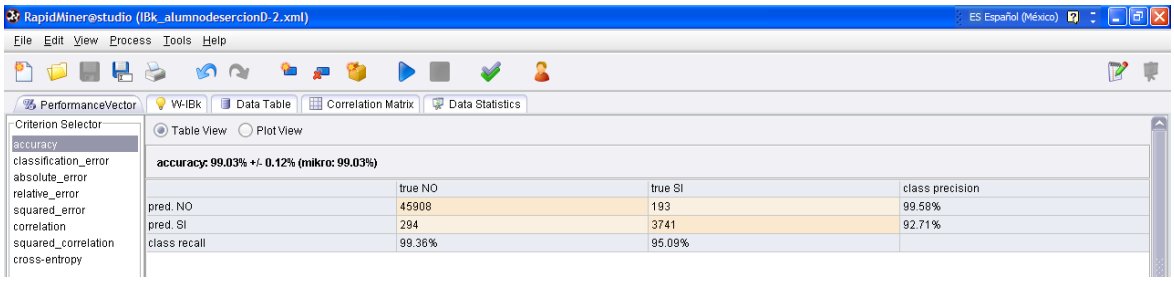


Figura 4.109. Mostrando el desempeño del modelo con la variable *terminomaterias*.

Los resultados de precisión (*precision*), exactitud (*accuracy*) y la correlación son los siguientes:

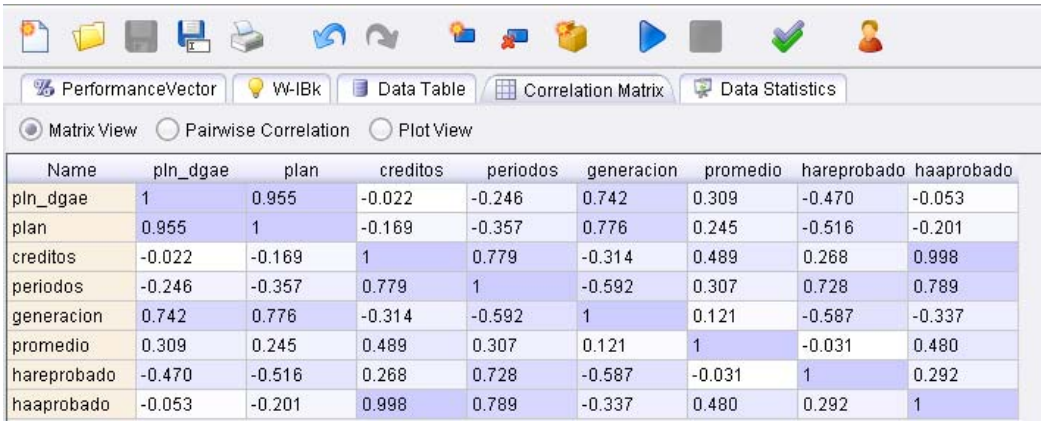
```
accuracy: 99.03% +/- 0.12% (mikro: 99.03%)
classification_error: 0.97% +/- 0.12% (mikro: 0.97%)
correlation: 0.934 +/- 0.008 (mikro: 0.934)
```

Y la tabla de la matriz de confusión es la siguiente:

	true NO	**true SI**	**class precision**
pred. NO	45908	193	99.58%
pred. SI	294	3741	92.71%
class recall	99.36%	95.09%	

En la que de las predicciones para los que sí terminan sus materias, 3741 son correctas contra 294 que no lo fueron dejando un 92.71% de precisión. Para las predicciones de los que no terminan sus materias se arroja que 45908 fueron correctas contra 193 que no lo fueron dejando un 99.58% de precisión. El resultado *recall* se refiere a la cobertura que tiene el modelo sobre los datos aprendidos para cada valor de la variable, es decir, "sí" o "no".

La matriz de correlación es la siguiente (*figura 4.110*):



Name	pln_dgae	plan	creditos	periodos	generacion	promedio	hareprobado	haaprobado
pln_dgae	1	0.955	-0.022	-0.246	0.742	0.309	-0.470	-0.053
plan	0.955	1	-0.169	-0.357	0.776	0.245	-0.516	-0.201
creditos	-0.022	-0.169	1	0.779	-0.314	0.489	0.268	0.998
periodos	-0.246	-0.357	0.779	1	-0.592	0.307	0.728	0.789
generacion	0.742	0.776	-0.314	-0.592	1	0.121	-0.587	-0.337
promedio	0.309	0.245	0.489	0.307	0.121	1	-0.031	0.480
hareprobado	-0.470	-0.516	0.268	0.728	-0.587	-0.031	1	0.292
haaprobado	-0.053	-0.201	0.998	0.789	-0.337	0.480	0.292	1

Figura 4.110. La matriz de correlación.

De la matriz de correlación se observa que hay una fuerte correlación entre las variables *haaprobado* y *periodos* (0.789), evidentemente también entre *haaprobado* y *creditos* (0.998). También entre *hareprobado* y *periodos* (0.728). Hay asimismo una cierta correlación entre *promedio* y *haaprobado*.

Esto fue la generación del modelo para la variable *terminomaterias*. Ahora se hará exactamente lo mismo pero para la variable *deserto*.

Los resultados que se obtuvieron para la variable *deserto* son los siguientes (*figura 4.111*):

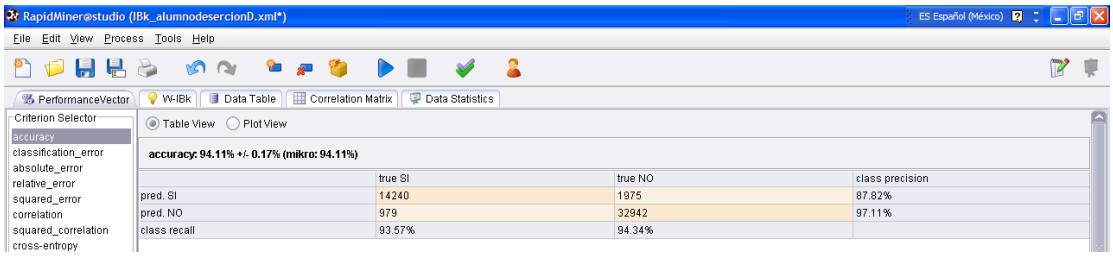


Figura 4.111. Mostrando el desempeño del modelo con la variable *deserto*.

Los resultados de presición (*precision*), exactitud (*accuracy*) y la correlación son los siguientes:

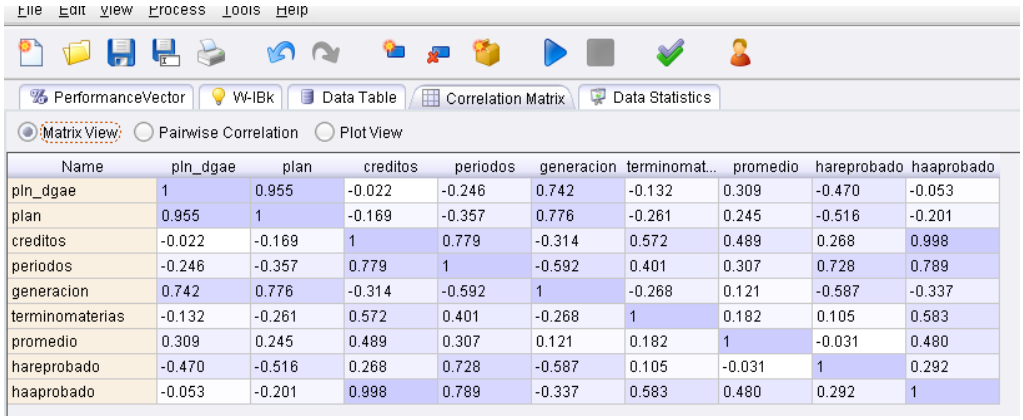
```
accuracy: 94.11% +/- 0.17% (mikro: 94.11%)
classification_error: 5.89% +/- 0.17% (mikro: 5.89%)
correlation: 0.864 +/- 0.005 (mikro: 0.864)
```

Y la tabla de la matriz de confusión es la siguiente:

	true SI	**true NO**	**class precision**
pred. SI	14240	1975	87.82%
pred. NO	979	32942	97.11%
class recall	93.57%	94.34%	

En la que de las predicciones para los que sí desertan, 12240 son correctas contra 1975 que no lo fueron dejando un 87.82% de precisión. Para las predicciones de los que no desertan se arroja que 32942 fueron correctas contra 979 que no lo fueron dejando un 97.11% de precisión.

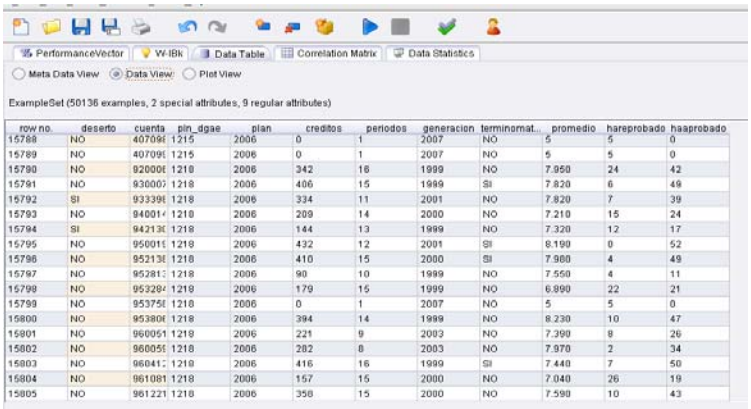
La matriz de correlación para este caso es la siguiente (*figura 4.112*):



Name	pln_dgae	plan	creditos	periodos	generacion	terminomat...	promedio	hareprobado	haaprobado
pln_dgae	1	0.955	-0.022	-0.246	0.742	-0.132	0.309	-0.470	-0.053
plan	0.955	1	-0.169	-0.357	0.776	-0.261	0.245	-0.516	-0.201
creditos	-0.022	-0.169	1	0.779	-0.314	0.572	0.489	0.268	0.998
periodos	-0.246	-0.357	0.779	1	-0.592	0.401	0.307	0.728	0.789
generacion	0.742	0.776	-0.314	-0.592	1	-0.268	0.121	-0.587	-0.337
terminomaterias	-0.132	-0.261	0.572	0.401	-0.268	1	0.182	0.105	0.583
promedio	0.309	0.245	0.489	0.307	0.121	0.182	1	-0.031	0.480
hareprobado	-0.470	-0.516	0.268	0.728	-0.587	0.105	-0.031	1	0.292
haaprobado	-0.053	-0.201	0.998	0.789	-0.337	0.583	0.480	0.292	1

Figura 4.112. La matriz de correlación para el caso de deserción.

Y los datos fueron (*figura 4.113*):



row no.	deserto	cuenta	pln_dgae	plan	creditos	periodos	generacion	terminomat...	promedio	hareprobado	haaprobado
15788	NO	407098	1215	2006	0	1	2007	NO	5	5	0
15789	NO	407098	1215	2006	0	1	2007	NO	5	5	0
15790	NO	920008	1218	2006	342	18	1999	NO	7.950	24	42
15791	NO	930007	1218	2006	406	15	1999	SI	7.820	6	49
15792	SI	933398	1218	2006	334	11	2001	NO	7.820	7	39
15793	NO	940014	1218	2006	209	14	2000	NO	7.210	15	24
15794	SI	942136	1218	2006	144	13	1999	NO	7.320	12	17
15795	NO	950015	1218	2006	432	12	2001	SI	8.190	0	52
15796	NO	952136	1218	2006	410	15	2000	SI	7.980	4	49
15797	NO	952815	1218	2006	90	10	1999	NO	7.550	4	11
15798	NO	953284	1219	2006	179	15	1999	NO	6.990	22	21
15799	NO	953758	1218	2006	0	1	2007	NO	5	5	0
15800	NO	953806	1218	2006	394	14	1999	NO	8.230	10	47
15801	NO	960051	1219	2006	221	9	2003	NO	7.390	8	26
15802	NO	960058	1218	2006	282	8	2003	NO	7.970	2	34
15803	NO	960413	1219	2006	416	16	1999	SI	7.440	7	50
15804	NO	961081	1218	2006	157	15	2000	NO	7.040	26	19
15805	NO	961221	1218	2006	358	15	2000	NO	7.590	10	43

Figura 4.113. El conjunto de datos donde la variable a predecir es la de *deserto*.

Se observa que el modelo tiene un mejor desempeño con la variable a predecir *terminomaterias* que la de *deserto*.

Prueba

Teniendo ya los modelos. El siguiente paso es probarlos con otros datos, llámese *el conjunto de datos de* ***prueba*** para saber si el modelo acierta en un número de casos aceptable. Recuérdese que estos modelos se entrenaron con los datos de las tablas *alumnodesercion0*, *1* y *2*; correspondientes a los semestres 2007-1, 2007-2 y 2008-1, respectivamente. Los datos de prueba son del semestre *2008-2*.

Los datos de prueba se obtuvieron con la siguiente consulta la cual obtiene los registros de los alumnos que están cursando y que son de la generación 1994 hasta la generación 2002:

```
SELECT * FROM alumnodesercion4
WHERE generacion IN (1994,1995,1996,1997,1998,1999,2000,2001,2002)
AND ultimoperiodo='20082'
AND deserto='NO'
AND terminomaterias='NO'
```

Se obtiene hasta la generación 2002 ya que a partir de la generación 2003 no se cuenta con los suficientes datos para un buen aprendizaje. Con pocos datos, evidentemente no se puede predecir con tanta confianza que cuando se tienen muchos datos.

Ahora lo que se hace es generar otro árbol de procesos que cargue los datos de prueba (de la consulta *SQL* anterior), el modelo generado y de ahí que obtenga las predicciones para esos datos de prueba. Como esos datos de prueba ya tienen registrado si desertan o si terminan todas sus materias, se podrá evaluar qué tan bien clasifica los registros los modelos generados.

Para el primer modelo que se generó con la variable *terminomaterias*, se exportaron los datos de la consulta anterior a un archivo *.csv* y sólo se usaron los atributos o columnas siguentes: *cuenta*, *pln_dgae*, *plan*, *creditos*, *periodos*, *generacion*, *promedio*, *hareprobado* y *haaprobado*. Cabe recalcar que ya no se incluye la variable

terminomaterias puesto que esa será la variable a predecir. La variable *deserto* no se incluye porque pondera los resultados, es decir, los resultados del modelo salen con un bajo desempeño.

Se abre un nuevo archivo en *Rapidminer* para hacer un nuevo árbol de procesos:

Sobre el operador *root* dar *clic* con el botón derecho del *mouse* y del menú contextual seleccionar *New Operator*, *IO*, *Examples*, *CSVExampleSource* (*figura 4.114*). El cual es un operador que carga un archivo en formato *.csv*.

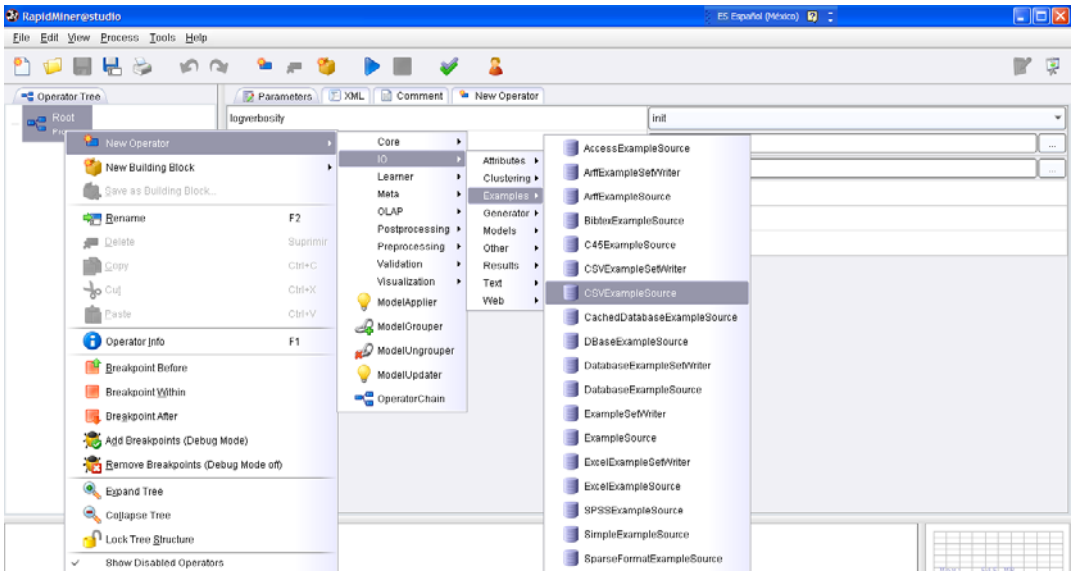


Figura 4.114. Seleccionando el operador que carga un archivo *.csv*.

Dar *clic* sobre este operador para visualizar los parámetros de este operador en la ventana del lado derecho. Dar *clic* sobre el botón que tiene puntos suspensivos … para indicar la ruta del archivo a abrir o cargar. En el parámetro *ID_name* teclear *cuenta* y en *ID_column* teclear 1. Esto indica que nuestra columna que funge como identificador es *cuenta* la cual se encuentra en la primera posición en el archivo. *Label_column* se deja como está en 0, ya que no se cuenta con ella desde que esa será la variable a predecir. El resto de los parámetros se dejan como estan. Por default. Véase la *figura 4.115*.

Nuevamente del menú contextual sobre *root* seleccionar *New Operator*, *Visualization*, *ModelVisualizer*. Esto con el fin de poder visualizar los datos cargados y corroborar que sean esos los que se desean o que se hayan cargado todos bien.

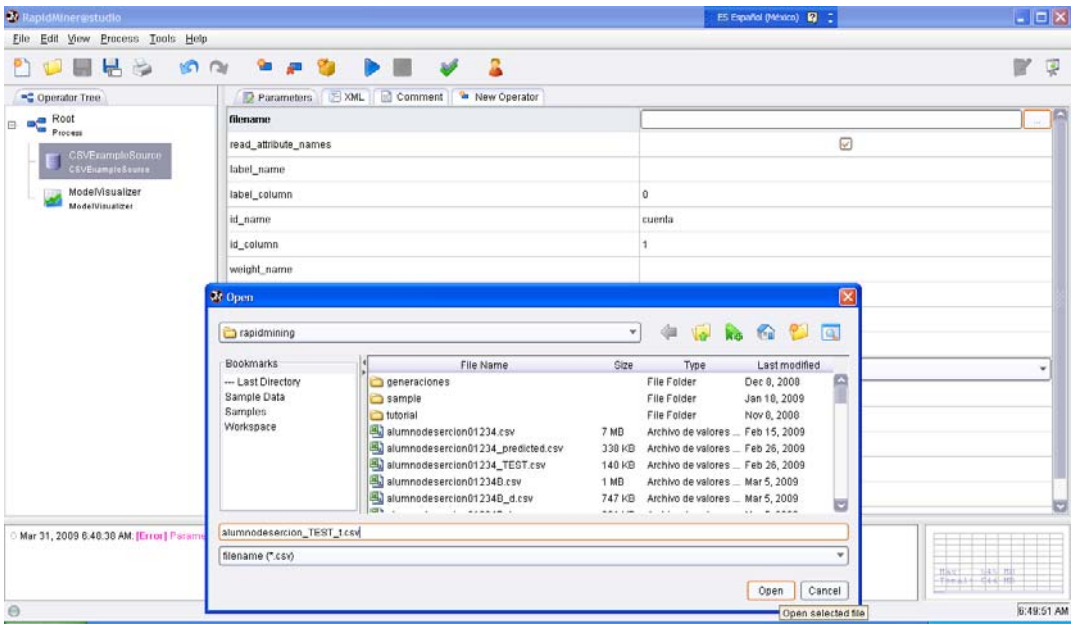


Figura 4.115. Modificando los parámetros del operador que carga el archivo con extensión *.csv*.

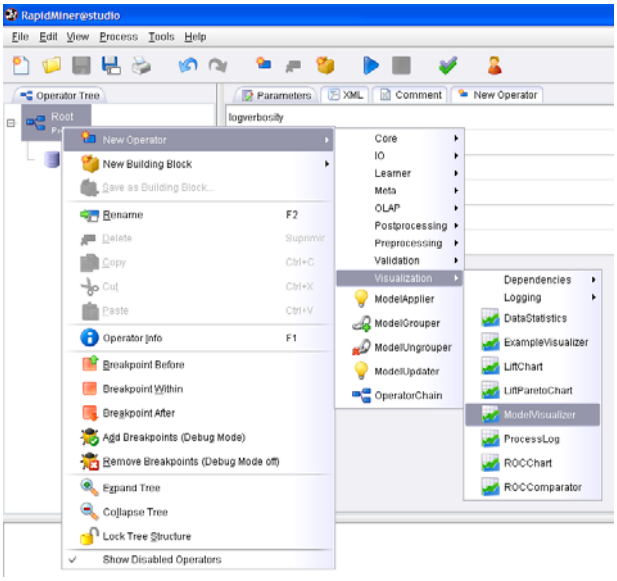


Figura 4.116. Seleccionando el operador que ayuda a visualizar los ejemplos cargados.

Luego seleccionar un operador que cargue el modelo generado anteriormente. Desde *root* con el menú contextual seleccionar *New Operator*, *IO*, *Models*, *ModelLoader* (*figura 4.117*).

Dar *clic* sobre el operador para visualizar los parámetros. Sólo se ve el botón de los puntos suspensivos (…) para cargar la ruta en donde se encuentra el modelo (*figura 4.118*).

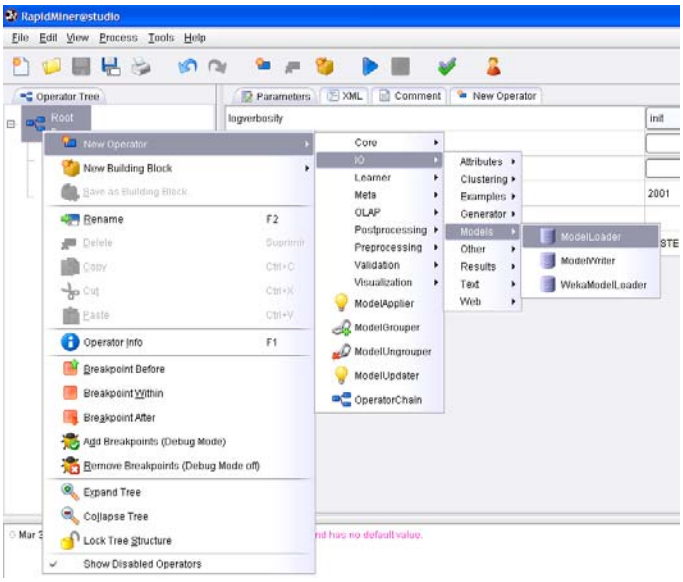


Figura 4.117. Seleccionando el operador que carga el modelo.

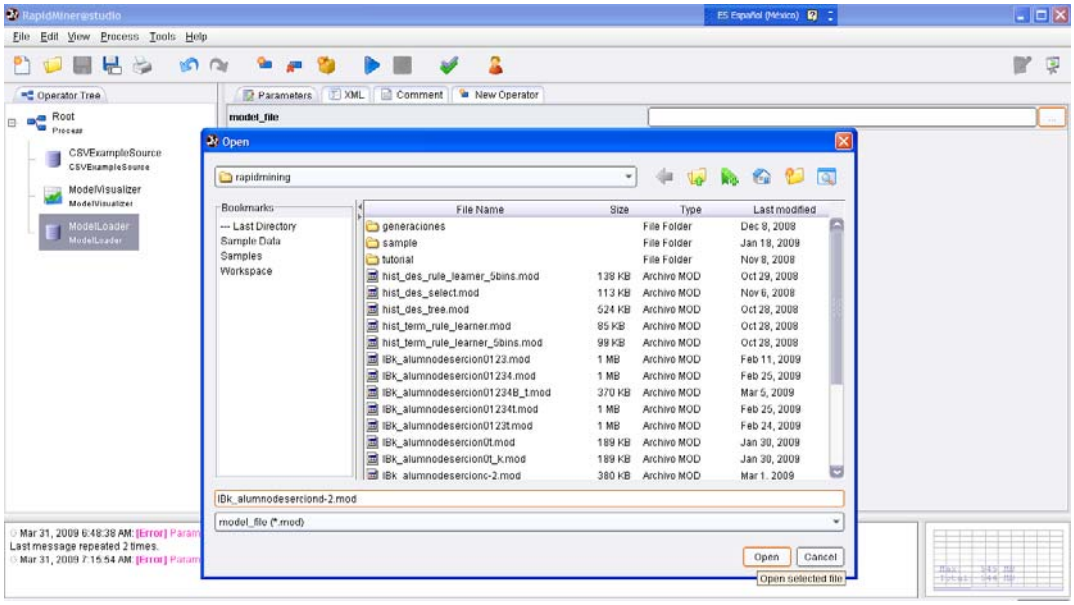


Figura 4.118. Indicando la ruta del modelo a cargar.

Finalmente seleccionar un operador que aplique el modelo a los datos de prueba. Dando *clic* derecho sobre *root* seleccionando del menú contextual se elige *New Operator*, *ModelApplier* (*figura 4.119*). Seleccionar este operador para visualizar sus parámetros

en la ventana del lado derecho y marcar las casillas faltantes que son la de *keep_model* y *create_view* (*figura 4.120*).

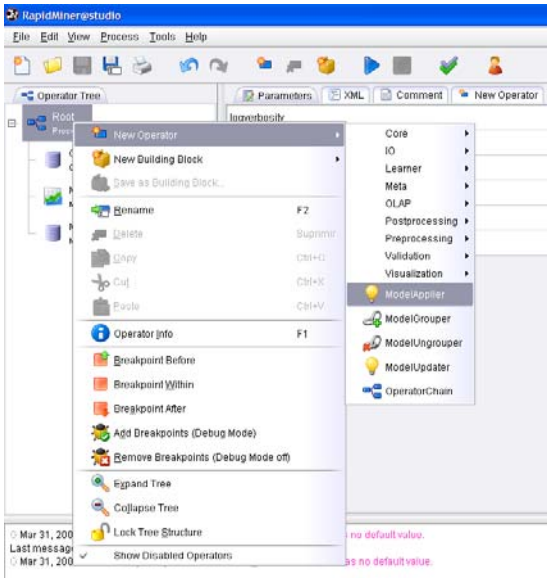


Figura 4.119. Seleccionando el operador que aplique el modelo.

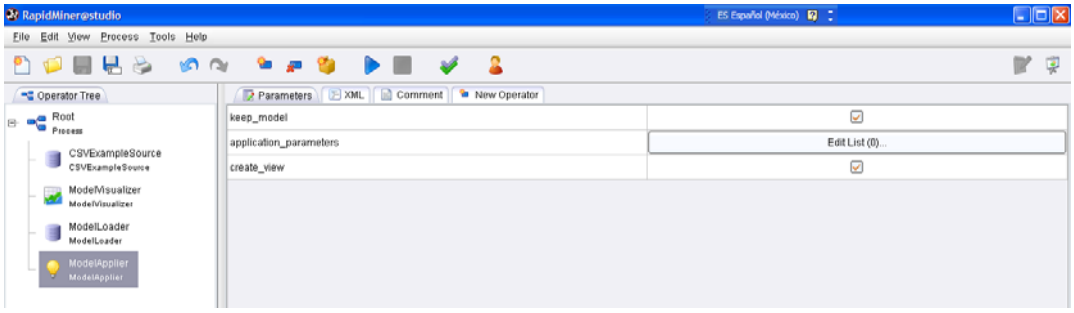


Figura 4.120. Seleccionando los parámetros del operador que aplica el modelo o *ModelApplier*.

Adicionalmente será cómodo agregar un operador que escriba los resultados a un archivo *.csv* para ya no tener necesidad de guardarlos y así poder cargarlos directamente a la base de datos. La razón es para que se pueda saber en cuántos casos el modelo acertó en la predicción o clasificación de nuestros registros de prueba con el fin de corroborar la efectividad del modelo.

Dando *clic* con el botón derecho del *mouse*, del menú contextual seleccionar *New Operator*, *IO*, *Examples*, *CSVExampleSetWriter* (*figura 4.121*).

Dando *clic* sobre este operador se visualiza los parámetros del mismo. Con el botón de los puntos suspensivos … seleccionar el nombre del archivo sobre el cual se desea escribir los resultados y en el parámetro *column_separator* teclear una coma, ya que por default se tiene un punto y coma (*figura 4.122*).

Luego dar *clic* sobre el botón de la palomita verde para corroborar que el proceso esté correcto y posteriormente dar clic sobre el botón de *play* para iniciar el proceso (*figura 4.123*). En ese instante, *Rapidminer* preguntará si se desea guardar el proceso.

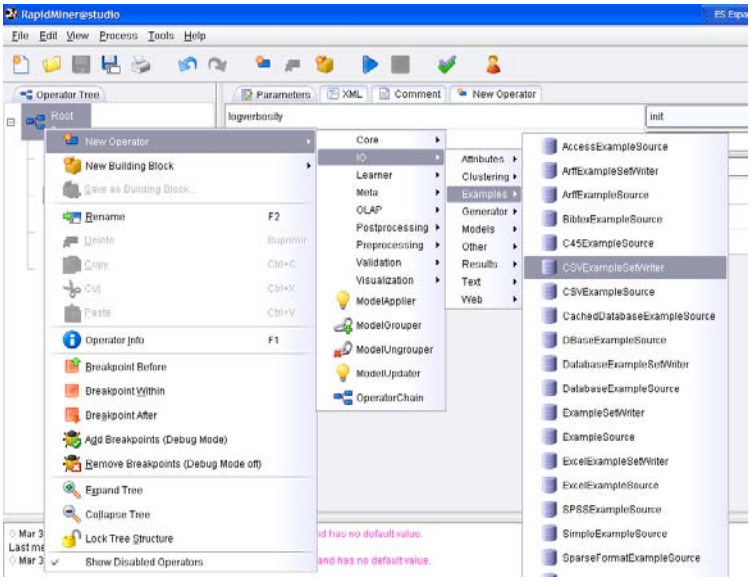


Figura 4.121. Seleccionando un operador que escribe los resultados en un archivo con formato *.csv*.

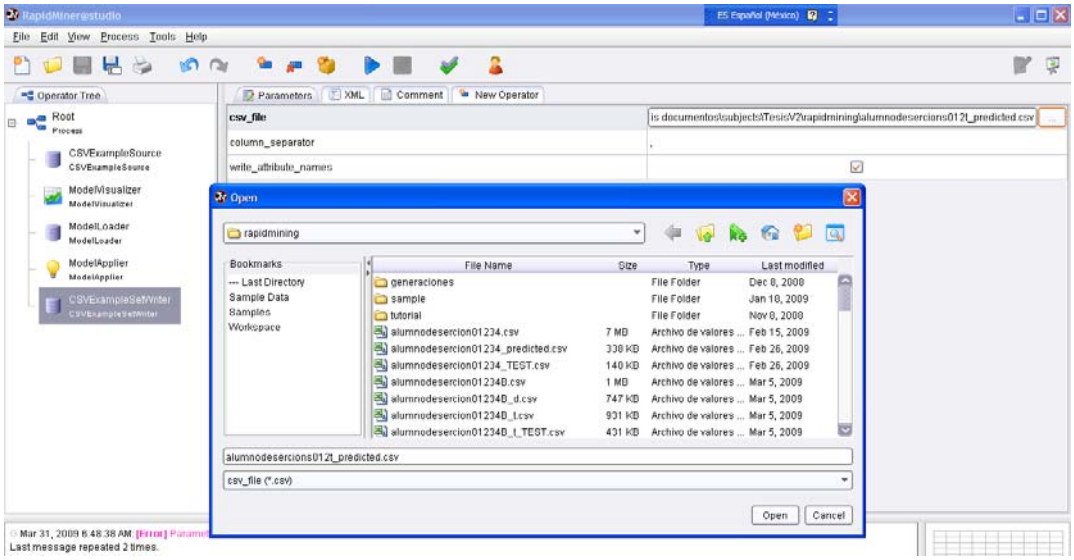


Figura 4.122. Seleccionando el archivo a escribir y seleccionando la coma como separador.

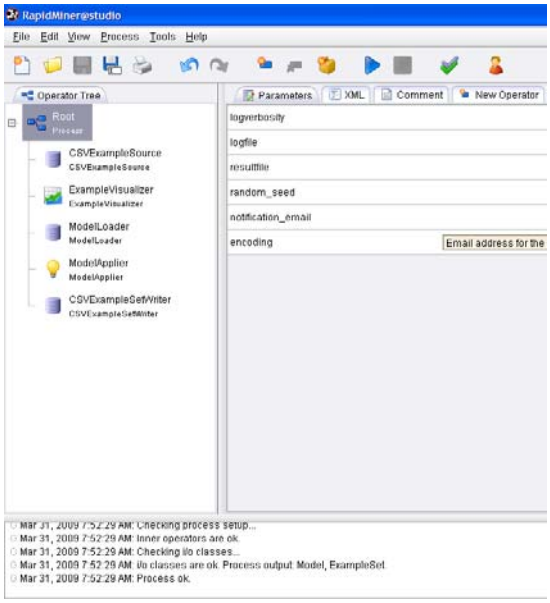
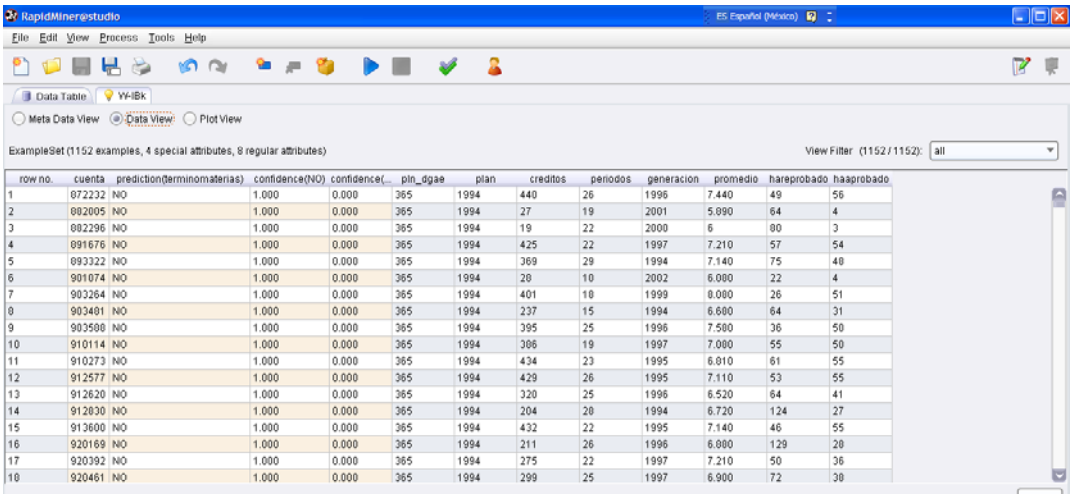


Figura 4.123. Dar *clic* sobre la palomita verde para verificar que el proceso esté bien y luego sobre el botón de *play* para iniciar el proceso.

Una vez que termina el proceso se muestran los resultados de las predicciones, es decir, *Rapidminer* automáticamente agrega tres nuevas columnas: una con el resultado de la predicción y las otras dos con el grado de confianza de ser, en este caso, un "sí" o un "no". Estas dos últimas columnas no son tan contundentes como para afirmar de la veracidad. Lo mejor es comprobar estos resultados con los *reales* para observar la efectividad del modelo.

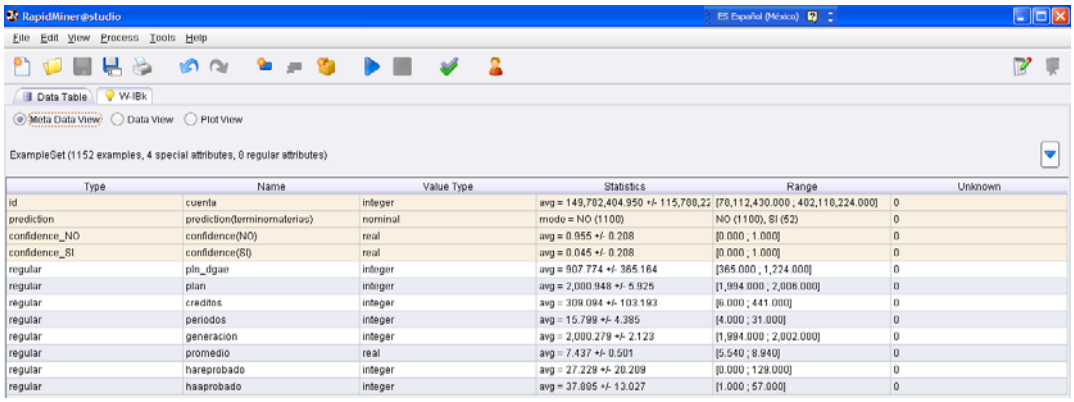
En la siguiente *figura 4.124* se observan los datos con los resultados de predicción



row no.	cuenta	prediction(terminomaterias)	confidence(NO)	confidence(...	pin_dgae	plan	creditos	periodos	generacion	promedio	hareprobado	haaprobado
1	072232	NO	1.000	0.000	365	1994	440	26	1996	7.440	49	56
2	082005	NO	1.000	0.000	365	1994	27	19	2001	5.890	64	4
3	082296	NO	1.000	0.000	365	1994	19	22	2000	6	80	3
4	091676	NO	1.000	0.000	365	1994	425	22	1997	7.210	57	54
5	093322	NO	1.000	0.000	365	1994	369	29	1994	7.140	75	48
6	901074	NO	1.000	0.000	365	1994	28	10	2002	6.000	22	4
7	903264	NO	1.000	0.000	365	1994	401	18	1999	8.000	26	51
8	903401	NO	1.000	0.000	365	1994	237	15	1994	6.600	64	31
9	903508	NO	1.000	0.000	365	1994	395	25	1996	7.500	36	50
10	910114	NO	1.000	0.000	365	1994	306	19	1997	7.000	55	50
11	910273	NO	1.000	0.000	365	1994	434	23	1995	6.010	61	55
12	912577	NO	1.000	0.000	365	1994	429	26	1995	7.110	53	55
13	912620	NO	1.000	0.000	365	1994	320	25	1996	6.520	64	41
14	912830	NO	1.000	0.000	365	1994	204	20	1994	6.720	124	27
15	913600	NO	1.000	0.000	365	1994	432	22	1995	7.140	46	55
16	920169	NO	1.000	0.000	365	1994	211	26	1996	6.000	129	28
17	920392	NO	1.000	0.000	365	1994	275	22	1997	7.210	50	36
18	920461	NO	1.000	0.000	365	1994	299	25	1997	6.900	72	38

Figura 4.124. Viendo los datos con sus respectivas predicciones.

Dando *clic* sobre *Meta Data View* se podrá ver datos de las variables que se usan, algunas estadísticas, el rango de sus valores además de saber si hay datos desconocidos (*figura 4.125*).

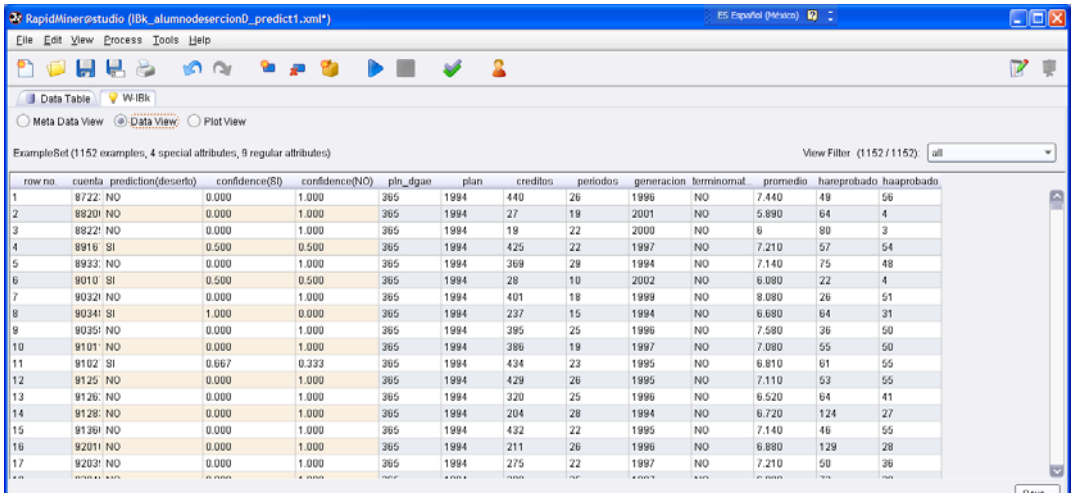


Type	Name	Value Type	Statistics	Range	Unknown
id	cuenta	integer	avg = 149,782,404.950 +/- 115,788,22	[78,112,430.000 ; 402,110,224.000]	0
prediction	prediction(terminomaterias)	nominal	mode = NO (1100)	NO (1100), SI (52)	0
confidence_NO	confidence(NO)	real	avg = 0.955 +/- 0.208	[0.000 ; 1.000]	0
confidence_SI	confidence(SI)	real	avg = 0.045 +/- 0.208	[0.000 ; 1.000]	0
regular	pln_dgae	integer	avg = 907.774 +/- 365.164	[365.000 ; 1,224.000]	0
regular	plan	integer	avg = 2,000.948 +/- 5.925	[1,994.000 ; 2,008.000]	0
regular	creditos	integer	avg = 309.094 +/- 103.193	[6.000 ; 441.000]	0
regular	periodos	integer	avg = 15.799 +/- 4.385	[4.000 ; 31.000]	0
regular	generacion	integer	avg = 2,000.279 +/- 2.123	[1,994.000 ; 2,002.000]	0
regular	promedio	real	avg = 7.437 +/- 0.501	[5.540 ; 8.940]	0
regular	hareprobado	integer	avg = 27.229 +/- 20.209	[0.000 ; 129.000]	0
regular	haaprobado	integer	avg = 37.895 +/- 13.027	[1.000 ; 57.000]	0

Figura 4.125. Viendo los metadatos, es decir, la información de las variables.

Ahora, se vuelve a hacer el mismo procedimiento pero con la variable etiqueta u objetivo o variable a predecir *desercion*. Lo que hay que cambiar básicamente es el archivo fuente cuyas variables son *cuenta*, *pln_dgae*, *plan*, *creditos*, *periodos*, *generacion*, *terminomaterias*, *promedio*, *hareprobado* y *haaprobado*. Se aclara que la variable *deserto* no se incluye en este archivo puesto que es la variable que se va a predecir.

Ahora para la variable *deserto* se obtienen los registros con su respectiva predicción (*figura 4.126*) así como los metadatos (*figura 4.127*):



row no.	cuenta	prediction(deserto)	confidence(SI)	confidence(NO)	pln_dgae	plan	creditos	periodos	generacion	terminomat...	promedio	hareprobado	haaprobado
1	8722	NO	0.000	1.000	365	1994	440	26	1996	NO	7.440	49	56
2	8820	NO	0.000	1.000	365	1994	27	19	2001	NO	5.890	64	4
3	8822	NO	0.000	1.000	365	1994	19	22	2000	NO	6	80	3
4	8916	SI	0.500	0.500	365	1994	425	22	1997	NO	7.210	57	54
5	8933	NO	0.000	1.000	365	1994	369	29	1994	NO	7.140	75	48
6	9010	SI	0.500	0.500	365	1994	28	10	2002	NO	6.080	22	4
7	9032	NO	0.000	1.000	365	1994	401	18	1999	NO	8.080	26	51
8	9034	SI	1.000	0.000	365	1994	237	15	1994	NO	6.680	64	31
9	9035	NO	0.000	1.000	365	1994	395	25	1996	NO	7.580	36	50
10	9101	NO	0.000	1.000	365	1994	386	19	1997	NO	7.080	55	50
11	9102	SI	0.667	0.333	365	1994	434	23	1995	NO	6.810	61	55
12	9125	NO	0.000	1.000	365	1994	429	26	1995	NO	7.110	53	55
13	9126	NO	0.000	1.000	365	1994	320	25	1996	NO	6.520	64	41
14	9128	NO	0.000	1.000	365	1994	204	28	1994	NO	6.720	124	27
15	9136	NO	0.000	1.000	365	1994	432	22	1995	NO	7.140	46	55
16	9201	NO	0.000	1.000	365	1994	211	26	1996	NO	6.880	129	28
17	9203	NO	0.000	1.000	365	1994	275	22	1997	NO	7.210	50	36

Figura 4.126. Predicciones para cada registro.

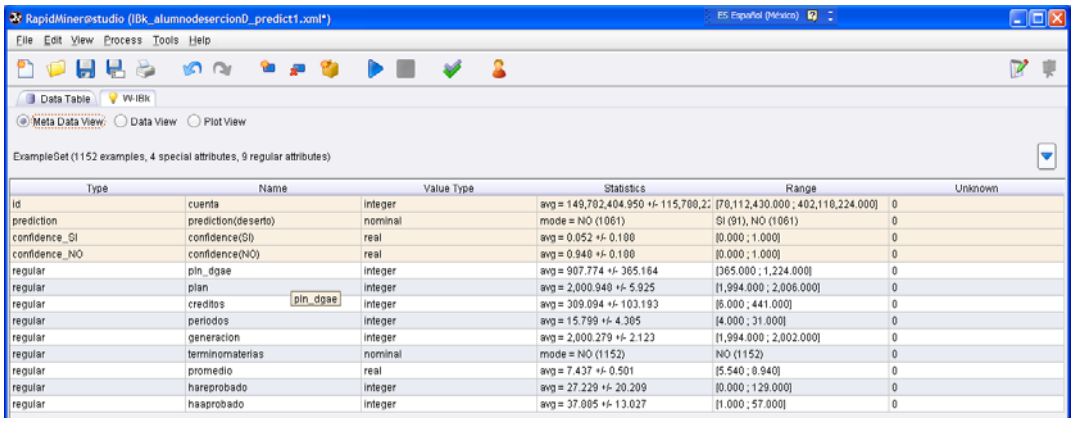


Figura 4.127. Metadatos de los registros predichos.

Las predicciones anteriores automáticamente se exportaron a un archivo en formato *.csv*, esos mismos se suben o se exportan a la base de datos para poder evaluar cuántos registros acertó el algoritmo de clasificación y tener una evaluación del desempeño del mismo.

Primero se crean dos tablas, una para las predicciones de *deserto* y otra para la variable *terminomaterias*. La tabla que contendrá las predicciones de deserción se generó con el siguiente script SQL:

```
CREATE TABLE `predictedd1d` (
  `indice` int(5) NOT NULL auto_increment,
  `cuenta` varchar(10) NOT NULL,
  `desertara` varchar(2) NOT NULL,
  `acerto` varchar(3) default NULL,
  `confianzaSI` float NOT NULL,
  `confianzaNO` float NOT NULL,
  `plan_dgae` int(4) NOT NULL,
  `plan` int(4) NOT NULL,
  `creditos` int(3) NOT NULL,
  `periodos` int(2) NOT NULL,
  `generacion` int(4) NOT NULL,
  `terminomaterias` varchar(2) NOT NULL,
  `promedio` float(5,2) NOT NULL,
  `hareprobado` int(3) NOT NULL,
  `haaprobado` int(3) NOT NULL,
  PRIMARY KEY  (`indice`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8
```

Y la tabla que contiene las predicciones de terminación de materias se generó con el siguiente script SQL:

```
CREATE TABLE `predictedd2t` (
  `indice` int(5) NOT NULL auto_increment,
  `cuenta` varchar(10) NOT NULL,
  `terminara` varchar(2) NOT NULL,
  `acerto` varchar(3) default NULL,
  `confianzaSI` float NOT NULL,
  `confianzaNO` float NOT NULL,
  `plan_dgae` int(4) NOT NULL,
  `plan` int(4) NOT NULL,
  `creditos` int(3) NOT NULL,
  `periodos` int(2) NOT NULL,
  `generacion` int(4) NOT NULL,
  `promedio` float(5,2) NOT NULL,
  `hareprobado` int(3) NOT NULL,
  `haaprobado` int(3) NOT NULL,
  PRIMARY KEY  (`indice`)) ENGINE=MyISAM  DEFAULT CHARSET=utf8
```

A ambas tablas se le agregó la columna *acerto* la cual indica con un "SÍ" o un "NO" si la predicción o clasificación hecha por el algoritmo *IBk* es acertada o no. Esta se compara con los datos reales de nuestra tabla *alumnodesercion*. Ir comparando estos datos manualmente lo hace complicado ya que tomaría mucho tiempo por lo que se programó un procedimiento para que lo hiciera. El procedimiento *acertod1.sql* para el caso de las deserciones se muestra a detalle en el anexo al final de este documento.

Y el procedimiento para la tabla de las predicciones de terminación de materias se llama *acertod2.sql* (sólo se cambian las variables, es muy parecido al procedimiento anterior) el cual se muestra asimismo en el anexo al final de este documento.

Una vez ejecutados estos procedimientos se obtienen las comparaciones para las deserciones (*figura 4.128*) y la terminación de materias (*figura 4.129*):

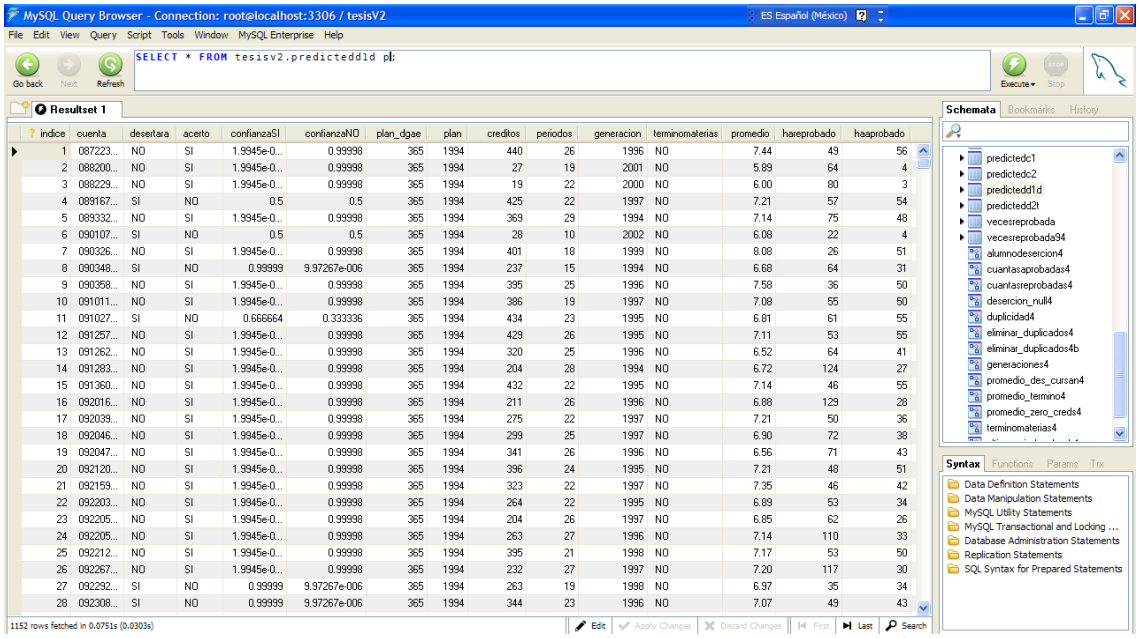


Figura 4.128. Visualizando las comparaciones de la tabla de predicciones para la deserción.

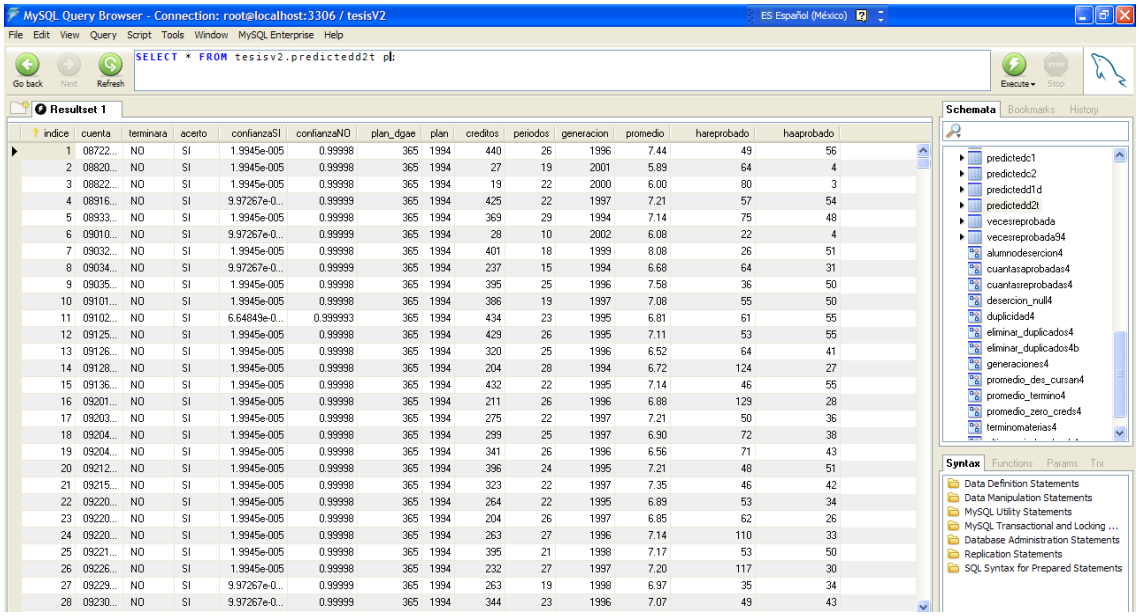


Figura 4.129. Visualizando las comparaciones de la tabla de predicciones para la terminación de materias.

Para evaluar el desempeño, se cuenta cuantos aciertos tuvo el clasificador *IBk* en cada tabla y obse tiene el porcentaje de aciertos.

Para ir contando los aciertos se hace una consulta de este tipo:

```
SELECT COUNT(cuenta) FROM tesisv2.predicteddld p
WHERE acerto='SI';
```

Entonces obteniendo el desempeño para cada tabla obse tiene que para la tabla de deserciones:

• Acertó 1061 --> **92.1%**
• No acertó: 91 --> 7.9%

Para la tabla de terminación de materias:
• Acertó: 1100 --> **95.49%**
• No acertó: 52 --> 4.51%

→ De un total de 1152 datos en ambas tablas.

Se observa que el desempeño del clasificador *IBk* es muy bueno. Ahora con base en lo anterior se procede a hacer predicciones.

Obteniendo el total de deserciones y de terminación de materias para cada generación se obtiene la siguiente tabla:

Generaciones.

1994: total 835, - a predecir: 19, desertaron: 603, terminomaterias: 213
1995: total 944, - a predecir: 35, desertaron: 642, terminomaterias: 267
1996: total 976, - a predecir: 42, desertaron: 870, terminomaterias: 64
1997: total 989, - a predecir: 61, desertaron: 848, terminomaterias: 80
1998: total 938, - a predecir: 69, desertaron: 774, terminomaterias: 95
1999: total 960, - a predecir: 81, desertaron: 735, terminomaterias: 144
2000: total 1059, - a predecir: 148, desertaron: 696, terminomaterias: 215
2001: total 971, - a predecir: 214, desertaron: 504, terminomaterias: 253
2002: total 1425, - a predecir: 483, desertaron: 484, terminomaterias: 458
---------- *Hasta esta generación se realizarán predicciones* ----------
2003: total 1649, - a predecir: 819, desertaron: 380, terminomaterias: 450
2004: total 1546, - a predecir:1098, desertaron: 302, terminomaterias: 146
2005: total 1742, - a predecir:1433, desertaron: 308, terminomaterias: 1
2006: total 1947, - a predecir:1632, desertaron: 315, terminomaterias: 0
2007: total 2083, - a predecir:1838, desertaron: 245, terminomaterias: 0
2008: total 2236, - a predecir:2157, desertaron: 79, terminomaterias: 0
- -

Haciendo predicciones desde la generacion 1994 hasta 2002 se tiene que:

Sumando predicciones de deserción sumando predicciones de terminación de materias

1994: desertaron: 603 + 4. No desertan: 15 Terminan:213, +15 sin saber

1995: desertaron: 642+11. No desertan: 24 Terminan:267, +24 sin saber

1996: desertaron: 870+7. No desertan: 34. Terminan:64+1. 34 sin saber.

1997: desertaron: 848+9. No desertan: 51. Terminan:80+1. 51 sin saber

1998: desertaron: 774+14. No desertan: 55. Terminan:95+1. 54 sin saber

1999: desertaron: 735+9. No desertan: 72. Terminan:144+72. 0 sin saber

2000: desertaron: 696+9. No desertan:139. Terminan:215+7. 132 sin saber

2001: desertaron: 504+7. No desertan:207. Terminan:253+17.190 sin saber

2002: desertaron: 484+21. No desertan:462. Terminan:458+27.435 sin saber

---

Para ir obteniendo el recuento de los registros por generación se hizo, por ejemplo, la consulta SQL:

```
SELECT * FROM predictedd1d
WHERE generacion='1994';
```

Se observa que han quedado registros sin saber si van a desertar o si van a terminar sus materias, esto se debe a que como se han utilizado dos variables distintas para la clasificación (deserción y terminación de materias), los datos obtenidos sin saber probablemente no encajaron o no fueron correctamente predichos por lo que quedan sobrando. Más adelante se muestra una modificación a la manera de predecir que es más exacta. Se quiere resaltar que no siempre a la primera el modelo va a salir perfecto, detrás de todo esto hay muchísimas pruebas que se hicieron y que no funcionaron muy bien, para llegar a esto se requirió de mucha experimentación y reacomodo de datos.

Dado que el clasificador ha aprendido aceptablemente bien con la parte de la deserción, los datos sin saber (que seguramente son los alumnos que no desertarían), bien se pueden sumar a la lista de los que terminan todas sus materias; entonces la suma final de los datos históricos más los que se obtuvieron para la predicción, quedaría como sigue:

- 1994: desertaron:607 (72.7%) Terminan:228 (27.3%)

- 1995: desertaron:653. (69.17%) Terminan:291 (30.83%)
- 1996: desertaron:877. (89.86%) Terminan: 99 (10.14%)
- 1997: desertaron:857. (86.65%) Terminan: 132 (13.35%)
- 1998: desertaron:788. (84%) Terminan: 150 (16%)
- 1999: desertaron:744. (77.5%) Terminan:216. (22.5%)
- 2000: desertaron:705. (66.58%) Terminan:354 (33.42%)
- 2001: desertaron:511. (52.63%) Terminan:460 (47.37%)
- 2002: desertaron:505. (35.44%) Terminan:920 (64.56%)

Graficando esto último se obtiene la siguiente *Figura 4.130*.

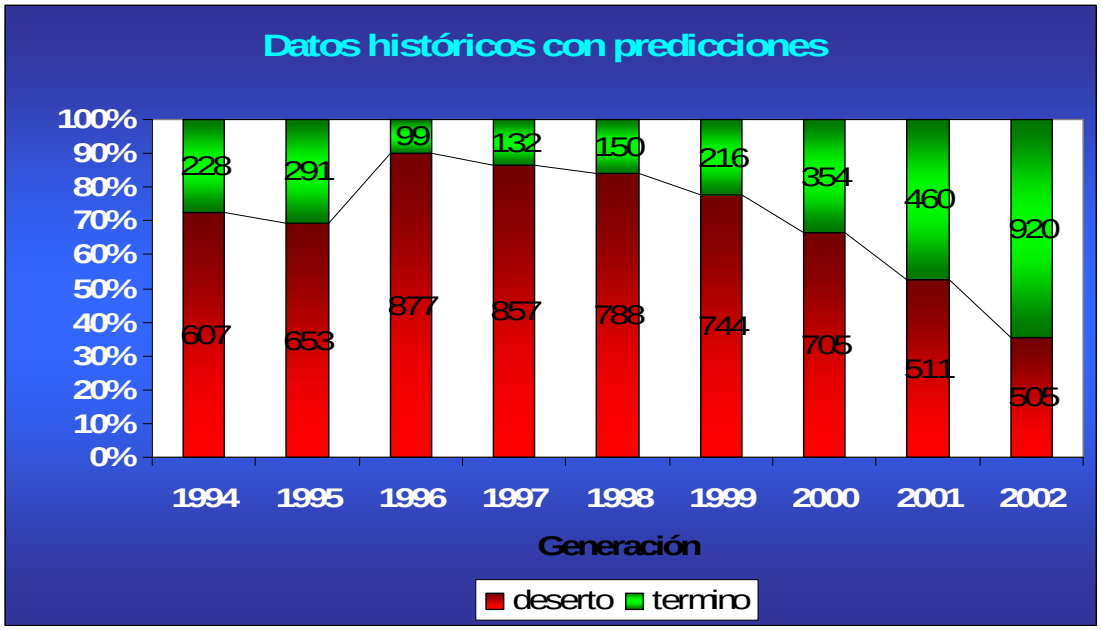


Figura. 4.130. Predicciones hasta la generación 2002 combinando las variables de deserción y terminación de materias con el clasificador *IBk*.

De la *Figura 4.130* se observa que la tendencia de alumnos que terminarán sus materias va en aumento. Después de esto se proporcionaron los nuevos datos de las calificaciones del semestre *2009-1*. Esto es de gran utilidad para evaluar el modelo, teniendo los nuevos datos se obtiene la siguiente *figura 4.131*:

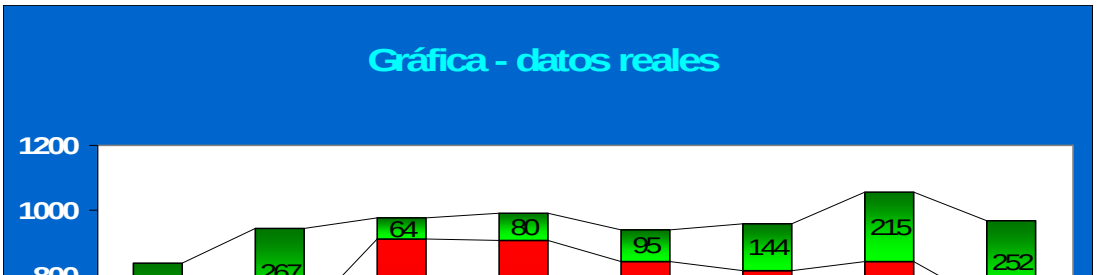


Figura 4.131. Datos reales de los alumnos que terminan sus materias y los que desertan por generación sin tomar en cuenta los alumnos que siguen cursando (que no han desertado pero que tampoco han terminado todas sus materias).

Se observa que en nuestras predicciones de la gráfica en la *figura 4.130*, resultaron ser demasiado optimistas, ya que en realidad hay menos alumnos que se esperaba que terminaran sus materias. Sin embargo, la tendencia se mantiene igual, es decir, que conforme avanza el tiempo, en las nuevas generaciones hay más y más alumnos que sí terminan todas sus materias.

Ya que se cuenta ahora con los datos más recientes, se procederá a hacer más predicciones. Ahora se incluirán más generaciones a predecir y, además, con un nuevo enfoque distinto: sólo se utilizará la variable de terminación de materias. Esto es porque la variable *terminomaterias* dice más que la variable *desercion* para este caso. Y es que si se predice solamente si el alumno termina o no sus materias, el no terminarlas, automáticamente significa que desertará.

Con los nuevos datos se volvieron a ejecutar los 12 procedimientos mostrados anteriormente para procesar los datos actualizados. Haciendo el mismo conteo por generación se obtiene la siguiente tabla:

Generación

1994: total 835, - a predecir: 0, desertaron: 622, terminomaterias: 213

1995: total 944, - a predecir: 1, desertaron: 676, terminomaterias: 267 // A predecir porque lleva 8 períodos

1996: total 976, - a predecir: 0, desertaron: 912, terminomaterias: 64

1997: total 989, - a predecir: 0, desertaron: 909, terminomaterias: 80

1998: total 938, - a predecir: 0, desertaron: 843, terminomaterias: 95

1999: total 960, - a predecir: 0, desertaron: 816, terminomaterias: 144

2000: total 1059, - a predecir: 3, desertaron: 841, terminomaterias: 215 |A PREDECIR

2001: total 971, - a predecir: 5, desertaron: 714, terminomaterias: 252 |

2002: total 1425, - a predecir: 263, desertaron: 658, terminomaterias: 504 |

2003: total 1647, - a predecir: 646, desertaron: 436, terminomaterias: 565 |

2004: total 1546, - a predecir: 909, desertaron: 352, terminomaterias: 285 |

2005: total 1736, - a predecir:1339, desertaron: 335, terminomaterias: 62 |

2006: total 1941, - a predecir:1564, desertaron: 377, terminomaterias: 0

2007: total 2063, - a predecir:1766, desertaron: 297, terminomaterias: 0

2008: total 2219, - a predecir:2050, desertaron: 169, terminomaterias: 0

2009: total 2204, - a predecir:2204, desertaron: 0, terminomaterias: 0

Para obtener los datos de entrenamiento se hizo lo siguiente:

Reflexionando en cuanto a cómo hacer que el modelo aprenda mejor sobre las deserciones y terminaciones de las materias, se llegó a la idea de que se puede contar con las evoluciones que va teniendo un alumno hasta terminar todas sus materias. Esto último marca una tendencia, la cual es totalmente apta para el aprendizaje del algoritmo. En las tablas *alumnodesercion* se cuentan con los alumnos que han terminado sus materias.

Lo mejor de todo esto es que se cuenta con alumnos que terminan sus materias hasta el semestre inmediato anterior *2009-1* por lo que eso significa que se cuenta con su avance académico desde el semestre *2007-1*. Bien se podría incluir más semestres anteriores a cambio de todavía mucho más tiempo invertido en esta tesis y más trabajo computacional, por supuesto, intenso; sin embargo, esto es suficiente para que un clasificador pueda aprender.

Primero se obtienen las cuentas de los alumnos que sí terminaron todas sus materias de la tabla *alumnodesercion4* la cual es la más reciente. Evidentemente no todas las cuentas terminaron antes, por lo que en las tablas anteriores, para las cuentas que apenas terminaron sus materias, en las tablas *alumnodesercion* anteriores estarán estas cuentas registradas como aquellas que aún no han terminado sus materias, es por eso que se tendrán registros como los que se muestran a continuación (*tabla 3*):

Tabla 3.

cuenta	pln_dgae	plan	creditos	periodos	generacion	terminomaterias	promedio	hareprobado	haaprobado
81359…	381	1994	386	11	2002	NO	7.61	12	48
81359…	381	1994	416	12	2002	NO	7.56	15	52
81359…	381	1994	444	13	2002	NO	7.7	17	56
81359…	381	1994	444	13	2002	NO	7.7	17	56
81359…	381	1994	444	13	2002	SI	7.7	17	56
85301…	381	1994	448	24	1994	NO	7.44	59	57
85301…	381	1994	448	24	1994	NO	7.44	59	57
85301…	381	1994	448	24	1994	NO	7.44	59	57
85301…	381	1994	448	24	1994	SI	7.44	59	57
86313…	805	1994	446	23	1995	NO	7.48	62	58
86313…	805	1994	446	23	1995	NO	7.48	62	58
86313…	805	1994	446	23	1995	NO	7.48	62	58
86313…	805	1994	446	23	1995	NO	7.48	62	58
86313…	805	1994	446	23	1995	SI	7.48	62	58
87205…	811	1994	449	14	1995	NO	7.71	22	55
87205…	811	1994	449	14	1995	NO	7.71	22	55
87205…	811	1994	449	14	1995	NO	7.71	22	55
87205…	811	1994	449	14	1995	NO	7.71	22	55
87205…	811	1994	449	14	1995	SI	7.71	22	55
87205…	365	1994	447	23	1994	NO	7.05	59	57
87205…	365	1994	447	23	1994	SI	7.05	59	57
87247…	365	1994	440	21	1996	NO	7.8	61	56
87247…	365	1994	449	22	1996	NO	7.77	61	57
87247…	365	1994	449	22	1996	NO	7.77	61	57

Tabla 4.1. Muestra del archivo de entrenamiento que tiene los resgistros históricos de los alumnos que sí terminaron todas sus materias.

Partiendo de que se tienen cuatro vistas minables (*alumnodesercion0*, *alumnodesercion1*, …) y que cada una de estas tablas tienen los datos actualizados hasta los semestres 2007-1, 2007-2, … respectivamente. Para que el modelo aprenda bien a partir de estos datos históricos, se pone en todos los valores de la columna *terminomaterias* el valor "SI" de aquellos alumnos que ya terminaron todas sus

materias, de tal forma que el modelo vaya aprendiendo la evolución y características de aquellos alumnos que sí terminan sus materias.

Para llegar a la tabla anterior, como se estaba mencionando, se obtuvieron primero las cuentas de los alumnos que sí terminaron todas sus materias de la tabla más reciente que es la de *alumnodesercion4*, una vez teniendo esas cuentas, obse tiene los registros de las demás tablas anteriores para aquellas cuentas que terminaron todas sus materias (*alumnodesercion0*, *alumnodesercion1*, …). La consulta SQL quedaría como sigue (como esta consulta abarca como 10 páginas, se omitieron los números de cuenta además por razones de confidencialidad):

```
WHERE generacion='1994';
SELECT * FROM alumnodesercion0
WHERE cuenta IN (0872…,4040…,…)
UNION
SELECT * FROM alumnodesercion1
WHERE cuenta IN (087205…,…)
UNION
SELECT * FROM alumnodesercion2
WHERE cuenta IN (087205…,…)
UNION
SELECT * FROM alumnodesercion
WHERE cuenta IN (087205…,…)
UNION
SELECT * FROM alumnodesercion4
WHERE cuenta IN (087205…,…)
ORDER BY cuenta,ultimoperiodo;
```

Además se agregó lo siguiente:

```
SELECT * FROM alumnodesercion4
WHERE deserto='SI'
ORDER BY generacion,plan,haaprobado
```

El cual son un conjunto de alumnos que ya desertaron. Estos datos se agregan junto con los datos de los alumnos que sí terminaron sus materias de tal forma que el clasificador aprenda las características de un alumno que sí termina todas sus materias y de aquél que deserta. Estos datos conforman el conjunto de datos de entrenamiento o *training set*. Y la consulta que se hace para obtener los datos de prueba es la siguiente:

```
SELECT * FROM alumnodesercion4
WHERE deserto='NO'
AND terminomaterias='NO'
ORDER BY generacion,plan,haaprobado
```

Este conjunto de prueba obtiene a todos los alumnos que están cursando (los que no han desertado ni han terminado todas sus materias). De estas predicciones se separarán por generaciones para obtener la gráfica de las siguientes predicciones.

Se construye un árbol de procesos en *Rapidminer* idéntico al que se tiene en la *figura 4.102*. Además se agregó otro operador que sirve para obtener más estadísticas. Para seleccionarlo, del menú contextual dando *clic* derecho del *mouse* se selecciona *New Operator*, *Visualtization*, *DataStatistics*. El árbol como queda, se muestra en la *figura 4.132*.

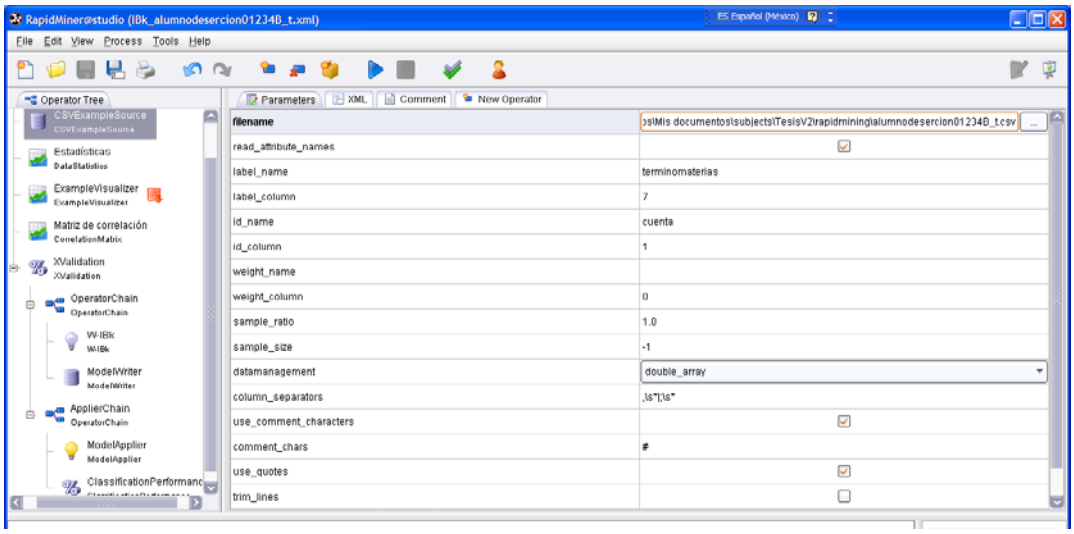


Figura 4.132. El árbol de procesos para predicciones con base en la variable *terminomaterias*.

De la misma *figura 4.132*, dando *clic* sobre el operador que carga nuestro archivo de datos, que contiene el conjunto de datos de entrenamiento, se verá a nuestra derecha la ventana de parámetros de este operador. Como se ha mencionado anteriormente, dar *clic* sobre el botón de los puntos suspensivos … para elegir la ruta del archivo. Se deja marcada la casilla *read_attribute_names* con la cual se indica que se lean los nombres de los atributos del archivo (es importante esto para que *Rapidminer* sepa cual columna es cual). En *label_name* teclear *terminomaterias* la cual es la columna o atributo a predecir. En *label_column* teclear 7 ya que es el número de la columna (contada de

izquierda a derecha a partir del 1) en donde se encuentra la columna *terminomaterias*. En *id_name* teclear *cuenta* el cual es el identificador único que distingue un alumno de otro y en *id_column* teclear 1 ya que es la primera columna de nuestro archivo del conjunto de datos de entrenamiento.

La siguiente modificación se hace sobre el operador *ModelWriter* en donde haciendo *clic* sobre el mismo, del lado derecho se visualizan los parámetros. Sólo hay que seleccionar el nombre del modelo para guardarlo y posteriormente usarlo con los datos de prueba. Véase la *figura 4.133.*

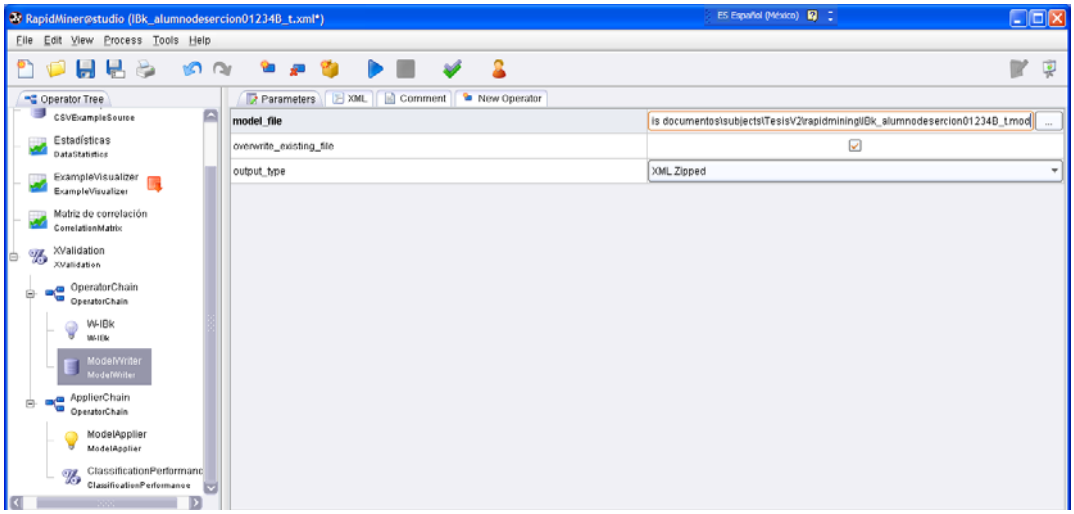


Figura 4.133. Seleccionando el nombre del modelo para guardarlo.

Verificar que el árbol esté bien armado y se procede a ejecutarlo.

Al terminar de ejecutarse, obse tiene los siguientes resultados de la evaluación del modelo (*figura 4.134*):

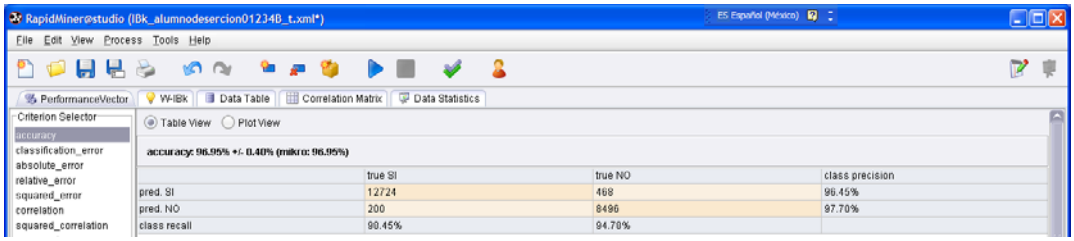


Figura 4.134. Viendo los resultados del desempeño del modelo.

accuracy: 96.95% +/- 0.40% (mikro: 96.95%)

classification_error: 3.05% +/- 0.40% (mikro: 3.05%)

correlation: 0.937 +/- 0.008 (mikro: 0.937)

Y la tabla de la matriz de confusión es la siguiente:

	true SI	**true NO**	**class precision**
pred. SI	12724	468	**96.45%**
pred. NO	200	8496	**97.70%**
class recall	98.45%	94.78%	

De la tabla anterior se observa que el modelo acierta en el 96.45% de los casos en cuanto a las predicciones o clasificaciones de los que sí terminan sus materias y en un 97.70% de los que no terminan (o desertan). Esta evaluación la hace *Rapidminer* automáticamente retomando un grupo aleatorio de datos y probándolos con el modelo tal y como se hizo manualmente comparando con SQL los datos clasificados con los datos reales. Esto es muy útil para decidir si el modelo se usa o no.

La matriz de correlación es la siguiente (*figura 4.135*):

Name	pln_dgae	plan	creditos	periodos	generacion	promedio	hareprobado	haaprobado
pln_dgae	1	0.768	0.084	0.006	0.296	0.102	0.123	0.067
plan	0.768	1	0.003	0.013	0.359	0.038	0.169	0.001
creditos	0.084	0.003	1	0.556	0.001	0.442	0.006	0.996
periodos	0.006	0.013	0.556	1	0.142	0.235	0.395	0.569
generacion	0.296	0.359	0.001	0.142	1	0.004	0.264	0.004
promedio	0.102	0.038	0.442	0.235	0.004	1	0.001	0.436
hareprobado	0.123	0.169	0.006	0.395	0.264	0.001	1	0.009
haaprobado	0.067	0.001	0.996	0.569	0.004	0.436	0.009	1

Figura 4.135. Matriz de correlación.

Ahora se procederá a crear un árbol de procesos para aplicar el modelo con los datos de prueba cuya muestra está en la *tabla 3*. Este árbol es parecido al árbol de la *figura 4.105*. Entonces se indica la ruta del archivo de los datos de prueba a cargar en los parámetros del operador *CSVExampleSource* (*figura 4.136*) y además se vuelve a indicar cuál es la columna identificadora en *id_name*, en este caso es *cuenta* y se encuentra en la columna 1 (*id_column: 1*).

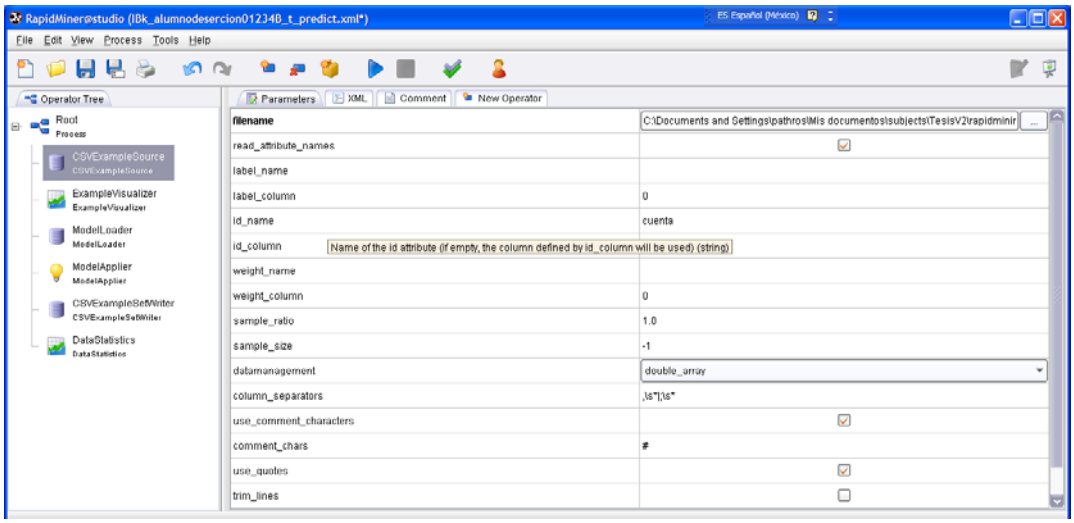


Figura 4.136. Indicando la ruta del archivo que contiene los datos de prueba así como los parámetros de las columnas.

Luego en el operador *ModelLoader* se indica la ruta del modelo generado en el proceso anterior (de la *figura 4.117*). Véase la *figura 4.137*.

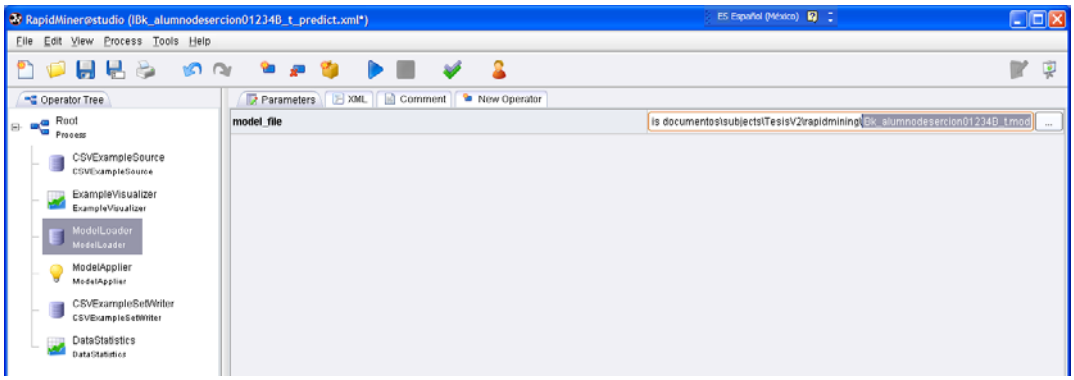


Figura 4.137. Indicando la ruta del modelo a cargar generado en el árbol de procesos de entrenamiento para aplicarse a los datos de prueba.

Finalmente se especifica la ruta del archivo sobre el cual se desea que se escriban los resultados en el operador *CSVExampleSetWriter*. Véase la *figura 4.138*.

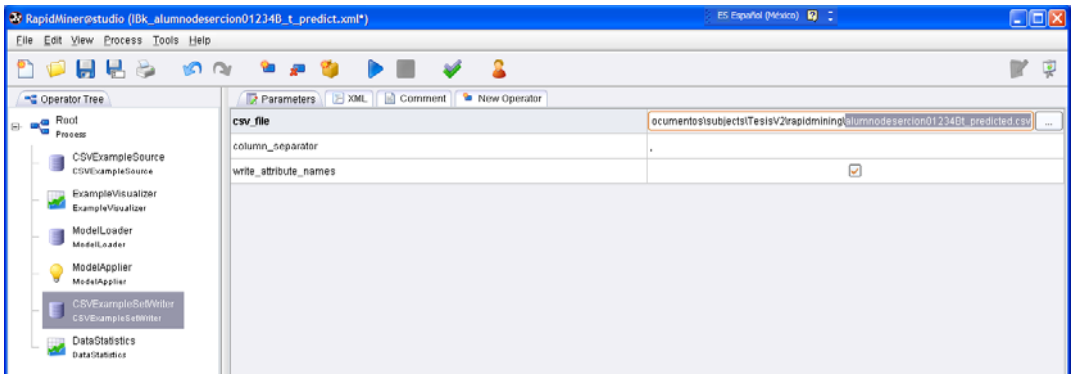


Figura 4.138. Indicando la ruta y el nombre del archivo para guardar los resultados.

Nótese que en este árbol se agregó otro operador (*DataStatistics*) que sirve sólo para ver estadísticas de los datos.

Después de ejecutar el árbol de procesos se tiene los resultados de las predicciones o de las clasificaciones. En la siguiente *tabla 4.2* se tiene una muestra de los resultados:

Tabla 4.2:

pln_dgae	plan	creditos	periodos	generacion	promedio	hareprobado	haaprobado	cuenta	prediction(terminomaterias)
1223	2006	257	8	1995	7.8	3	30	91257***	NO
1210	2006	312	15	2000	7.21	22	38	97349***	SI
1202	2006	128	13	2000	7.12	16	15	400001***	NO
1190	2006	207	15	2000	7.31	20	24	97316***	NO
1223	2006	348	9	2001	8.44	7	41	98247***	SI
1223	2006	367	9	2001	8	9	43	98253***	SI
1218	2006	386	13	2001	7.93	7	46	93339***	SI
1215	2006	390	14	2001	8.41	8	49	98210***	SI
365	1994	278	13	2001	7.92	34	35	94000***	NO
1181	2006	178	10	2002	7.43	7	20	95207***	SI
1190	2006	219	10	2002	8.44	10	26	99248***	SI
1182	2006	227	10	2002	7.62	11	27	99312***	SI
1190	2006	236	11	2002	7.47	6	28	99226***	SI
1190	2006	274	10	2002	8.26	3	33	99018***	SI
1181	2006	304	13	2002	7.58	9	36	96157***	SI
1190	2006	319	13	2002	7.87	8	38	99230***	SI
1184	2006	358	14	2002	8.76	11	41	96250***	SI
408	1994	331	13	2002	8.07	19	42	99155***	SI
1184	2006	372	13	2002	7.8	10	43	99153***	SI
1190	2006	356	13	2002	8.14	11	43	99298***	SI
411	1994	351	14	2002	7.84	40	45	99214***	SI
1202	2006	375	13	2002	8.35	14	45	99271***	SI
1218	2006	378	14	2002	7.41	10	45	99190***	SI
1223	2006	378	13	2002	7.23	16	45	99043***	SI
1182	2006	383	13	2002	7.87	8	46	98276***	SI

1188	2006	395	14	2002	7.7	14	46	99145***	SI
1188	2006	394	13	2002	8.17	11	46	99300***	SI
1190	2006	382	14	2002	7.88	13	46	99074***	SI
1223	2006	381	13	2002	7.83	6	46	402035***	SI
1223	2006	379	13	2002	7.47	13	46	402112***	SI
1182	2006	389	14	2002	7.81	9	47	99065***	SI
1182	2006	389	14	2002	7.6	15	47	401006***	SI

Una vez obtenidos estos datos, se procede a ordenarlos por generación, luego se ordenan según desertarán o según si terminarán todas sus materias. Entonces se llega a la *tabla 4.3*:

Tabla 4.3.

generación	desertó	terminó
1994	607	228
1995	677	267
1996	877	99
1997	857	132
1998	788	150
1999	744	216
2000	843	216
2001	715	256
2002	871	554
2003	760	887
2004	788	758
2005	1040	696

Por lo que graficando se obtienen finalmente predicciones hasta la generación 2005 (*Figura 4.139*):

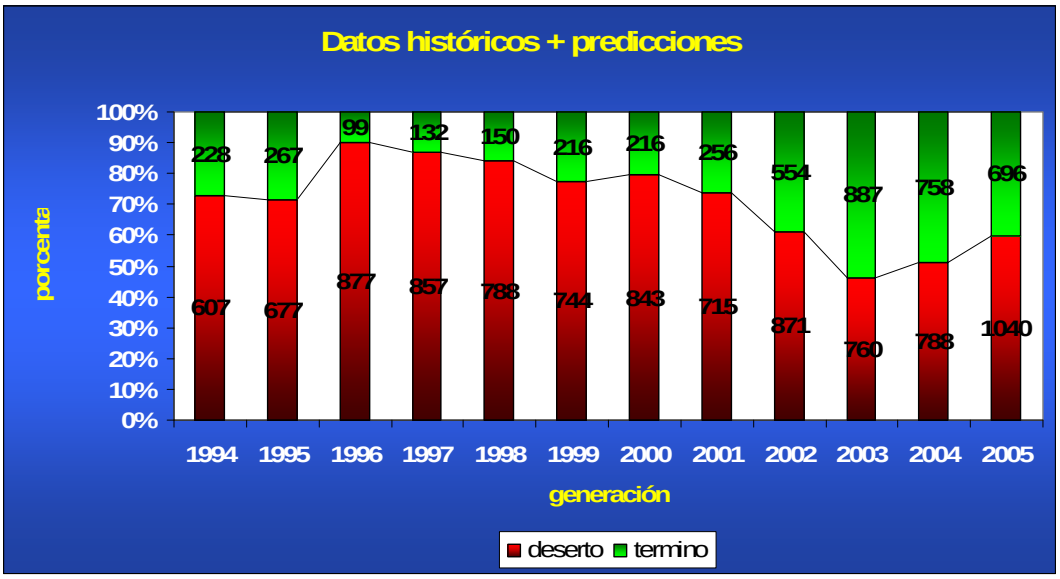


Figura 4.139. Predicciones de deserción y de terminación de materias hasta la generación 2005.

Esta gráfica tiene datos más realistas. Conserva la tendencia que se ha venido viendo en las gráficas anteriores. Se observa que la tendencia en el número de alumnos que sí terminan todas sus materias aumenta conforme van pasando las generaciones pero sólo hasta la generación 2003 la cual tiene y tendrá el mayor número de los que terminan todas sus materias. Cabe aclarar que los que terminan, no todos lo hacen en el mismo semestre; unos se retrasan y van terminando después. Para las generaciones 2004 y 2005 tiende a disminuir la tendencia.

4.3.2 Utilizando Árboles de decisión para encontrar *cuellos de botella*.

En este apartado se presentará la fase en donde se realiza la *Minería de Datos* utilizando los árboles de decisión. Se escogió este tipo de técnica porque es el método más fácil de utilizar y entender.

En este estudio se escogió primeramente el árbol de decisión *Random Forest* y esta vez se utilizará con fines de obtener reglas sobre la deserción dejando fuera un poco el atributo generación, no sólo se excluirá dicho atributo, se le dará un enfoque distinto; esta vez se tratará de encontrar la relación de deserción intentando encontrar cuellos de botella en las materias de las carreras de estos dos planes de estudios.

En la *figura 4.140* se muestra la construcción en *Rapidminer* de éste árbol, y en la *figura* 4.141 los modelos que se generaron. Cabe resaltar que ya se ha explicado detalladamente la construcción de árboles de procesos en *Rapidminer*, por lo se puede saltarnos estos pasos.

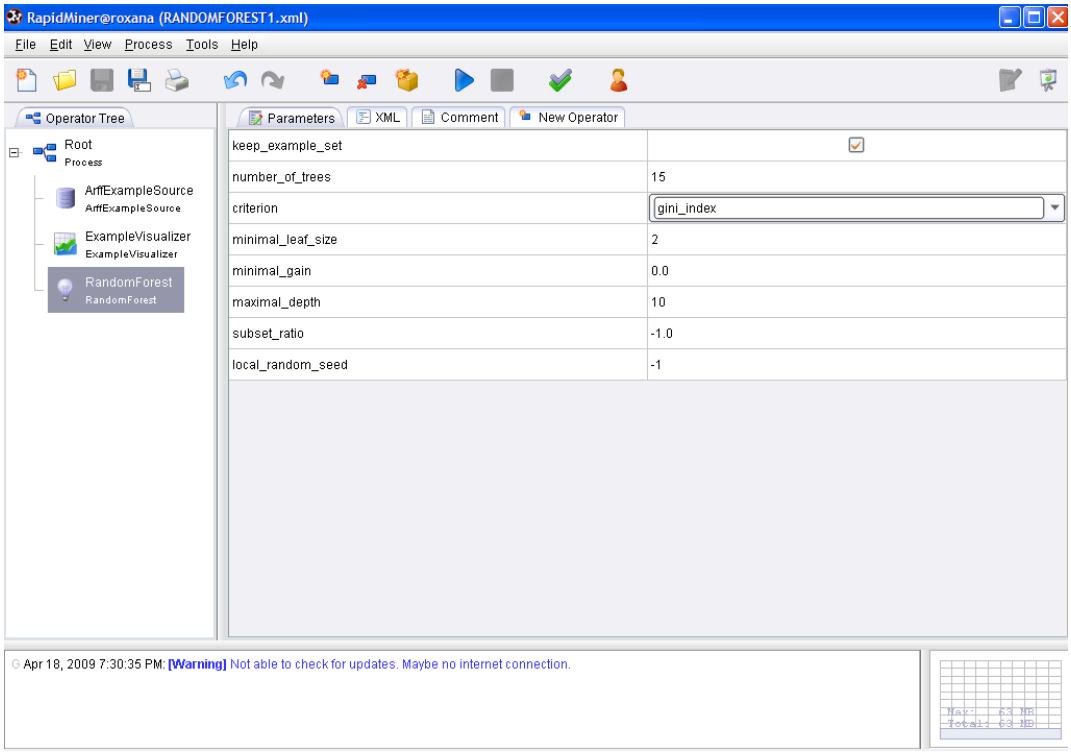


Figura 4.140. Construcción del árbol Random Forest.

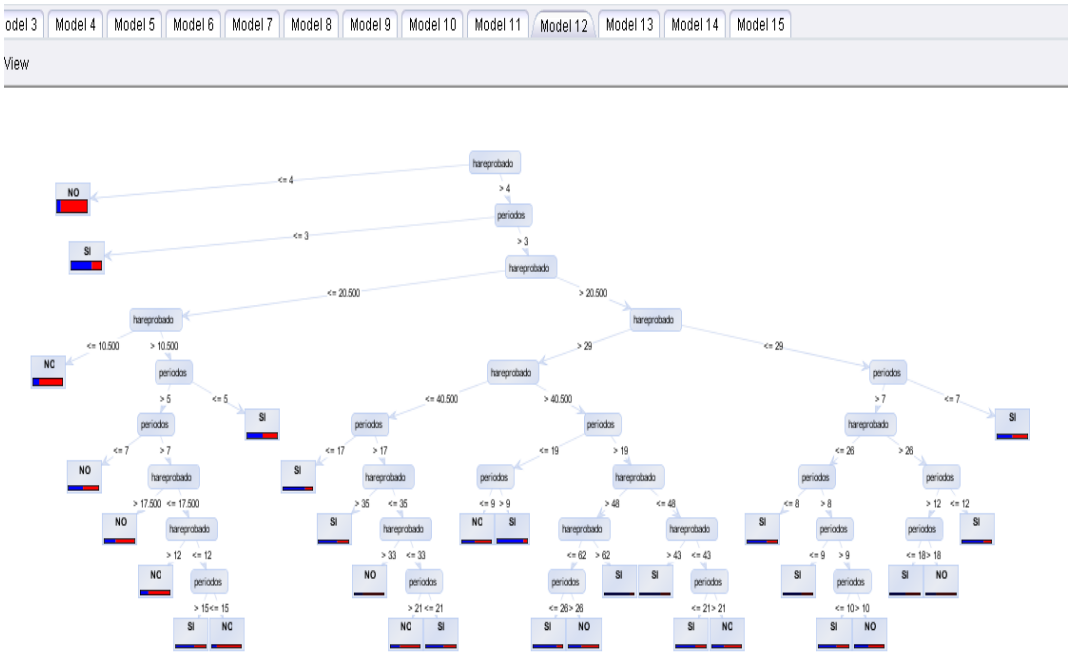


Figura 4.141. Árbol Random Forest. En este caso, se visualiza el árbol generado número 12.

Para nuestro caso se escogieron los modelos o árboles 7 y 12, ya que estos proporcionaban mas información al estar más ramificados.

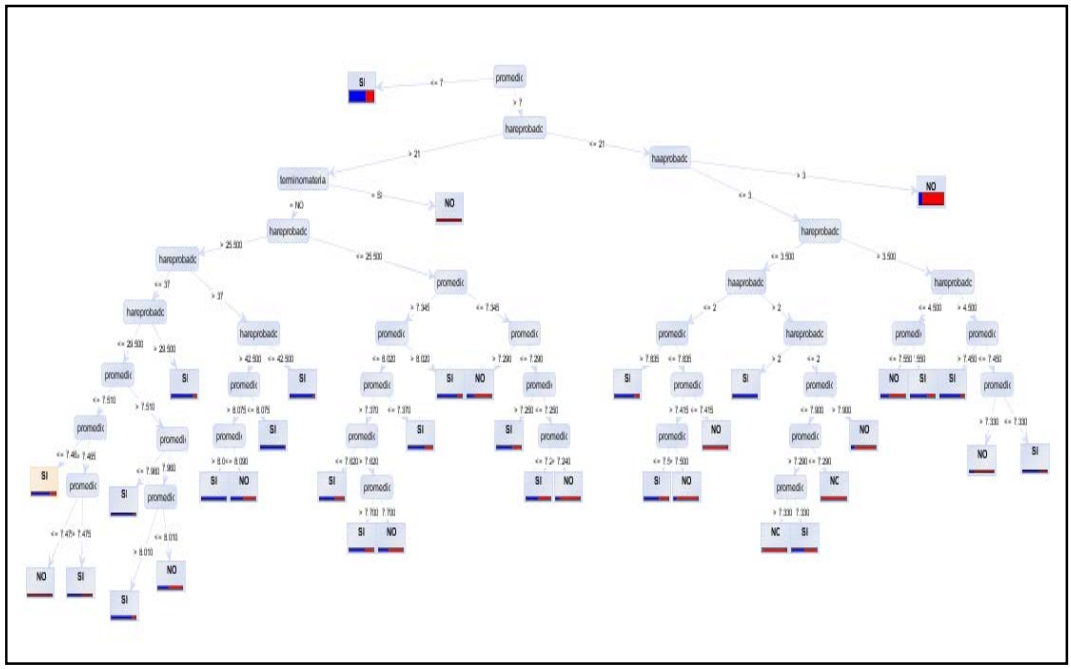


Figura. 4.142. Árbol Random Forest modelo o árbol 7.

Como se observa, son varias las ramificaciones que clasifican y/o agrupan las semejanzas que tiene los alumnos para que el atributo *deserto* sea “SI” o “NO”.

No todas las ramas son significativas para estudiarlas, es decir, no todas tienen trascendencia para proporcionar una regla ya que muchas de ellas muestran reglas con muy pocos datos; aunque a simple vista se observe que dicha rama muestre un 100% en el atributo que interesa (que en nuestro caso sería **deserto).** Sin embargo, las ramificaciones cuyas reglas abarcan pocos datos, son muy útiles para la detección de casos anómalos o extraordinarios.

Para visualizar cuáles ramas son significativas (resaltadas en ***negritas***), se añade un operador que convierta el árbol de decisión gráfico en un árbol mostrado parcialmente en texto (para ver el árbol *Tree1* completo, véase en el anexo al final de este documento):

```
Tree 1
promedio <= 7: SI {SI=5800, NO=2978}
promedio > 7
|   hareprobado <= 21
|   |   haaprobado <= 3
|   |   |   …
…           …
…           …
…           …
…           …
…           …
|   |   |   |   |   hareprobado > 42.500
|   |   |   |   |   |   promedio <= 8.075: SI {SI=423, NO=4}
|   |   |   |   |   |   promedio > 8.075
|   |   |   |   |   |   |   promedio <= 8.090: NO {SI=1, NO=1}
|   |   |   |   |   |   |   promedio > 8.090: SI {SI=20, NO=0}
|   |   terminomaterias = SI: NO {SI=0, NO=618}
```

La primera regla a estudiar es la rama que se muestra en la *figura 4.143*:

Se tomó esta rama ya que en la forma escrita mostró que el caso de incidencias en el atributo *deserto* era de **{SI=423, NO=4}** (los números de casos que sí desertan y de los que no).

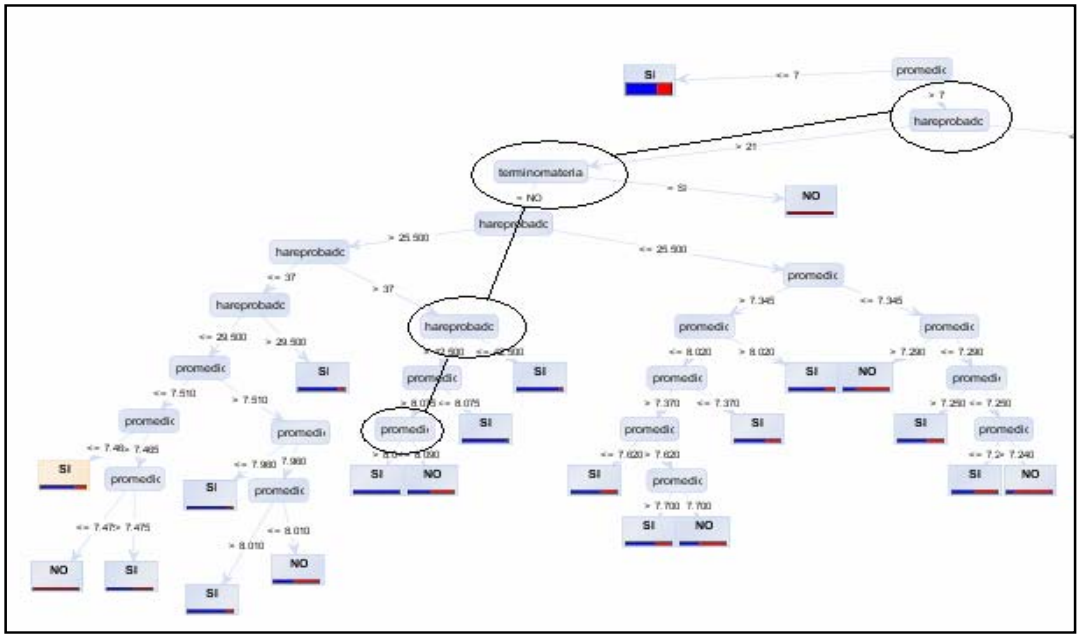


Figura. 4.143 Primera Regla a estudiar.

Convirtiendo ya sea la forma escrita o visual de dicha rama del árbol para estudiar los datos en MySQL Query Browser se obtiene el siguiente Query o consulta :

SELECT * FROM alumnodesercion4 WHERE hareprobado >42.5 AND promedio BETWEEN 7.01 AND 8.075 AND terminomaterias='NO';

Se le da la instrucción de que el rango del promedio sea 7.01, ya que no toma el valor de 7. En la Figura 4.144 Se muestra los datos arrojados por el *Query* o consulta concordando y validando lo que arrojó la regla (número de alumnos que tienen estas características)

A simple vista aún no se pueden conocer las características que tienen los alumnos desertan ya que, la *vista minable* con la cual se obtuvieron las reglas, no tiene toda la información de los alumnos por lo que se utilizan únicamente los números de cuenta de este *query* o consulta para utilizarlos en la tabla de *historias* donde se encuentran todos los registros académicos de cada uno de los alumnos.

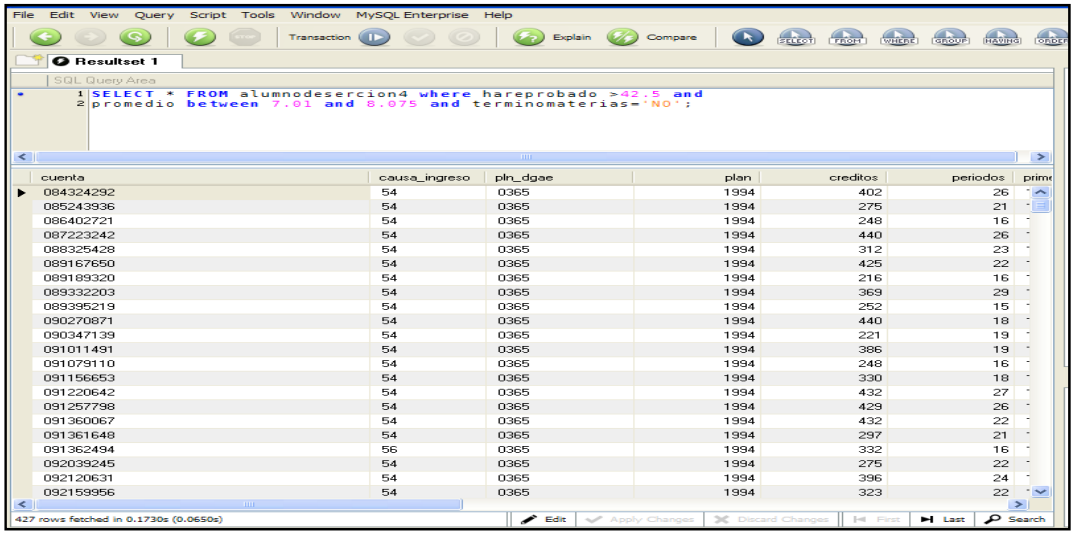


Figura. 4.144. Visualización de los datos obtenidos por la regla.

Esto se realizó de la siguiente manera:

- Puesto que realizar una consulta SQL que relacionara las dos tablas era demasiado pesado para el tipo de cómputo con el que se realizó dicho estudio, se realizó por medio de una consulta *SQL* usando *PHP* y se ejecutó desde el servidor local *AppServ*. *Vease Fig. 4.145*:

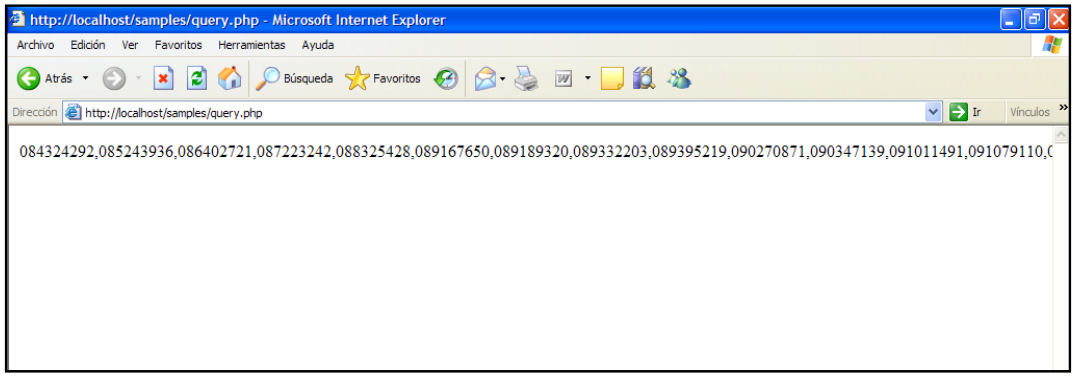
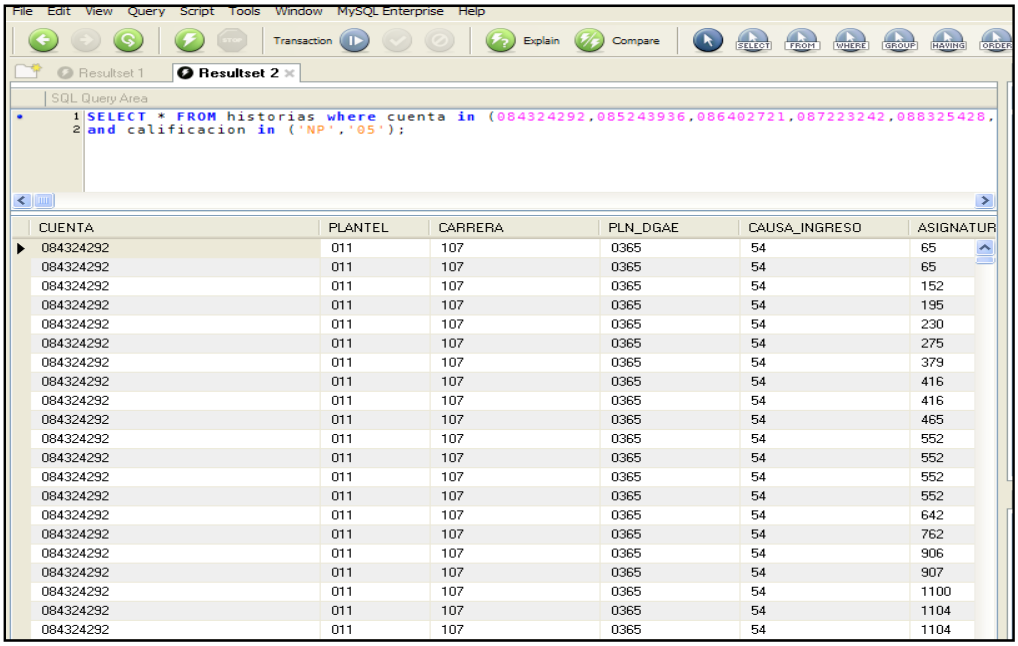


Figura 4.145 Consulta del Query en PHP.

- El Query en la tabla de Historias especifica que sólo se quiere los datos de las materias cuya calificación sea NP o 05, ya que esas calificaciones son las trascendentes o decisivas para que un alumno no siga avanzando o que se desaliente y opte por la deserción. *Vease Fig 4.146*.

SELECT * FROM historias WHERE cuenta IN (08432**,08524****,08640****,08722**** .....) AND calificacion IN ('NP','05');**



CUENTA	PLANTEL	CARRERA	PLN_DGAE	CAUSA_INGRESO	ASIGNATUR
084324292	011	107	0365	54	65
084324292	011	107	0365	54	65
084324292	011	107	0365	54	152
084324292	011	107	0365	54	195
084324292	011	107	0365	54	230
084324292	011	107	0365	54	275
084324292	011	107	0365	54	379
084324292	011	107	0365	54	416
084324292	011	107	0365	54	416
084324292	011	107	0365	54	465
084324292	011	107	0365	54	552
084324292	011	107	0365	54	552
084324292	011	107	0365	54	552
084324292	011	107	0365	54	552
084324292	011	107	0365	54	642
084324292	011	107	0365	54	762
084324292	011	107	0365	54	906
084324292	011	107	0365	54	907
084324292	011	107	0365	54	1100
084324292	011	107	0365	54	1104
084324292	011	107	0365	54	1104

Figura. 4.146 Registros de alumnos de la regla obtenida con calificaciones de 05 y NP.

Para visualizar con mayor certeza las materias trascendentales en este grupo de alumnos que mostraba deserción en casi más de 90% de las incidencias (427), se utilizó SSPS generando el gráfico de Pareto. *Vease Fig. 4.147*:

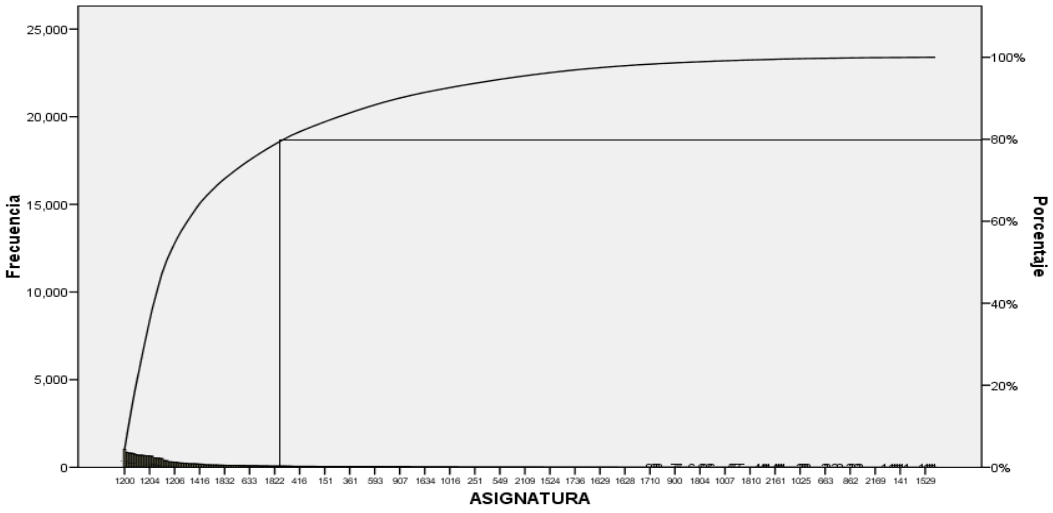


Figura 4.147 Pareto de las materias más reprobadas para este grupo de alumnos.

Como se sabe, el gráfico de pareto utiliza el 80% de las incidencias que influyen para la toma de decisiones. En la siguiente figura 4.148 se muestra la clave de las asignaturas y el nombre de las mismas.

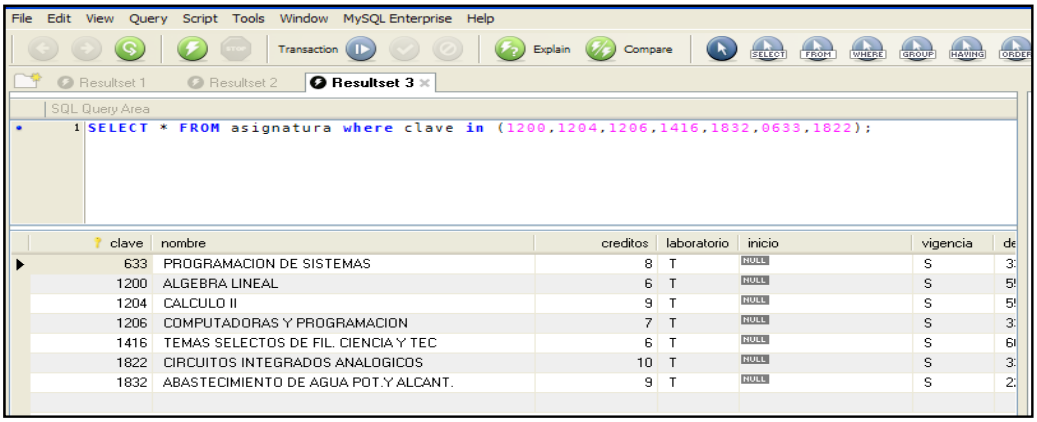


Figura 4.148. Claves de las Materia más reprobadas para estos alumnos.

Analizando los resultados.

Analizando en la tabla *vecesreprobada*, los datos de las personas arrojadas por la regla de deserción en la materia de programación de Sistemas. Véase la *figura 4.149*.

```
SELECT * FROM vecesreprobada WHERE cuenta IN
(08432****,08524****,08640****,08722****.....) AND claveasignatura=0633
AND vecesreprobada>0;
```

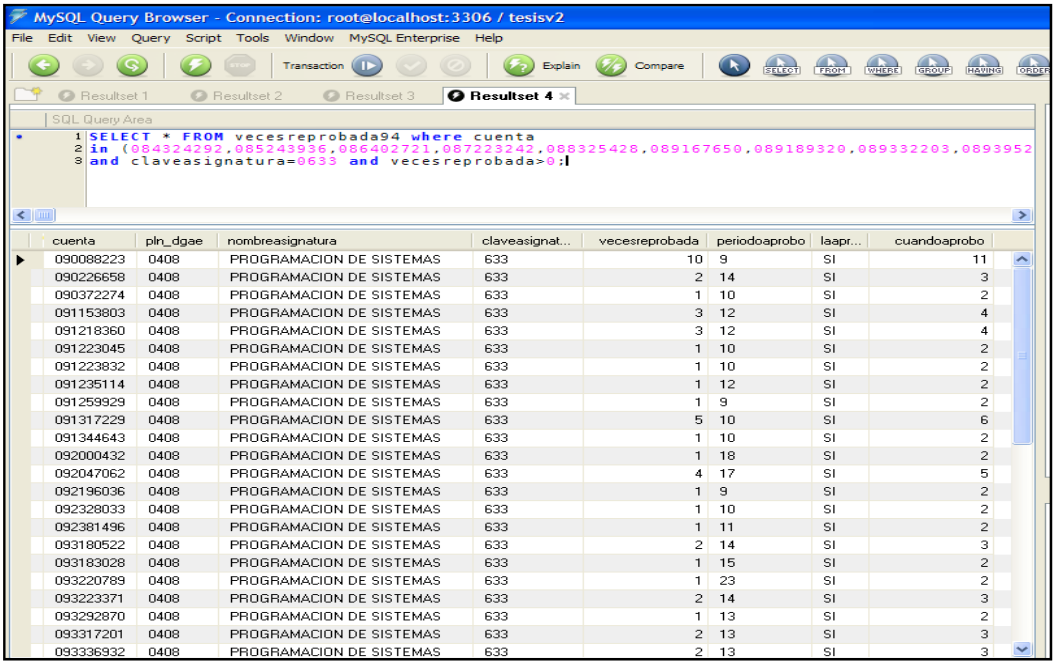
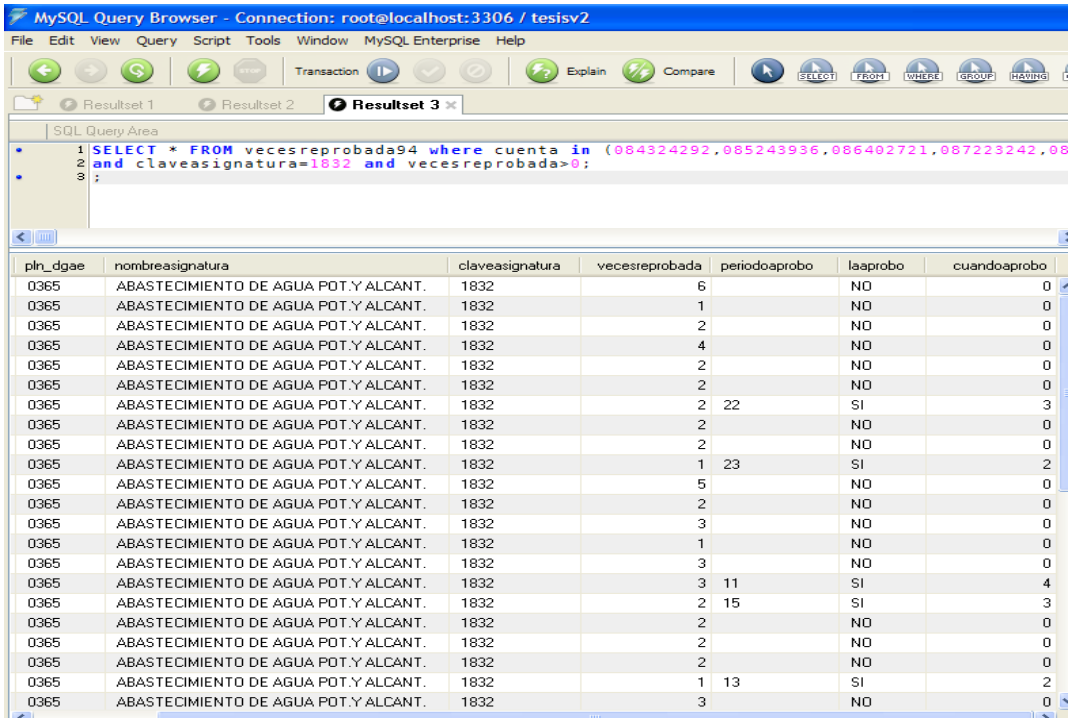


Figura. 4.149. Análisis de la materia de programación de sistemas.

Dicha materia sólo se imparte para los alumnos de Ingeniería en Computación plan 1994, y corresponde al quinto semestre; como se puede apreciar, estos alumnos no la aprobaron en el semestre correspondiente y varios de ellos la reprobaron más de una vez.

Lo trascendente del reprobar esta materia es que para cursar las materias de Sistemas Operativos y Compiladores es necesario aprobarla. Por lo que se considera que este fue un cuello de botella para este grupo de alumnos que sí desertaron.

Ahora se analizará la materia de *Abastecimiento de agua potable y alcantarillado*. Véase la *figura 4.150*.



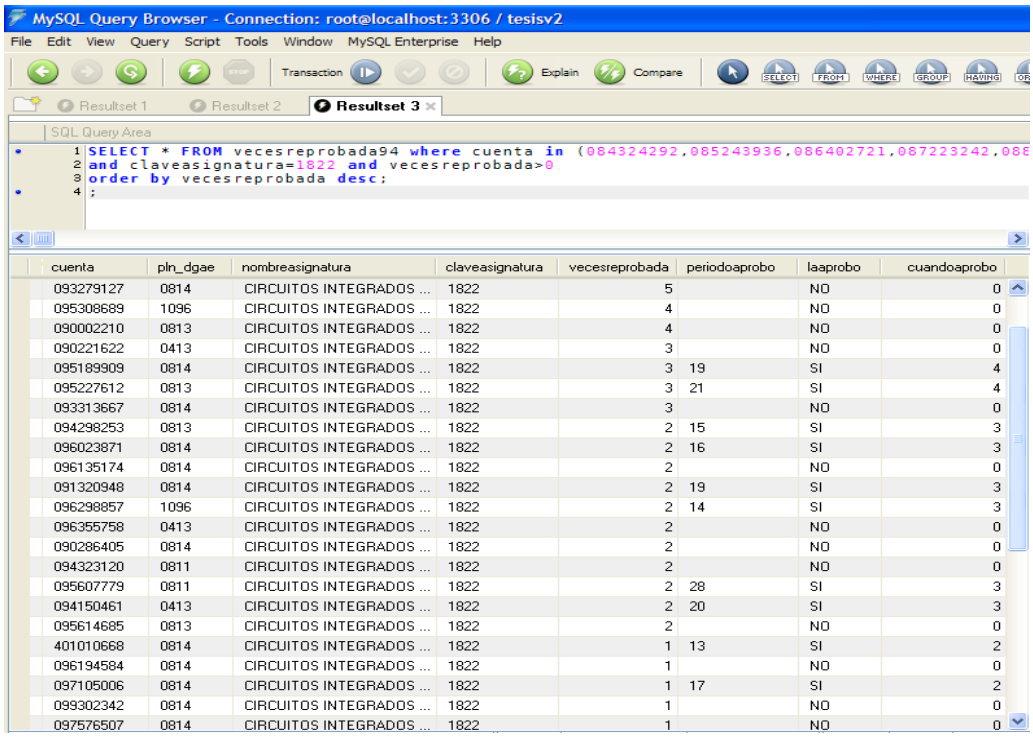
pln_dgae	nombreasignatura	claveasignatura	vecesreprobada	periodoaprobo	laaprobo	cuandoaprobo
0365	ABASTECIMIENTO DE AGUA POT.Y ALCANT.	1832	6		NO	0
0365	ABASTECIMIENTO DE AGUA POT.Y ALCANT.	1832	1		NO	0
0365	ABASTECIMIENTO DE AGUA POT.Y ALCANT.	1832	2		NO	0
0365	ABASTECIMIENTO DE AGUA POT.Y ALCANT.	1832	4		NO	0
0365	ABASTECIMIENTO DE AGUA POT.Y ALCANT.	1832	2		NO	0
0365	ABASTECIMIENTO DE AGUA POT.Y ALCANT.	1832	2		NO	0
0365	ABASTECIMIENTO DE AGUA POT.Y ALCANT.	1832	2	22	SI	3
0365	ABASTECIMIENTO DE AGUA POT.Y ALCANT.	1832	2		NO	0
0365	ABASTECIMIENTO DE AGUA POT.Y ALCANT.	1832	2		NO	0
0365	ABASTECIMIENTO DE AGUA POT.Y ALCANT.	1832	1	23	SI	2
0365	ABASTECIMIENTO DE AGUA POT.Y ALCANT.	1832	5		NO	0
0365	ABASTECIMIENTO DE AGUA POT.Y ALCANT.	1832	2		NO	0
0365	ABASTECIMIENTO DE AGUA POT.Y ALCANT.	1832	3		NO	0
0365	ABASTECIMIENTO DE AGUA POT.Y ALCANT.	1832	1		NO	0
0365	ABASTECIMIENTO DE AGUA POT.Y ALCANT.	1832	3		NO	0
0365	ABASTECIMIENTO DE AGUA POT.Y ALCANT.	1832	3	11	SI	4
0365	ABASTECIMIENTO DE AGUA POT.Y ALCANT.	1832	2	15	SI	3
0365	ABASTECIMIENTO DE AGUA POT.Y ALCANT.	1832	2		NO	0
0365	ABASTECIMIENTO DE AGUA POT.Y ALCANT.	1832	2		NO	0
0365	ABASTECIMIENTO DE AGUA POT.Y ALCANT.	1832	2		NO	0
0365	ABASTECIMIENTO DE AGUA POT.Y ALCANT.	1832	1	13	SI	2
0365	ABASTECIMIENTO DE AGUA POT.Y ALCANT.	1832	3		NO	0

Figurs.4.150. Análisis de la materia de Abastecimiento de agua potable y alcantarillado.

Esta materia es impartida para los alumnos de Ing. Civil. La regla arrojó que era el plan DGAE 0365 el cual pertenece al plan 1994 y se imparte en el octavo semestre. Como puede apreciarse, esta materia es reprobada más de una vez en este grupo de alumnos y la mayoría de ellos no la han aprobado.

A pesar de que esta materia no está seriada con otras materias, lo cual podría implicar un factor importante para clasificarla como un cuello de botella, sí es relevante el hecho de que la mayoría de este grupo de alumnos no pudieron aprobarla.

Ahora se analizará la materia de *Circuitos Integrados Analógicos*. Véase la *figura 4.151*:

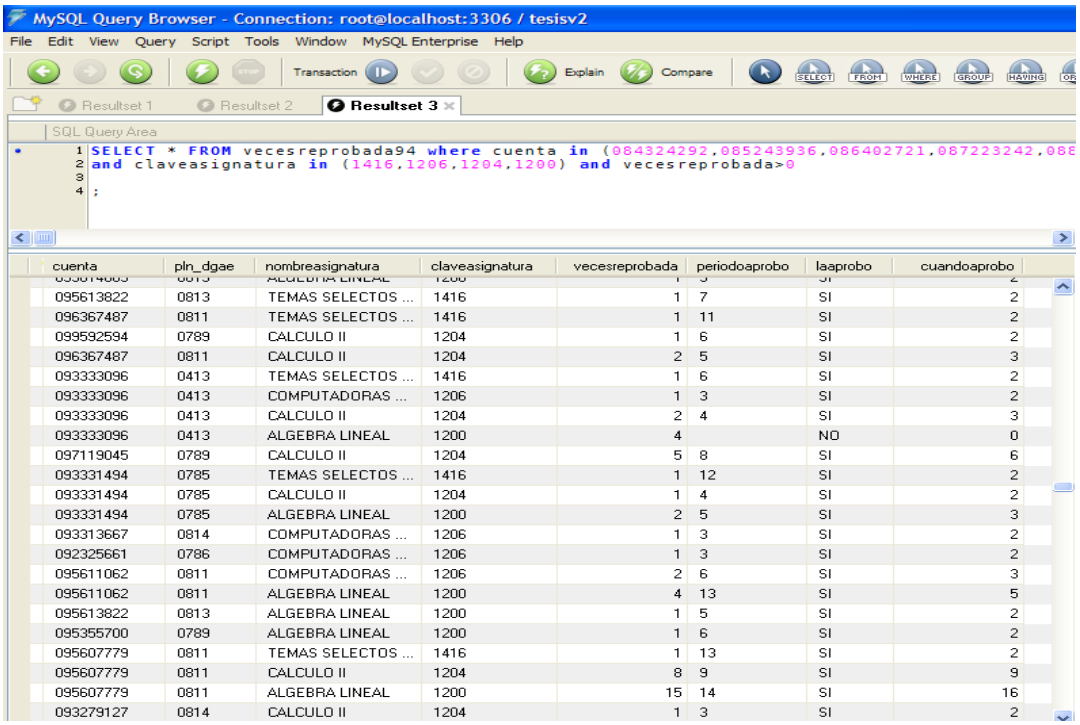


cuenta	pln_dgae	nombreasignatura	claveasignatura	vecesreprobada	periodoaprobo	laaprobo	cuandoaprobo
093279127	0814	CIRCUITOS INTEGRADOS ...	1822	5		NO	0
095308689	1096	CIRCUITOS INTEGRADOS ...	1822	4		NO	0
090002210	0813	CIRCUITOS INTEGRADOS ...	1822	4		NO	0
090221622	0413	CIRCUITOS INTEGRADOS ...	1822	3		NO	0
095189909	0814	CIRCUITOS INTEGRADOS ...	1822	3	19	SI	4
095227612	0813	CIRCUITOS INTEGRADOS ...	1822	3	21	SI	4
093313667	0814	CIRCUITOS INTEGRADOS ...	1822	3		NO	0
094298253	0813	CIRCUITOS INTEGRADOS ...	1822	2	15	SI	3
096023871	0814	CIRCUITOS INTEGRADOS ...	1822	2	16	SI	3
096135174	0814	CIRCUITOS INTEGRADOS ...	1822	2		NO	0
091320948	0814	CIRCUITOS INTEGRADOS ...	1822	2	19	SI	3
096298857	1096	CIRCUITOS INTEGRADOS ...	1822	2	14	SI	3
096355758	0413	CIRCUITOS INTEGRADOS ...	1822	2		NO	0
090286405	0814	CIRCUITOS INTEGRADOS ...	1822	2		NO	0
094323120	0811	CIRCUITOS INTEGRADOS ...	1822	2		NO	0
095607779	0811	CIRCUITOS INTEGRADOS ...	1822	2	28	SI	3
094150461	0413	CIRCUITOS INTEGRADOS ...	1822	2	20	SI	3
095614685	0813	CIRCUITOS INTEGRADOS ...	1822	2		NO	0
401010668	0814	CIRCUITOS INTEGRADOS ...	1822	1	13	SI	2
096194584	0814	CIRCUITOS INTEGRADOS ...	1822	1		NO	0
097105006	0814	CIRCUITOS INTEGRADOS ...	1822	1	17	SI	2
099302342	0814	CIRCUITOS INTEGRADOS ...	1822	1		NO	0
097576507	0814	CIRCUITOS INTEGRADOS ...	1822	1		NO	0

Figura 4.151. Análisis de la materia de Circuitos integrados analógicos.

Esta materia se imparte en el Octavo Semestre para los alumnos de Ingeniería Eléctrica Electrónica. Cabe resaltar que esta materia no está seriada con otras. Sin embargo, un 60% de estos alumnos tampoco la ha podido aprobar y los que la han llegado a aprobar lo han hecho en períodos o semestres muy por encima de los 10 semestres cursados establecidos para dicha carrera.

Ahora se analizarán las materias restantes en conjunto ya que estas son cursadas por todas las carreras y se encuentran impartidas en el cuello de botella más conocido por todos los Alumnos y Profesores, denominado *Ciencias Básicas*. Véase la *figura 4.152*.



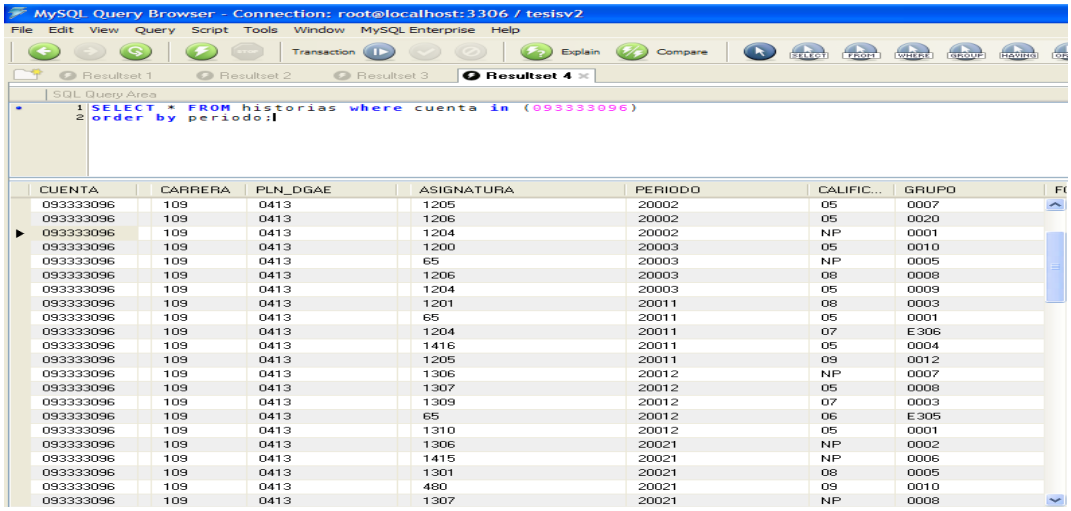
cuenta	pln_dgae	nombreasignatura	claveasignatura	vecesreprobada	periodoaprobo	laaprobo	cuandoaprobo
095613822	0813	TEMAS SELECTOS ...	1416	1	7	SI	2
096367487	0811	TEMAS SELECTOS ...	1416	1	11	SI	2
099592594	0789	CALCULO II	1204	1	6	SI	2
096367487	0811	CALCULO II	1204	2	5	SI	3
093333096	0413	TEMAS SELECTOS ...	1416	1	6	SI	2
093333096	0413	COMPUTADORAS ...	1206	1	3	SI	2
093333096	0413	CALCULO II	1204	2	4	SI	3
093333096	0413	ALGEBRA LINEAL	1200	4		NO	0
097119045	0789	CALCULO II	1204	5	8	SI	6
093331494	0785	TEMAS SELECTOS ...	1416	1	12	SI	2
093331494	0785	CALCULO II	1204	1	4	SI	2
093331494	0785	ALGEBRA LINEAL	1200	2	5	SI	3
093313667	0814	COMPUTADORAS ...	1206	1	3	SI	2
092325661	0786	COMPUTADORAS ...	1206	1	3	SI	2
095611062	0811	COMPUTADORAS ...	1206	2	6	SI	3
095611062	0811	ALGEBRA LINEAL	1200	4	13	SI	5
095613822	0813	ALGEBRA LINEAL	1200	1	5	SI	2
095355700	0789	ALGEBRA LINEAL	1200	1	6	SI	2
095607779	0811	TEMAS SELECTOS ...	1416	1	13	SI	2
095607779	0811	CALCULO II	1204	8	9	SI	9
095607779	0811	ALGEBRA LINEAL	1200	15	14	SI	16
093279127	0814	CALCULO II	1204	1	3	SI	2

Figura 4.152. Análisis de las materias de Ciencias Básicas.

Lo relevante en estas materias no es el hecho que éstas sean reprobadas una sola vez, ya que este dato es bien conocido; lo trascendente en nuestro punto de vista es que materias como *Álgebra Lineal*, *Cálculo II* y *Computadoras y programación*, las cuales son materias que pertenecen al segundo semestre, estén siendo aprobadas en la mayoría de los casos en semestres o períodos mayores al cuarto.

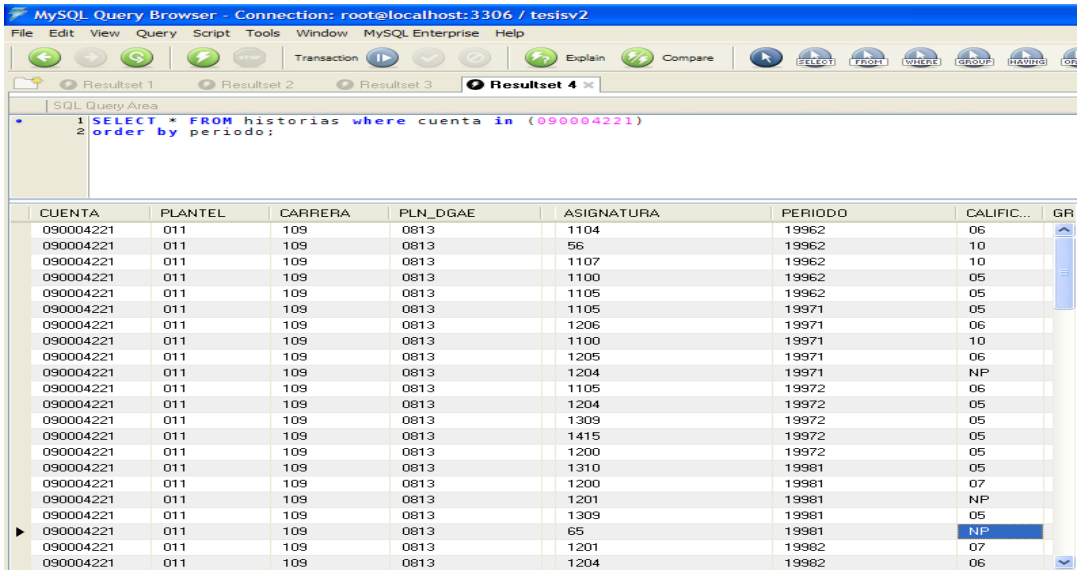
El mismo patrón se presenta en la materia de Temas Selectos de Filosofía, materia que debe ser cursada en los semestres cuarto, quinto y sexto; está siendo aprobada en semestres superiores a éstos.

Analizando las características de algunos de estos alumnos, se encontró que desde el primer semestre tuvieron dos o más materias reprobadas y que en sus siguientes semestres continuaban con dicho patrón recalcando que nunca tenían más de cinco materias por cada semestre, lo cual ocasionó que su atraso fuera creciendo más. Véase las *figuras 4.153* y *4.154*.



CUENTA	CARRERA	PLN_DGAE	ASIGNATURA	PERIODO	CALIFIC...	GRUPO	FO
093333096	109	0413	1205	20002	05	0007	
093333096	109	0413	1206	20002	05	0020	
093333096	109	0413	1204	20002	NP	0001	
093333096	109	0413	1200	20003	05	0010	
093333096	109	0413	65	20003	NP	0005	
093333096	109	0413	1206	20003	08	0008	
093333096	109	0413	1204	20003	05	0009	
093333096	109	0413	1201	20011	08	0003	
093333096	109	0413	65	20011	05	0001	
093333096	109	0413	1204	20011	07	E306	
093333096	109	0413	1416	20011	05	0004	
093333096	109	0413	1205	20011	09	0012	
093333096	109	0413	1306	20012	NP	0007	
093333096	109	0413	1307	20012	05	0008	
093333096	109	0413	1309	20012	07	0003	
093333096	109	0413	65	20012	06	E305	
093333096	109	0413	1310	20012	05	0001	
093333096	109	0413	1306	20021	NP	0002	
093333096	109	0413	1415	20021	NP	0006	
093333096	109	0413	1301	20021	08	0005	
093333096	109	0413	480	20021	09	0010	
093333096	109	0413	1307	20021	NP	0008	

Figura 4.153. Ejemplo de historial con un considerable atraso desde los primeros semestres.



CUENTA	PLANTEL	CARRERA	PLN_DGAE	ASIGNATURA	PERIODO	CALIFIC...	GR
090004221	011	109	0813	1104	19962	06	
090004221	011	109	0813	56	19962	10	
090004221	011	109	0813	1107	19962	10	
090004221	011	109	0813	1100	19962	05	
090004221	011	109	0813	1105	19962	05	
090004221	011	109	0813	1105	19971	05	
090004221	011	109	0813	1206	19971	06	
090004221	011	109	0813	1100	19971	10	
090004221	011	109	0813	1205	19971	06	
090004221	011	109	0813	1204	19971	NP	
090004221	011	109	0813	1105	19972	06	
090004221	011	109	0813	1204	19972	05	
090004221	011	109	0813	1309	19972	05	
090004221	011	109	0813	1415	19972	05	
090004221	011	109	0813	1200	19972	05	
090004221	011	109	0813	1310	19981	05	
090004221	011	109	0813	1200	19981	07	
090004221	011	109	0813	1201	19981	NP	
090004221	011	109	0813	1309	19981	05	
090004221	011	109	0813	65	19981	NP	
090004221	011	109	0813	1201	19982	07	
090004221	011	109	0813	1204	19982	06	

Figura 4.154. Ejemplo de historial con un considerable atraso desde los primeros semestres.

Utilizando Otro árbol de decisión *Random Forest,* Árbol ó Modelo 12.

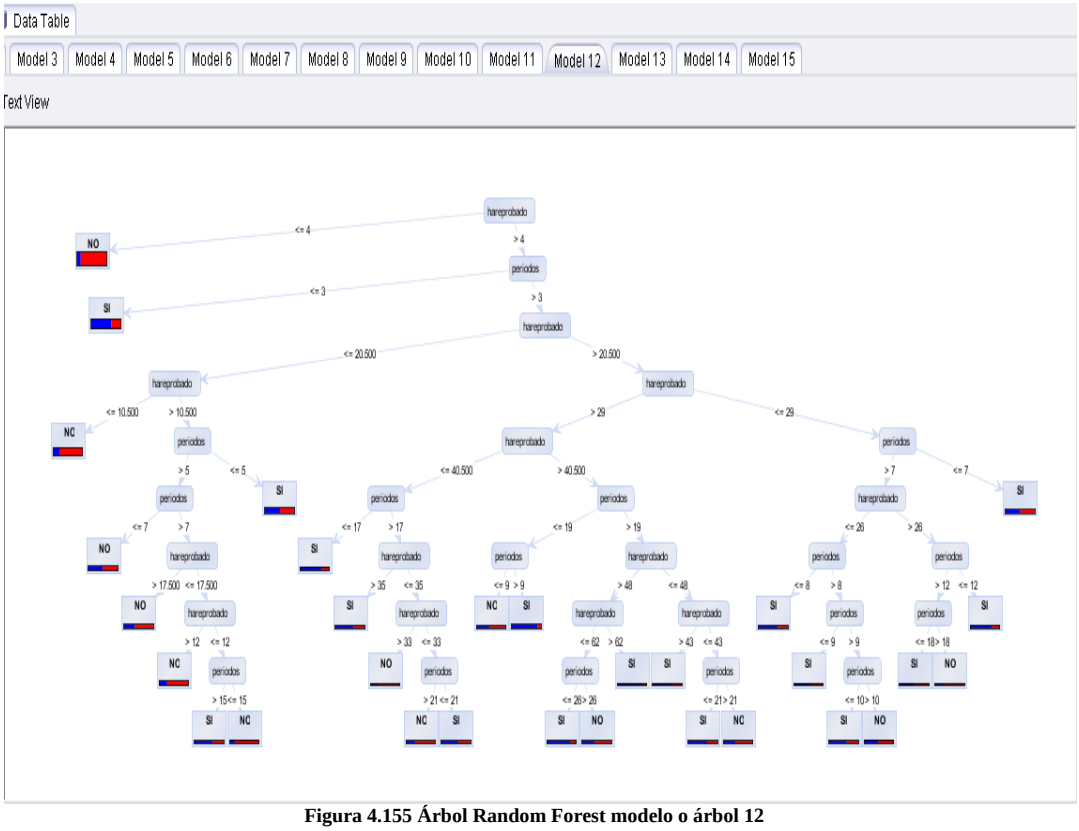


Figura 4.155 Árbol Random Forest modelo o árbol 12

Tal y como se hizo con el modelo o árbol 7, se pide que se muestre el árbol en forma de texto. A continuación se muestra parte del mismo. Para ver el árbol de texto completo (*Tree 2*), véase el anexo al final de este documento:

Tree 2

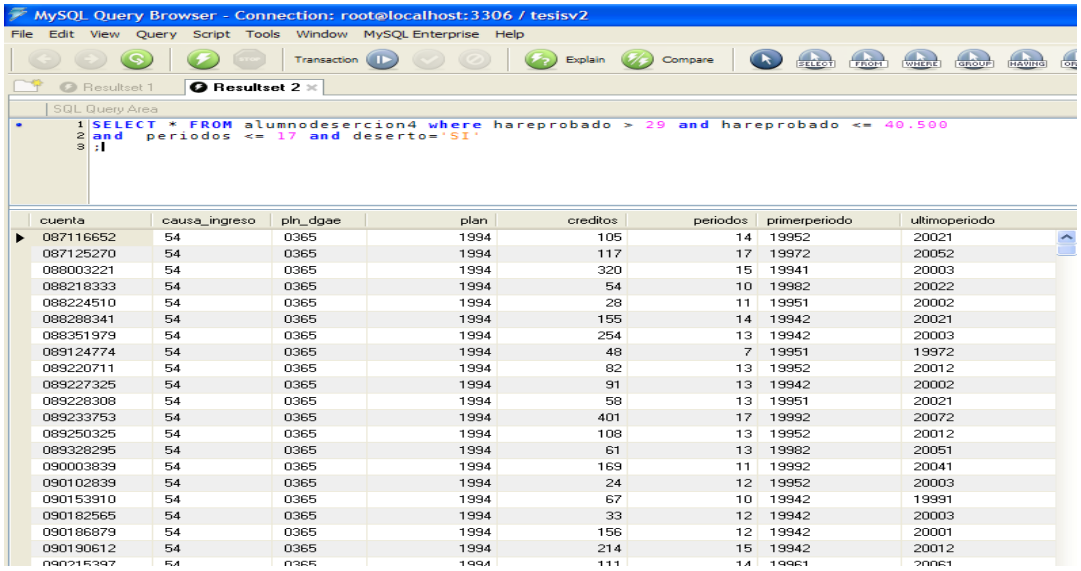
```
hareprobado <= 4: NO {SI=720, NO=5254}
hareprobado > 4
…
…
…

|   |   |   |   |   periodos > 7
|   |   |   |   |   |   hareprobado <= 17.500
|   |   |   |   |   |   |   hareprobado <= 12
|   |   |   |   |   |   |   |   periodos <= 15: NO {SI=91, NO=415}
|   |   |   |   |   |   |   |   periodos > 15: SI {SI=13, NO=8}
|   |   |   |   |   |   |   hareprobado > 12: NO {SI=350, NO=893}
…
…
…
|   |   |   hareprobado > 29
|   |   |   |   hareprobado <= 40.500
|   |   |   |   |   periodos <= 17: SI {SI=656, NO=230}
|   |   |   |   …
…
```

```
…
|   |   |   |   hareprobado > 40.500
|   |   |   |   |   periodos <= 19
|   |   |   |   |   |   periodos <= 9: NO {SI=4, NO=5}
|   |   |   |   |   |   periodos > 9: SI {SI=545, NO=79}
|   |   |   |   |   …
…
…
```

Ahora se analizará la primera regla de este árbol. Véase la *figura 4.156*.

SELECT * FROM alumnodesercion4 WHERE hareprobado > 29 AND hareprobado <= 40.500 AND periodos <= 17 AND deserto='SI';



cuenta	causa_ingreso	pln_dgae	plan	creditos	periodos	primerperiodo	ultimoperiodo
087116652	54	0365	1994	105	14	19952	20021
087125270	54	0365	1994	117	17	19972	20052
088003221	54	0365	1994	320	15	19941	20003
088218333	54	0365	1994	54	10	19982	20022
088224510	54	0365	1994	28	11	19951	20002
088288341	54	0365	1994	155	14	19942	20021
088351979	54	0365	1994	254	13	19942	20003
089124774	54	0365	1994	48	7	19951	19972
089220711	54	0365	1994	82	13	19952	20012
089227325	54	0365	1994	91	13	19942	20002
089228308	54	0365	1994	58	13	19951	20021
089233753	54	0365	1994	401	17	19992	20072
089250325	54	0365	1994	108	13	19952	20012
089328295	54	0365	1994	61	13	19982	20051
090003839	54	0365	1994	169	11	19992	20041
090102839	54	0365	1994	24	12	19952	20003
090153910	54	0365	1994	67	10	19942	19991
090182565	54	0365	1994	33	12	19942	20003
090186879	54	0365	1994	156	12	19942	20001
090190612	54	0365	1994	214	15	19942	20012
090215397	54	0365	1994	111	14	19961	20061

Figura 4.156. Visualización de los datos obtenidos por la regla.

Realizando el mismo procedimiento de utilizar los datos de éstos alumnos con calificaciones de 05 y NP se obtuvo la gráfica de *Pareto* para visualizar las materias que tienen más relevancia en este grupo de alumnos donde la regla arrojó que sí desertaban. *Véase la figura 4.157*.

Las que entran dentro del 80% de los casos relevantes en la gráfica de Pareto, vuelven a ser la asignatura 1416, '*Temas selectos de filosofía, ciencia y tecnología*' y la 1100, *Álgebra*. Véase la *figura 4.158*.

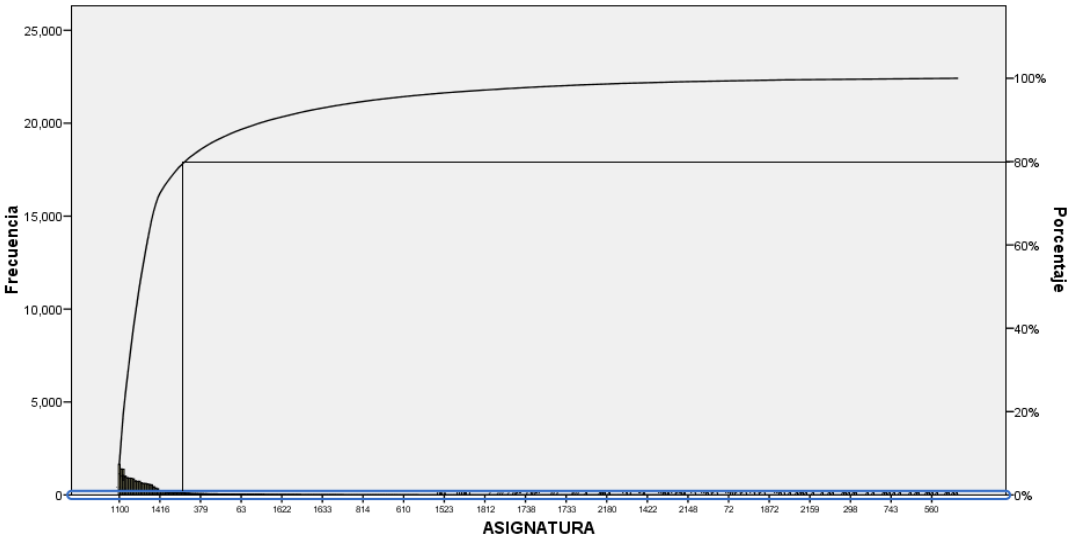
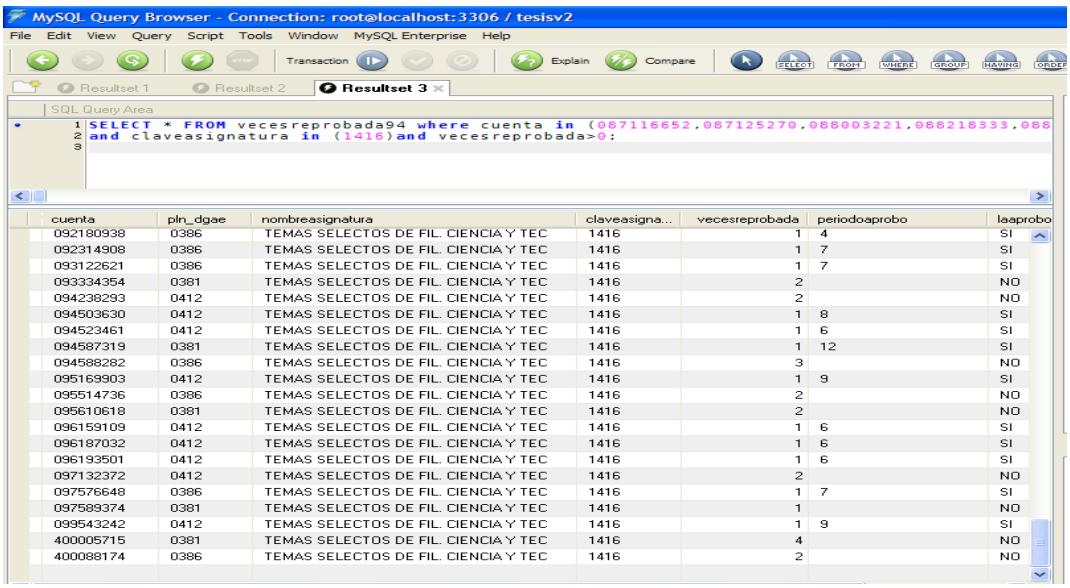


Figura 4.157. Diagrama de Pareto de las materias más reprobadas para este grupo de alumnos.

Analizando la materia de Temas selectos de filosofía ciencia y tecnología:

Se vuelve a repetir el tipo de patrón que en el grupo de alumnos analizado anteriormente. Aunque la reprueban una sola vez, el período o semestre aprobado está por encima del período o semestre donde tuvieron que haberla acreditado.



cuenta	pln_dgae	nombreasignatura	claveasigna...	vecesreprobada	periodoaprobo	laaprobo
092180938	0386	TEMAS SELECTOS DE FIL. CIENCIA Y TEC	1416	1	4	SI
092314908	0386	TEMAS SELECTOS DE FIL. CIENCIA Y TEC	1416	1	7	SI
093122621	0386	TEMAS SELECTOS DE FIL. CIENCIA Y TEC	1416	1	7	SI
093334354	0381	TEMAS SELECTOS DE FIL. CIENCIA Y TEC	1416	2		NO
094238293	0412	TEMAS SELECTOS DE FIL. CIENCIA Y TEC	1416	2		NO
094503630	0412	TEMAS SELECTOS DE FIL. CIENCIA Y TEC	1416	1	8	SI
094523461	0412	TEMAS SELECTOS DE FIL. CIENCIA Y TEC	1416	1	6	SI
094587319	0381	TEMAS SELECTOS DE FIL. CIENCIA Y TEC	1416	1	12	SI
094588282	0386	TEMAS SELECTOS DE FIL. CIENCIA Y TEC	1416	3		NO
095169903	0412	TEMAS SELECTOS DE FIL. CIENCIA Y TEC	1416	1	9	SI
095514736	0386	TEMAS SELECTOS DE FIL. CIENCIA Y TEC	1416	2		NO
095610618	0381	TEMAS SELECTOS DE FIL. CIENCIA Y TEC	1416	2		NO
096159109	0412	TEMAS SELECTOS DE FIL. CIENCIA Y TEC	1416	1	6	SI
096187032	0412	TEMAS SELECTOS DE FIL. CIENCIA Y TEC	1416	1	6	SI
096193501	0412	TEMAS SELECTOS DE FIL. CIENCIA Y TEC	1416	1	6	SI
097132372	0412	TEMAS SELECTOS DE FIL. CIENCIA Y TEC	1416	2		NO
097576648	0386	TEMAS SELECTOS DE FIL. CIENCIA Y TEC	1416	1	7	SI
097589374	0381	TEMAS SELECTOS DE FIL. CIENCIA Y TEC	1416	1		NO
099543242	0412	TEMAS SELECTOS DE FIL. CIENCIA Y TEC	1416	1	9	SI
400005715	0381	TEMAS SELECTOS DE FIL. CIENCIA Y TEC	1416	4		NO
400088174	0386	TEMAS SELECTOS DE FIL. CIENCIA Y TEC	1416	2		NO

Figura 4.158. Análisis de la materia Temas selectos de filosofía, ciencia y tecnología.

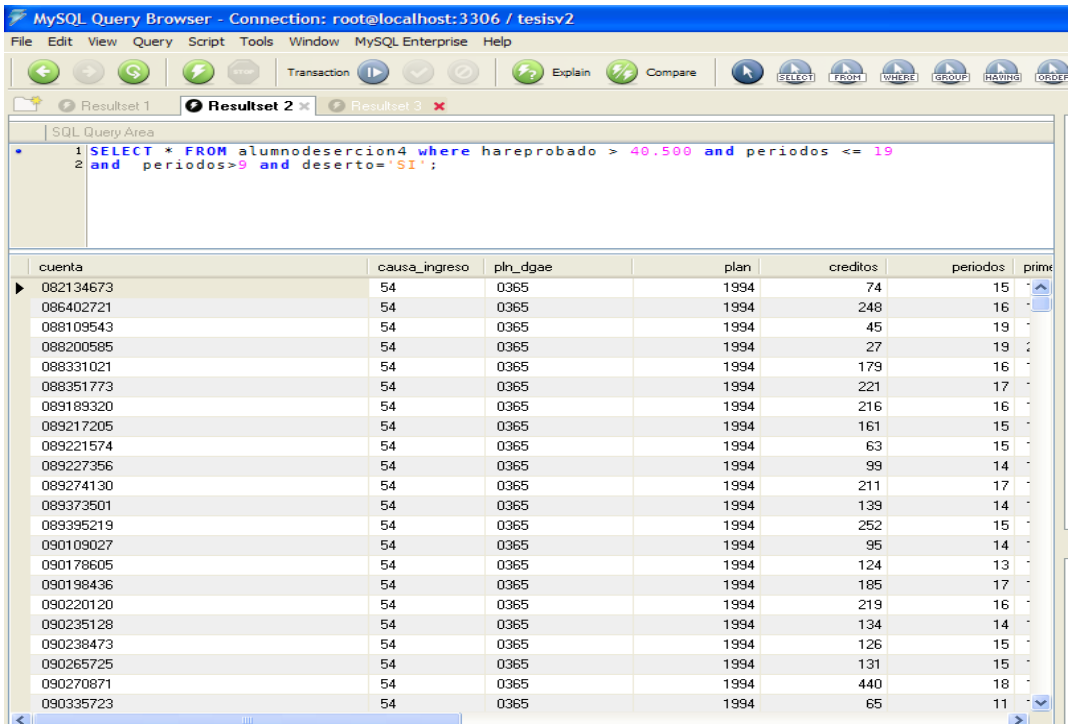
Asimismo, es interesante y trascendente el hecho de que dicha materia pertenece al grupo de materias del área de humanidades. Para la mayoría de los estudiantes, se hace

relativamente sencilla acreditarla, ya que en muchos de los grupos es bien sabido que lo único que hay que hacer es cumplir con asistencia y tareas relativamente sencillas, comparadas con la carga de trabajo de las demás materias dentro de nuestra área de estudio.

La materia 1100 – Álgebra, es una materia conocida por su alto índice de reprobación y, por ello, no es considerado el hallazgo de mayor trascendencia aunque siendo una materia con seriación sí influye en el atraso curricular de los alumnos.

Estudiando la siguiente regla. Véase la *figura 4.159*.

SELECT * FROM alumnodesercion4 WHERE hareprobado > 40.500 AND periodos <= 19 AND periodos>9 AND deserto='SI';



cuenta	causa_ingreso	pln_dgae	plan	creditos	periodos	prime
082134673	54	0365	1994	74	15	
086402721	54	0365	1994	248	16	
088109543	54	0365	1994	45	19	
088200585	54	0365	1994	27	19	
088331021	54	0365	1994	179	16	
088351773	54	0365	1994	221	17	
089189320	54	0365	1994	216	16	
089217205	54	0365	1994	161	15	
089221574	54	0365	1994	63	15	
089227356	54	0365	1994	99	14	
089274130	54	0365	1994	211	17	
089373501	54	0365	1994	139	14	
089395219	54	0365	1994	252	15	
090109027	54	0365	1994	95	14	
090178605	54	0365	1994	124	13	
090198436	54	0365	1994	185	17	
090220120	54	0365	1994	219	16	
090235128	54	0365	1994	134	14	
090238473	54	0365	1994	126	15	
090265725	54	0365	1994	131	15	
090270871	54	0365	1994	440	18	
090335723	54	0365	1994	65	11	

Figura 4.159. Visualización de los datos obtenidos para la siguiente regla.

Se obtiene la gráfica de *Pareto* para encontrar aquellas materias relevantes para la deserción de este nuevo grupo de alumnos. Véase la *figura 4.160*.

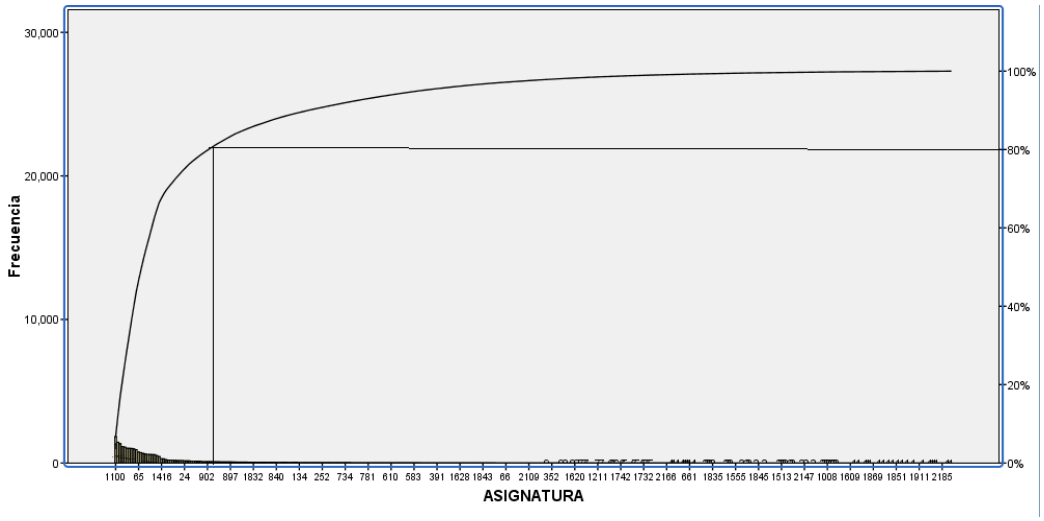


Figura 4.160. Diagrama de Pareto de las materias más reprobadas para este grupo.

Las materias que entran dentro del área del 80% de la gráfica de *Pareto* son: 1100 Álgebra, 65 Estática, 1416 Temas selectos de filosofía ciencia y tecnología, 24 Análisis de circuitos eléctricos y 902 Teoría Electromagnética.

Analizando la materia de Estática se tiene. Véase la *figura 4.161*.

MySQL Query Browser - Connection: root@localhost:3306 / tesisv2

File Edit View Query Script Tools Window MySQL Enterprise Help

Transaction Explain Compare SELECT FROM WHERE GROUP HAVING ORDER

Resultset 1 Resultset 2 Resultset 3 **Resultset 4**

SQL Query Area

```
1 SELECT * FROM vecesreprobada94 where cuenta in (082134673,086402721,088109543,088200585,088
2 and claveasignatura in (65)and vecesreprobada>0;
3
```

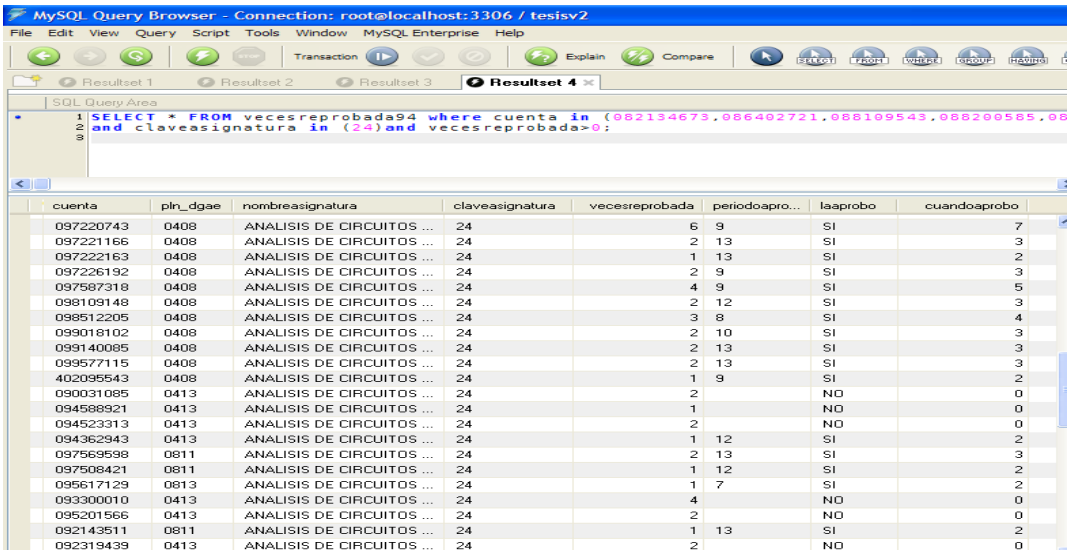
cuenta	pln_dgae	nombreasignatura	claveasignatura	vecesreprobada	periodoaprobo	laaprobo	cuandoaprobo
095610821	0365	ESTATICA	65	5		NO	0
095612313	0365	ESTATICA	65	2	6	SI	3
096162330	0365	ESTATICA	65	1	7	SI	2
096519253	0365	ESTATICA	65	1	4	SI	2
096553143	0365	ESTATICA	65	8	13	SI	9
096555752	0365	ESTATICA	65	1	4	SI	2
097169064	0365	ESTATICA	65	1	4	SI	2
097322940	0365	ESTATICA	65	1	9	SI	2
097569426	0365	ESTATICA	65	3	11	SI	4
097575524	0365	ESTATICA	65	1	3	SI	2
098163474	0365	ESTATICA	65	3	7	SI	4
098518654	0365	ESTATICA	65	3	7	SI	4
098525993	0365	ESTATICA	65	2	6	SI	3
099569628	0365	ESTATICA	65	2	8	SI	3
094589461	0413	ESTATICA	65	1	4	SI	2
094589076	0413	ESTATICA	65	3		NO	0
094588921	0413	ESTATICA	65	2	12	SI	3
094588718	0413	ESTATICA	65	8	11	SI	9
094393262	0814	ESTATICA	65	2	5	SI	3
094362943	0413	ESTATICA	65	1	4	SI	2
094291797	0413	ESTATICA	65	1		NO	0
094274871	0814	ESTATICA	65	1	4	SI	2

Figura 4.161. Análisis de la materia Estática.

Estática es una materia impartida en Ciencias Básicas para todas las carreras y ésta, en algunos casos, está seriada con dos o más materias. Lo trascendente y relevante es que dicha materia tiene el mayor índice de reprobación en la carrera de Ingeniería Civil y se sabe que dicha materia para esta carrera es importante.

Por otra parte, se vuelve a repetir el patrón de conducta con otras materias de Ciencias Básicas: es reprobada varias veces y, a pesar de que es impartida en el segundo semestre, se aprueba en semestres superiores al cuarto.

Analizando los datos de Análisis de Circuitos Eléctricos y Teoría Electromagnética se tiene (*Figura 4.162*):



cuenta	pln_dgae	nombreasignatura	claveasignatura	vecesreprobada	periodoapro...	laaprobo	cuandoaprobo
097220743	0408	ANALISIS DE CIRCUITOS ...	24	6	9	SI	7
097221166	0408	ANALISIS DE CIRCUITOS ...	24	2	13	SI	3
097222163	0408	ANALISIS DE CIRCUITOS ...	24	1	13	SI	2
097226192	0408	ANALISIS DE CIRCUITOS ...	24	2	9	SI	3
097587318	0408	ANALISIS DE CIRCUITOS ...	24	4	9	SI	5
098109148	0408	ANALISIS DE CIRCUITOS ...	24	2	12	SI	3
098512205	0408	ANALISIS DE CIRCUITOS ...	24	3	8	SI	4
099018102	0408	ANALISIS DE CIRCUITOS ...	24	2	10	SI	3
099140085	0408	ANALISIS DE CIRCUITOS ...	24	2	13	SI	3
099577115	0408	ANALISIS DE CIRCUITOS ...	24	2	13	SI	3
402095543	0408	ANALISIS DE CIRCUITOS ...	24	1	9	SI	2
090031085	0413	ANALISIS DE CIRCUITOS ...	24	2		NO	0
094588921	0413	ANALISIS DE CIRCUITOS ...	24	1		NO	0
094523313	0413	ANALISIS DE CIRCUITOS ...	24	2		NO	0
094362943	0413	ANALISIS DE CIRCUITOS ...	24	1	12	SI	2
097569598	0811	ANALISIS DE CIRCUITOS ...	24	2	13	SI	3
097508421	0811	ANALISIS DE CIRCUITOS ...	24	1	12	SI	2
095617129	0813	ANALISIS DE CIRCUITOS ...	24	1	7	SI	2
093300010	0413	ANALISIS DE CIRCUITOS ...	24	4		NO	0
095201566	0413	ANALISIS DE CIRCUITOS ...	24	2		NO	0
092143511	0811	ANALISIS DE CIRCUITOS ...	24	1	13	SI	2
092319439	0413	ANALISIS DE CIRCUITOS ...	24	2		NO	0

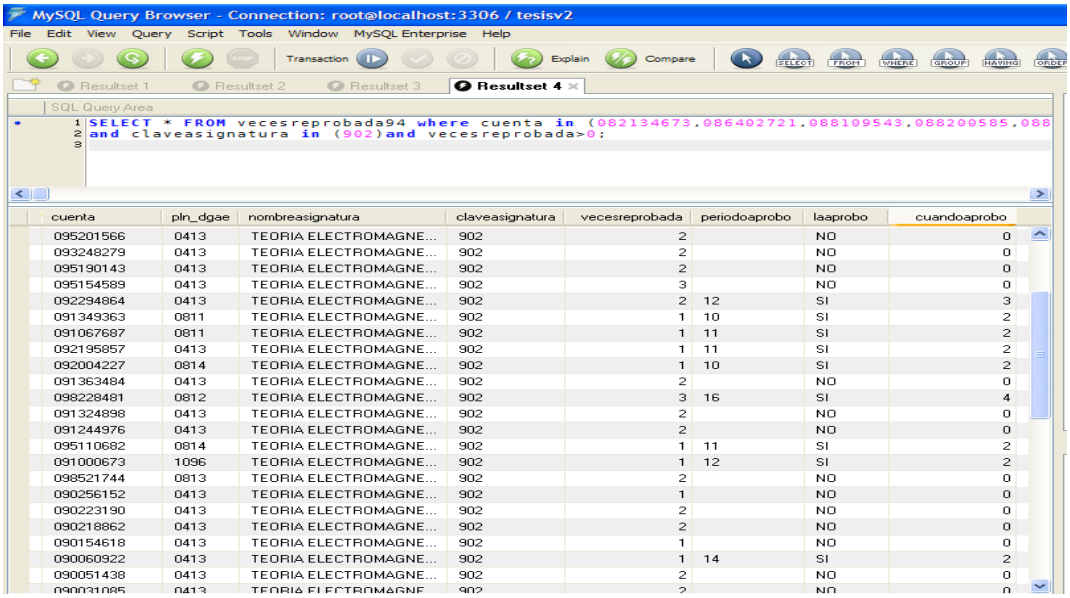
Figura 4.162. Análisis de la materia de Análisis de circuitos eléctricos.

La materia de Análisis de Circuitos Eléctricos sólo es impartida para los alumnos que pertenecen a la división de la DIE en el sexto semestre. En la mayoría de los casos, estos alumnos llegaron a aprobarla en periodos posteriores al especificado en el plan de estudios. Por otra parte, cabe resaltar que la mayoría de ellos pertenecen a la carrera de Ingeniería en Computación (*figura 4.163*):

Para el caso de la carrera Ingeniería Eléctrica Electrónica se imparte en el sexto semestre. No es una materia seriada, pero como se puede observar en la *figura 4.163*, la

mayoría de estos alumnos no pudieron acreditarla y la mayoría de estos la cursó dos veces.

Las materias de Álgebra y Temas selectos de filosofía ciencia y tecnología, ya no serán estudiadas en este grupo de alumnos, ya que ya han sido estudiadas con anterioridad.



cuenta	pln_dgae	nombreasignatura	claveasignatura	vecesreprobada	periodoaprobo	laaprobo	cuandoaprobo
095201566	0413	TEORIA ELECTROMAGNE...	902	2		NO	0
093248279	0413	TEORIA ELECTROMAGNE...	902	2		NO	0
095190143	0413	TEORIA ELECTROMAGNE...	902	2		NO	0
095154589	0413	TEORIA ELECTROMAGNE...	902	3		NO	0
092294864	0413	TEORIA ELECTROMAGNE...	902	2	12	SI	3
091349363	0811	TEORIA ELECTROMAGNE...	902	1	10	SI	2
091067687	0811	TEORIA ELECTROMAGNE...	902	1	11	SI	2
092195857	0413	TEORIA ELECTROMAGNE...	902	1	11	SI	2
092004227	0814	TEORIA ELECTROMAGNE...	902	1	10	SI	2
091363484	0413	TEORIA ELECTROMAGNE...	902	2		NO	0
098228481	0812	TEORIA ELECTROMAGNE...	902	3	16	SI	4
091324898	0413	TEORIA ELECTROMAGNE...	902	2		NO	0
091244976	0413	TEORIA ELECTROMAGNE...	902	2		NO	0
095110682	0814	TEORIA ELECTROMAGNE...	902	1	11	SI	2
091000673	1096	TEORIA ELECTROMAGNE...	902	1	12	SI	2
098521744	0813	TEORIA ELECTROMAGNE...	902	2		NO	0
090256152	0413	TEORIA ELECTROMAGNE...	902	1		NO	0
090223190	0413	TEORIA ELECTROMAGNE...	902	2		NO	0
090218862	0413	TEORIA ELECTROMAGNE...	902	2		NO	0
090154618	0413	TEORIA ELECTROMAGNE...	902	1		NO	0
090060922	0413	TEORIA ELECTROMAGNE...	902	1	14	SI	2
090051438	0413	TEORIA ELECTROMAGNE...	902	2		NO	0
090031085	0413	TEORIA ELECTROMAGNE	902	2		NO	0

Figura 4.163. Análisis de la materia Teoría electromagnética.

Árbol de decisión.

El árbol de decisión es de los más utilizados en el estudio de la Minería de Datos; lo anterior debido a su facilidad de uso. En la *figura 4.164* se mostrará la construcción del Árbol de Decisión y en la *figura 4.165 se mostrará* el modelo del mismo.

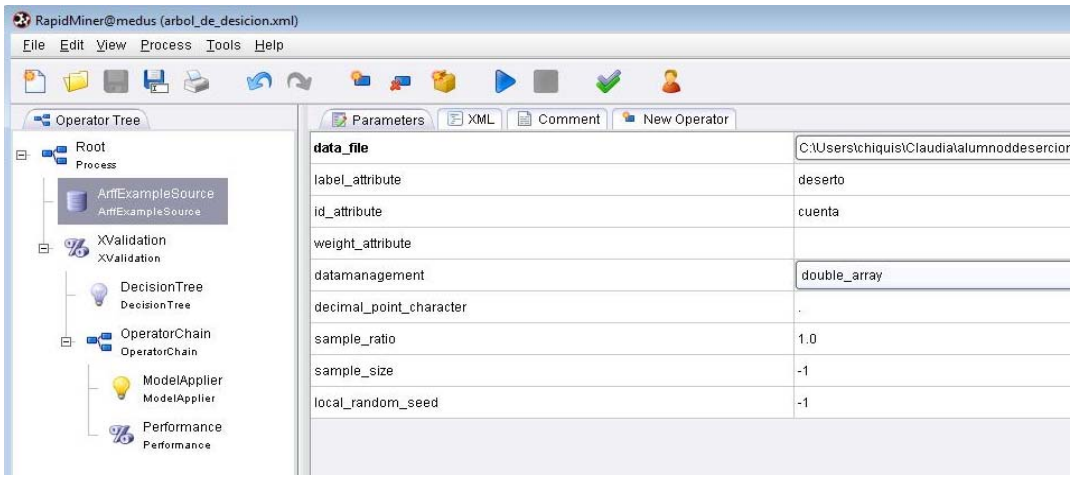


Figura 4.164. Construcción del árbol de decisión.

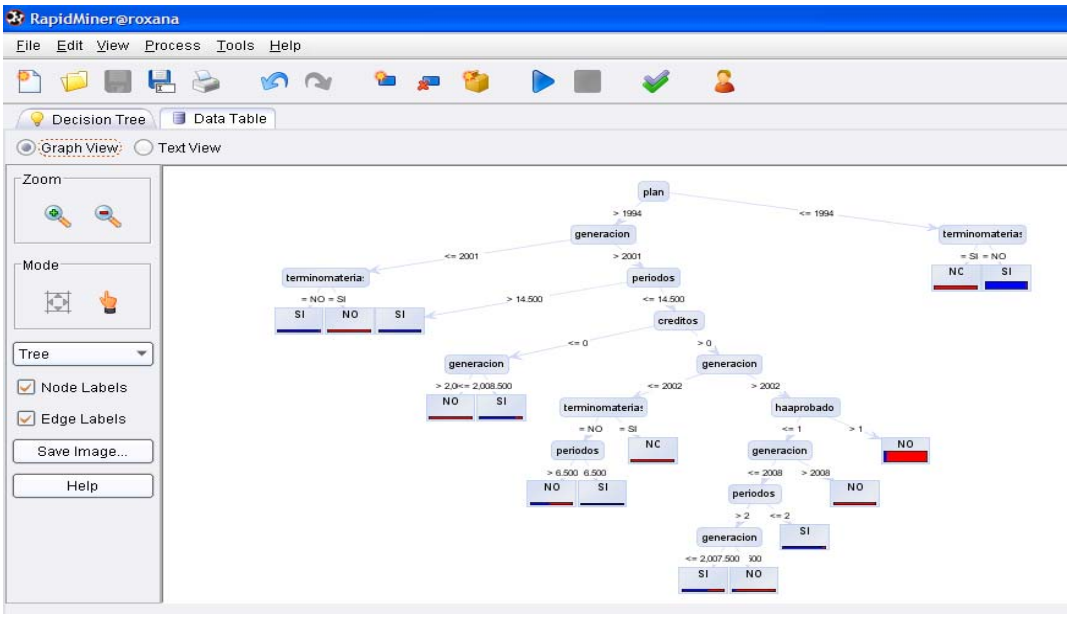


Figura 4.165. Árbol de decisión en *Rapidminer*.

Agregando el operador *Tree2RuleConverter*, se puede convertir el árbol en texto. Con dicho cambio, se pueden visualizar mejor las reglas. Este operador no fue agregado al Árbol *Random Forest* (Bosque Aleatorio), ya que éste al generar varios árboles, no es compatible con dicho operador (*figura 4.166*).

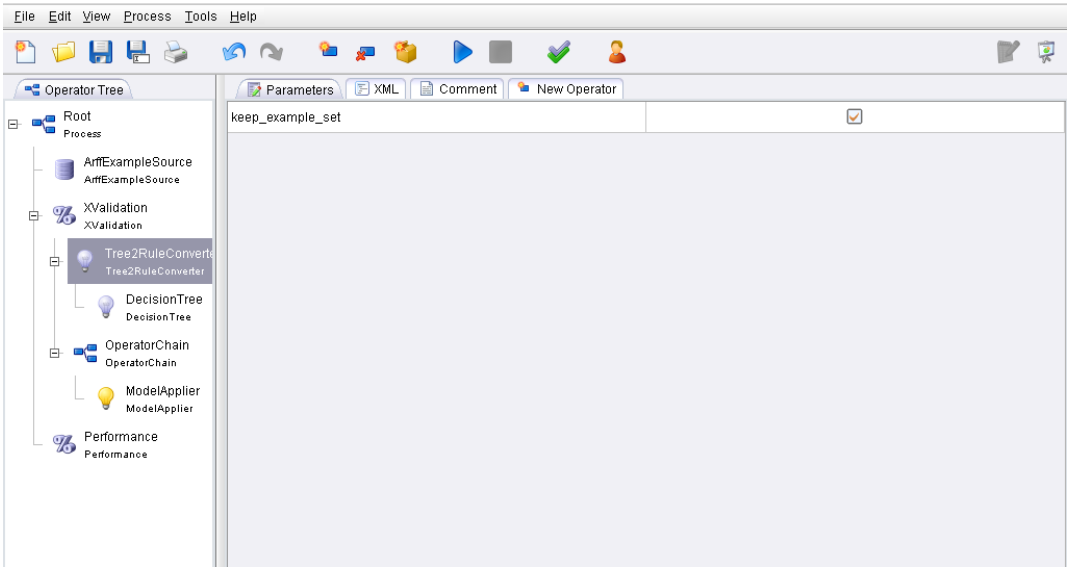


Figura 4.166. Agregando el operador Tree2RuleConverter.

A continuación se muestra parte del árbol obtenido en texto. Para verlo completo véase el anexo al final de este documento.

RuleModel

```
if plan <= 1994 and terminomaterias = NO then SI  (7066 / 191)
if plan <= 1994 and terminomaterias = SI then NO  (0 / 1738)
if plan > 1994 and generacion <= 2001 and terminomaterias = NO
then SI  (437 / 8)
if plan > 1994 and generacion <= 2001 and terminomaterias = SI then NO
(0 / 134)
if plan > 1994 and generacion > 2001 and periodos <= 14.500 and
creditos <= 0 and generacion <= 2,008.500 then SI  (278 / 49)
…
…
correct: 21171 out of 22460 training examples.
```

Evaluando el modelo.

Por medio del operador Performance se evaluará el desempeño del modelo generado. *Véase la figura 4.167*:

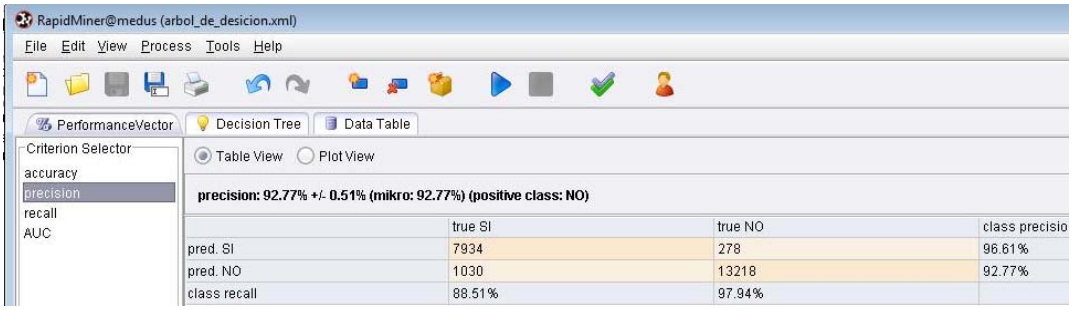


Figura 4.167. Evaluación del modelo por parte del operador Performance.

Los resultados de precisión (*precision*), exactitud (*accuracy*) y la AUC (*Area under curve*) son los siguientes:

```
Precision: 92.77% +/- 0.51 % (mikro: 92.77%)
classification_error: 0.97% +/- 0.12% (mikro: 0.97%)
correlation: 0.934 +/- 0.008 (mikro: 0.934)
```

Y la tabla de la matriz de confusión es la siguiente:

	true NO	true SI	class precision
pred. NO	7934	278	96.61%

pred. SI	1030	13218	92.77%
class recall	88.51%	97.94%	

También se obtiene la gráfica de análisis de ROC (Received Characteristic Operator). *Figura 4.168*.

Figura 4.168. Gráfica de análisis ROC.

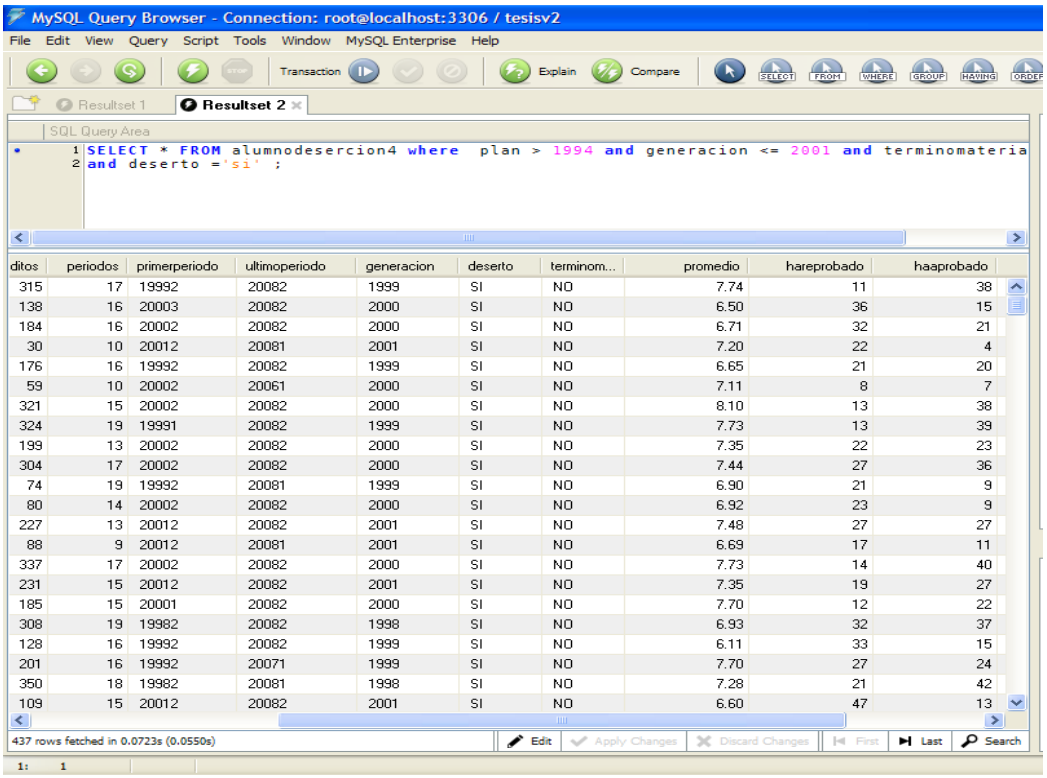
Visualización e interpretación del modelo.

Como se puede observar, este árbol arrojó reglas donde los que sobresalen son datos del nuevo plan de estudios 2006 y que en la mayoría de ellos corresponde a la no deserción pero, a pesar de ello, se enfocará a las reglas sobre deserción.

Como en el árbol de *Random Forest*, se seleccionarán aquellas reglas más trascendentes, es decir las que contengan mayor número de incidencias.

Se aplica la primera regla de este nuevo árbol en MySQL Query Browser (*figura 4.169*).

SELECT * FROM alumnodesercion4 WHERE plan > 1994 AND generacion <= 2001 AND terminomaterias = 'NO' AND deserto ='SI';



ditos	periodos	primerperiodo	ultimoperiodo	generacion	deserto	terminom...	promedio	hareprobado	haaprobado
315	17	19992	20082	1999	SI	NO	7.74	11	38
138	16	20003	20082	2000	SI	NO	6.50	36	15
184	16	20002	20082	2000	SI	NO	6.71	32	21
30	10	20012	20081	2001	SI	NO	7.20	22	4
176	16	19992	20082	1999	SI	NO	6.65	21	20
59	10	20002	20061	2000	SI	NO	7.11	8	7
321	15	20002	20082	2000	SI	NO	8.10	13	38
324	19	19991	20082	1999	SI	NO	7.73	13	39
199	13	20002	20082	2000	SI	NO	7.35	22	23
304	17	20002	20082	2000	SI	NO	7.44	27	36
74	19	19992	20081	1999	SI	NO	6.90	21	9
80	14	20002	20082	2000	SI	NO	6.92	23	9
227	13	20012	20082	2001	SI	NO	7.48	27	27
88	9	20012	20081	2001	SI	NO	6.69	17	11
337	17	20002	20082	2000	SI	NO	7.73	14	40
231	15	20012	20082	2001	SI	NO	7.35	19	27
185	15	20001	20082	2000	SI	NO	7.70	12	22
308	19	19982	20082	1998	SI	NO	6.93	32	37
128	16	19992	20082	1999	SI	NO	6.11	33	15
201	16	19992	20071	1999	SI	NO	7.70	27	24
350	18	19982	20081	1998	SI	NO	7.28	21	42
109	15	20012	20082	2001	SI	NO	6.60	47	13

Figura 4.169. Visualización de los datos de la regla obtenida.

Obteniendo la gráfica de *Pareto* (*figura 4.170*):

Las materias más significativas en esta selección fueron la 1100 (Algebra), 1402 (Hidráulica Básica), 1209 (Dibujo Mecánico e Industrial), 1547 (Dinámica de Sistemas Físicos), 1654 (Dispositivos y Circuitos Electrónicos).

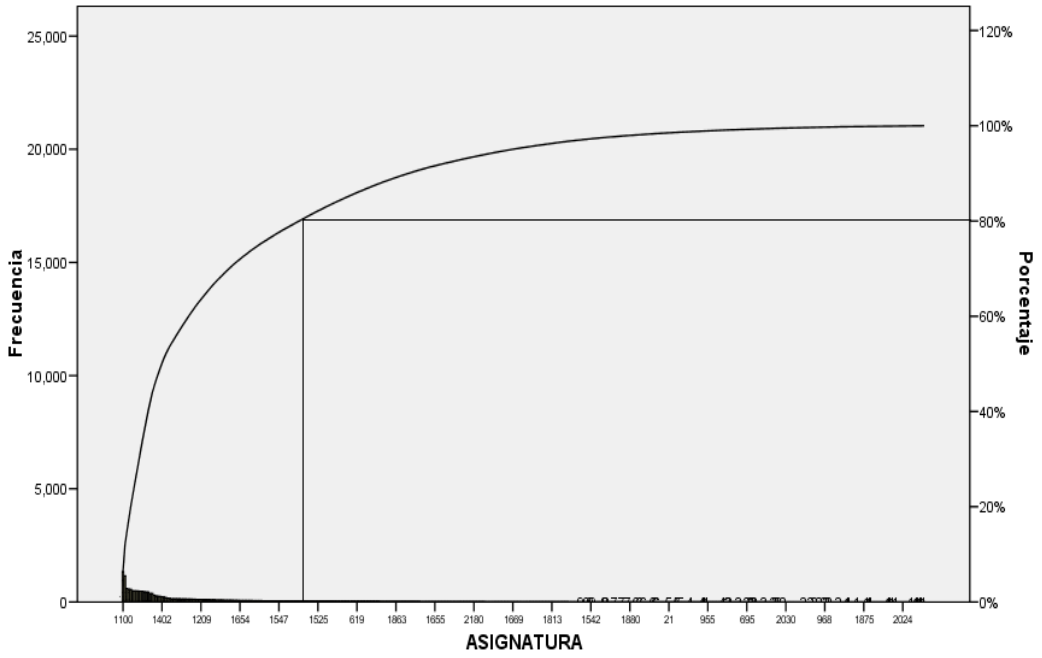


Figura 4.170. Diagrama de Pareto de las materias más reprobadas para este grupo.

Analizando los datos de Hidráulica Básica.

Es una materia que se imparte para los alumnos de Ingeniería Civil plan 2006 en el quinto semestre, y que está seriada con otras dos para sexto semestre. Asimismo se puede observar que, aunque la mayoría de los alumnos que aprobaron esta materia, lo hicieron hasta después de una ocasión por lo que es relevante el atraso de dicha materia al estar seriada con otras dos materias. Véase la *figura 4.171*.

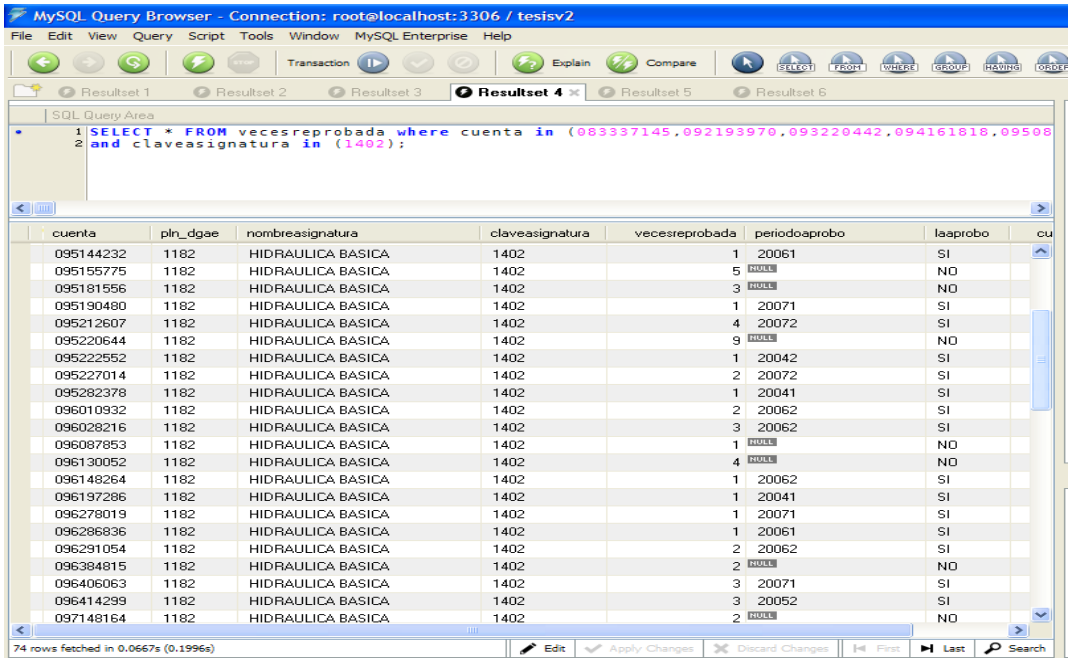


Figura 4.171. Análisis de la materia Hidráulica básica.

Analizando los datos de dibujo Mecánico e Industrial (*figura 4.172*).

Esta materia se imparte a las carreras de Ingeniería Industrial, Mecánica y Mecatrónica.

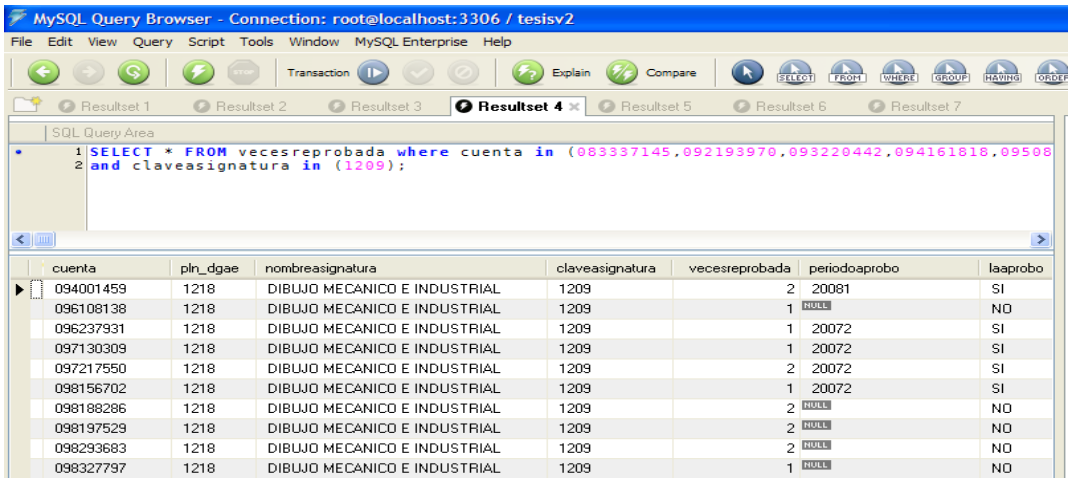


Figura 4.172. Análisis de la materia Dibujo mecánico e industrial.

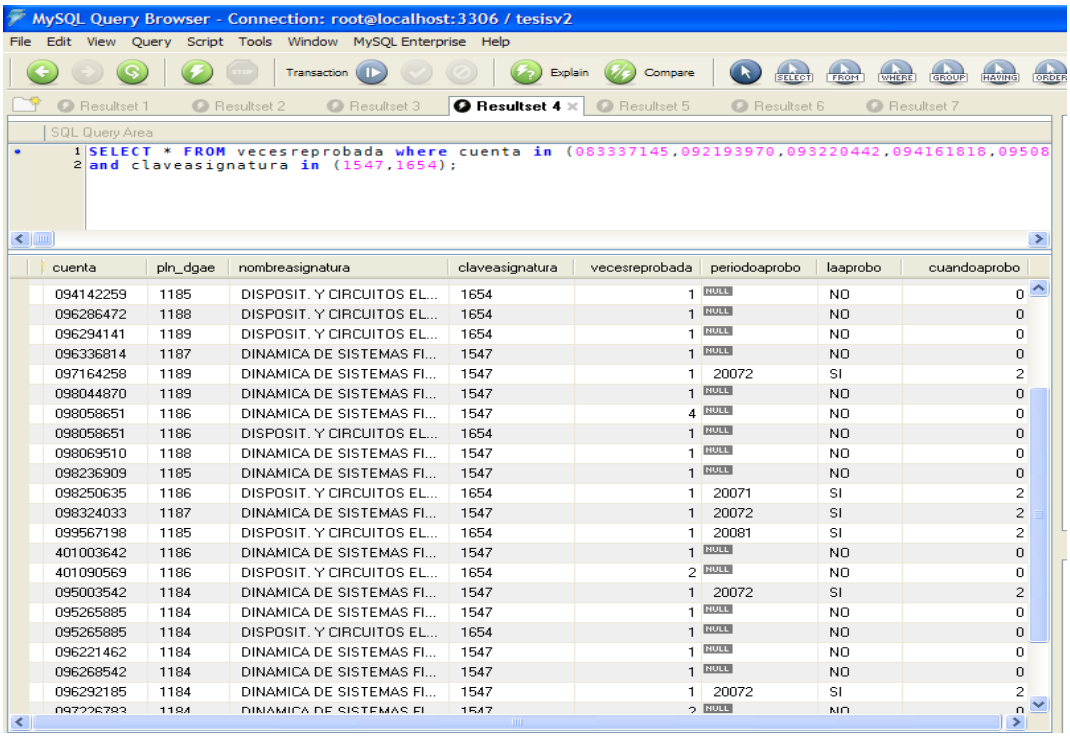
En esta materia, los alumnos que se atrasan fueron únicamente los Ingenieros mecánicos y fueron nuevamente los que cambiaron de plan de estudios. Cabe resaltar que dicha

materia se revalidó por la materia de Análisis Grafico por lo que estos alumnos no pudieron aprobar esa materia ni tampoco la de Dibujo Mecánico e Industrial.

Analizando los datos de Dinámica de Sistemas Físicos y Dispositivos y Circuitos Eléctricos (*figura 4.173*).

La materia de Dinámica de Sistemas Físicos en el plan 2006 únicamente se imparte a los alumnos de la carrera de Ingeniería eléctrica electrónica en el quinto semestre y está seriada con una materia del sexto semestre.

La materia de Dispositivos y Circuitos Eléctricos se imparte a las carreras de Ingeniería en Computación y Eléctrica Electrónica en el sexto semestre, la segunda tiene una seriación con otras dos materias para el séptimo semestre.



cuenta	pln_dgae	nombreasignatura	claveasignatura	vecesreprobada	periodoaprobo	laaprobo	cuandoaprobo
094142259	1185	DISPOSIT. Y CIRCUITOS EL...	1654	1	NULL	NO	0
096286472	1188	DISPOSIT. Y CIRCUITOS EL...	1654	1	NULL	NO	0
096294141	1189	DISPOSIT. Y CIRCUITOS EL...	1654	1	NULL	NO	0
096336814	1187	DINAMICA DE SISTEMAS FI...	1547	1	NULL	NO	0
097164258	1189	DINAMICA DE SISTEMAS FI...	1547	1	20072	SI	2
098044870	1189	DINAMICA DE SISTEMAS FI...	1547	1	NULL	NO	0
098058651	1186	DINAMICA DE SISTEMAS FI...	1547	4	NULL	NO	0
098058651	1186	DISPOSIT. Y CIRCUITOS EL...	1654	1	NULL	NO	0
098069510	1188	DINAMICA DE SISTEMAS FI...	1547	1	NULL	NO	0
098236909	1185	DINAMICA DE SISTEMAS FI...	1547	1	NULL	NO	0
098250635	1186	DISPOSIT. Y CIRCUITOS EL...	1654	1	20071	SI	2
098324033	1187	DINAMICA DE SISTEMAS FI...	1547	1	20072	SI	2
099567198	1185	DISPOSIT. Y CIRCUITOS EL...	1654	1	20081	SI	2
401003642	1186	DINAMICA DE SISTEMAS FI...	1547	1	NULL	NO	0
401090569	1186	DISPOSIT. Y CIRCUITOS EL...	1654	2	NULL	NO	0
095003542	1184	DINAMICA DE SISTEMAS FI...	1547	1	20072	SI	2
095265885	1184	DINAMICA DE SISTEMAS FI...	1547	1	NULL	NO	0
095265885	1184	DISPOSIT. Y CIRCUITOS EL...	1654	1	NULL	NO	0
096221462	1184	DINAMICA DE SISTEMAS FI...	1547	1	NULL	NO	0
096268542	1184	DINAMICA DE SISTEMAS FI...	1547	1	NULL	NO	0
096292185	1184	DINAMICA DE SISTEMAS FI...	1547	1	20072	SI	2
097226783	1184	DINAMICA DE SISTEMAS FI...	1547	2	NULL	NO	0

Figura 4.173. Análisis de las materias Dispositivos y circuitos eléctricos y Dinámica de sistemas físicos.

Para los alumnos de Ingeniería en Computación no tiene gran trascendencia la materia de Dispositivos y Circuitos Eléctricos ya que no son muchos los casos que se encontraron dentro de estos alumnos. Pero los alumnos de Ingeniería Eléctrica sí tienen problemas ya que estas dos materias al reprobarlas, representan un verdadero atraso en su carrera puesto que las dos tienen seriación con dos más.

Estudiando la siguiente regla (*véase la Figura 1.174*):

SELECT * FROM alumnodesercion4 WHERE plan > 1994 AND generacion > 2001 AND periodos <= 14.500 AND creditos <= 0 AND generacion <= 2008 AND deserto='SI';

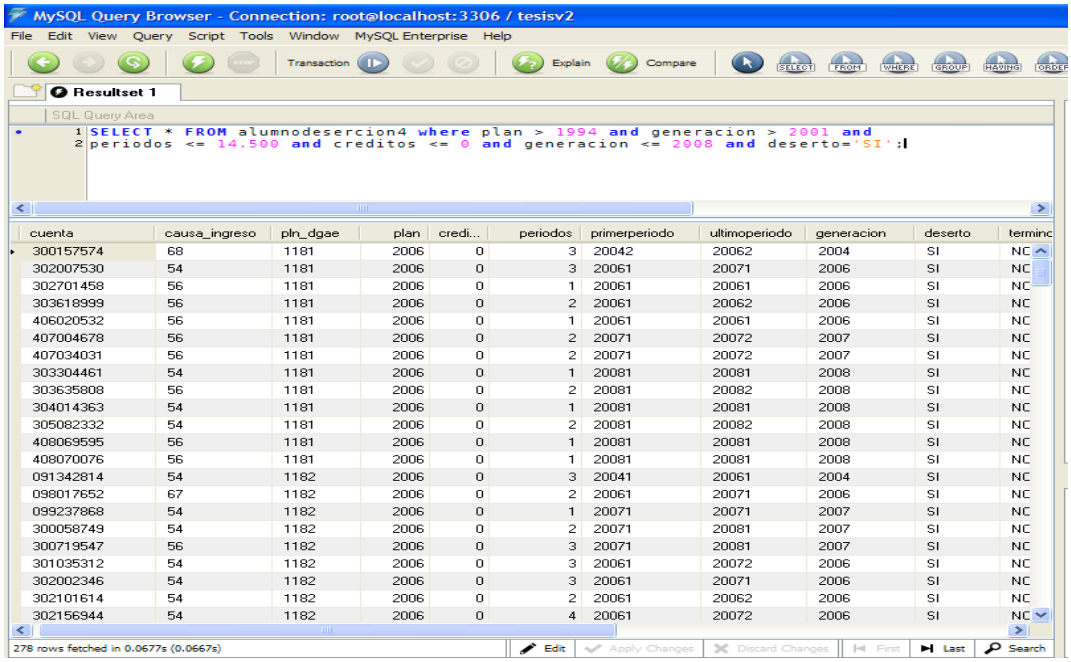


Figura 4.174. Visualización de los datos para la siguiente regla.

Se realiza el procedimiento para obtener el número de cuenta para visualizar el historial de los alumnos (*figura 4.175*):

Lo trascendente al obtener el historial de estos alumnos fue encontrar que no tenían ninguna materia aprobada.

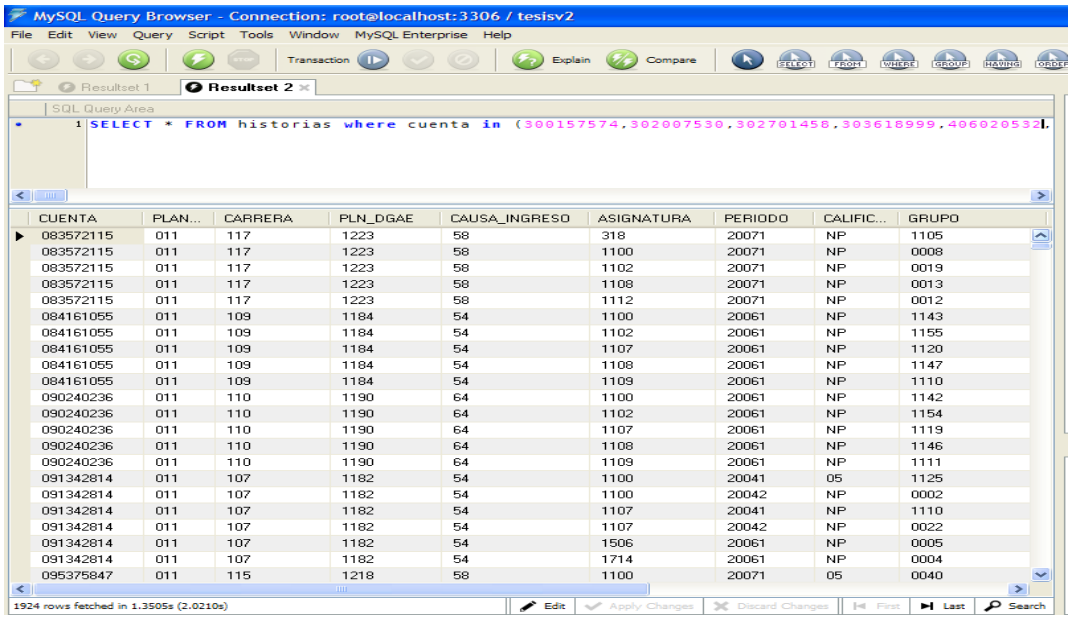


Figura 4.175. Viualización de historiales.

Se obtendrá la gráfica de *Pareto* para visualizar las materias más trascendentes en el índice de reprobación de estos alumnos (*figura 4.176*):

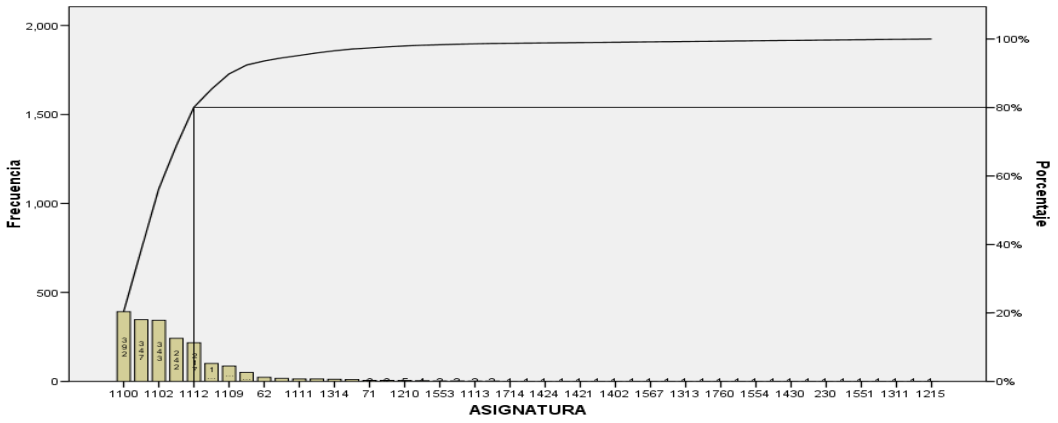


Figura 4.176. Diagrama de Pareto para las materias más reprobadas para este grupo de alumnos.

Las cuales son 1100(Algebra), 1102(Geometría Analítica) y la 1112(Computación para ingenieros)

De la consulta que se realizó en vecesreprobada, se pudo observar que todas las materias pertenecían a la División de Ciencias Básicas y a no más del tercer semestre. *Véase Figura 4.177.*

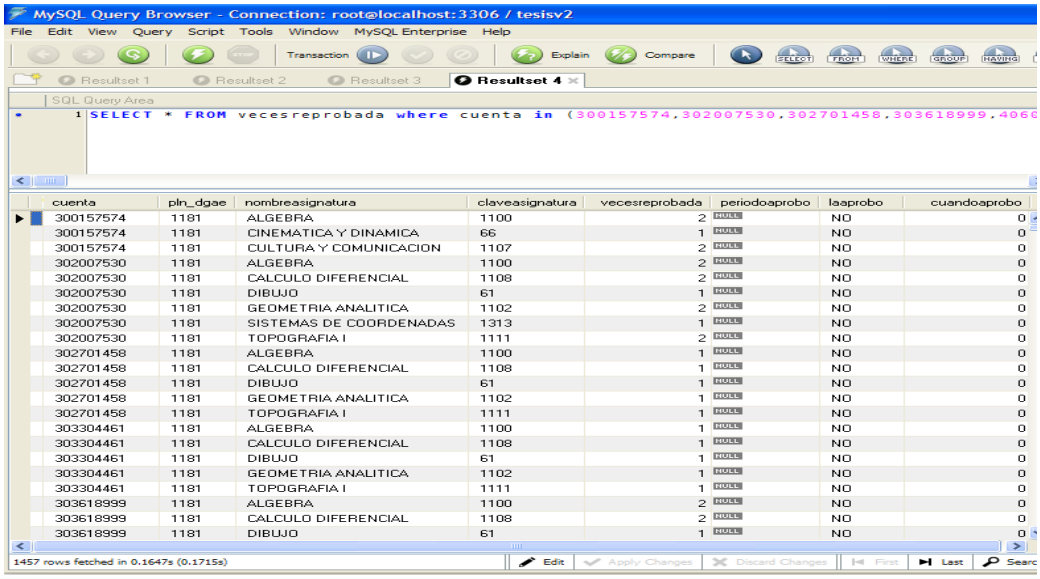


Figura 4.177. Visualización de la tabla *vecesreprobada*.

La regla indicó, por el rango de generaciones menores a la 2001, que se trataba de alumnos que se habían cambiado de plan de estudios. Al realizar el análisis de sus historiales y no encontrar materias aprobadas, se puede deducir que tampoco en el plan anterior habían aprobado ninguna materia ya que al hacer cambio de plan, se les revalidaban las materias aprobadas.

Desafortunadamente, estos alumnos ni con el cambio de plan de estudios pudieron obtener resultados satisfactorios en su carrera.

Por último, se aplicará la regla más representativa de los alumnos que no desertan (*figura 4.178*):

SELECT * FROM alumnodesercion4 WHERE plan > 1994 AND generacion > 2001 AND periodos <= 14.500 AND creditos > 0 AND generacion > 2002 AND haaprobado > 1 AND deserto ='NO';

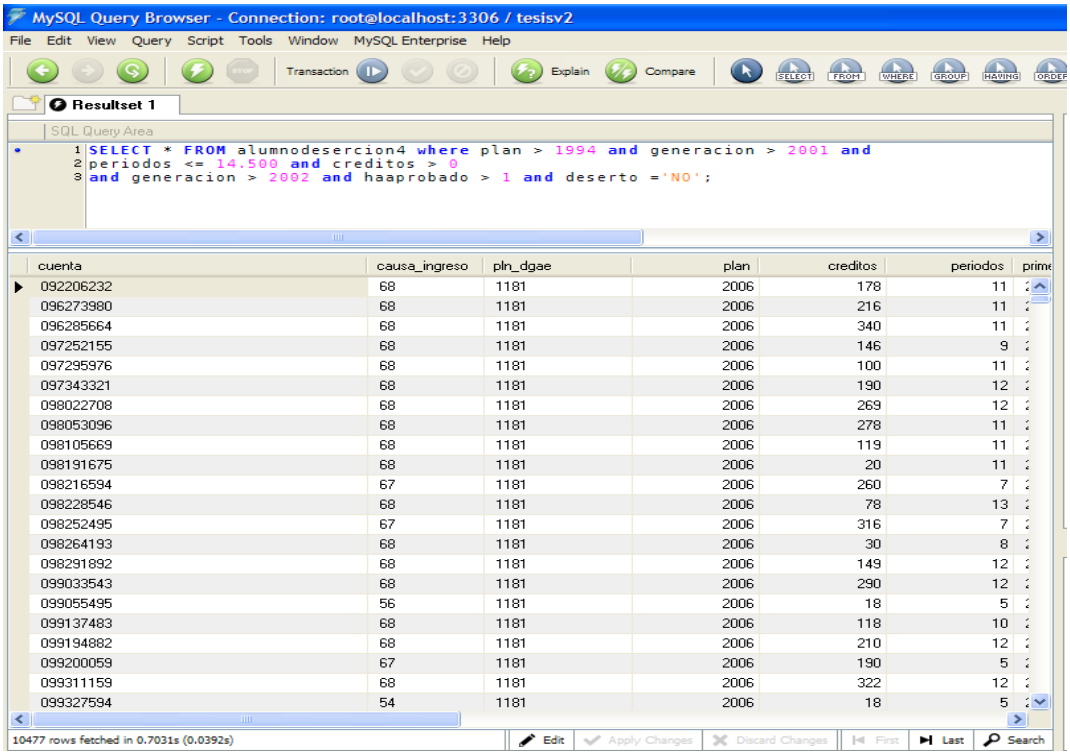


Figura 4.178. Visualización de los datos de la regla de los alumnos que no desertan.

El promedio más significativo para este grupo de alumnos es de 7 a 8. Existen promedios superiores o inferiores al mismo, pero la mayoría de ellos se concentra dentro de este rango (*figura 4.179 y 4.180*).

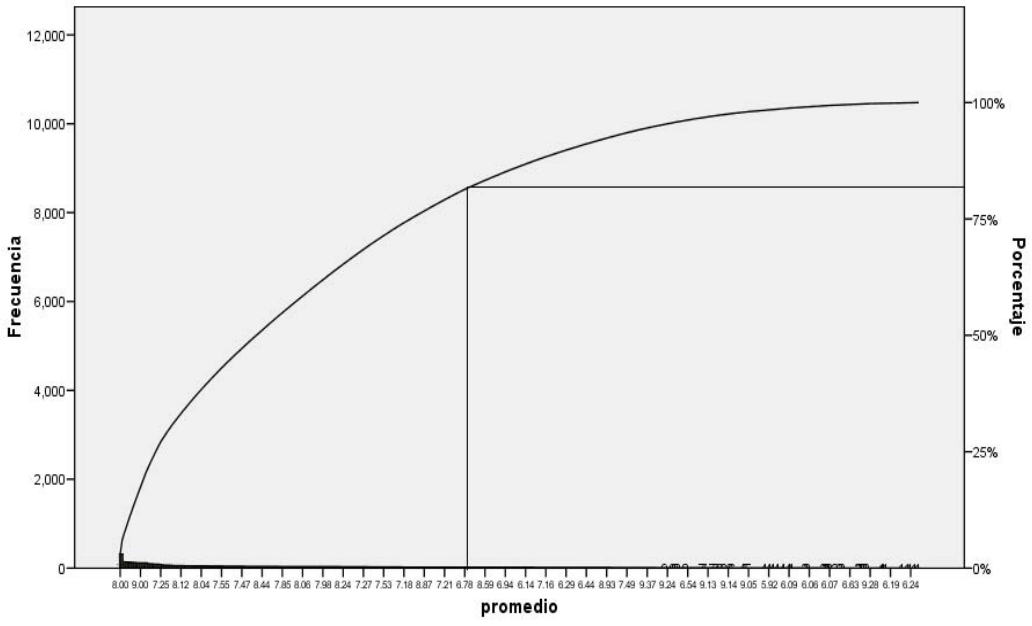


Figura 4.179. Diagrama de Pareto del promedio de los alumnos que no desertan.

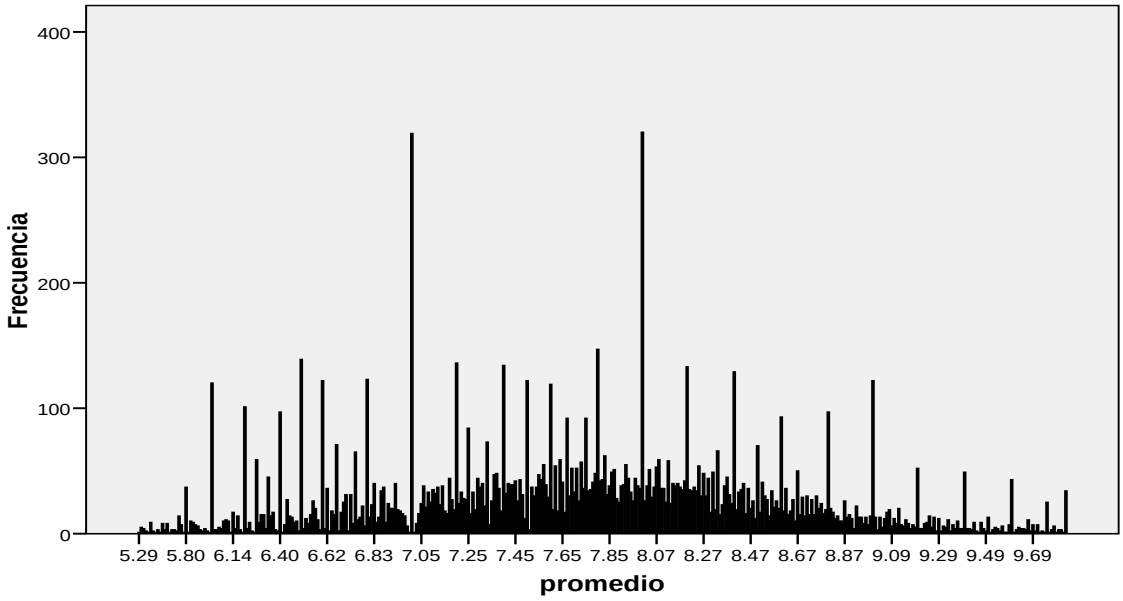


Figura 4.180. Gráfica de promedios por frecuencia.

También se realizan la gráfica de *Pareto* y la gráfica de frecuencias para el atributo de *hareprobado* (*figuras 4.181 y 4.182*):

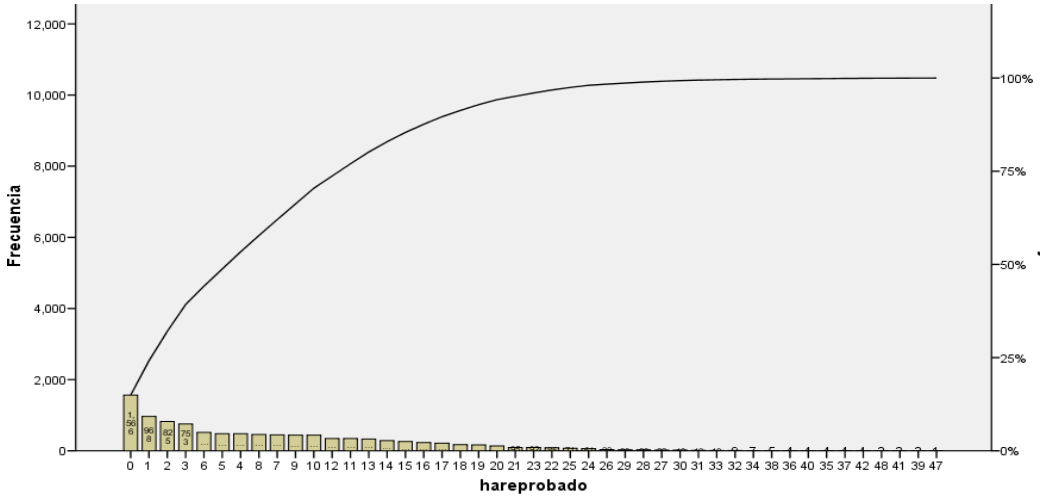


Figura 4.181. Diagrama de Pareto para el atributo *hareprobado*.

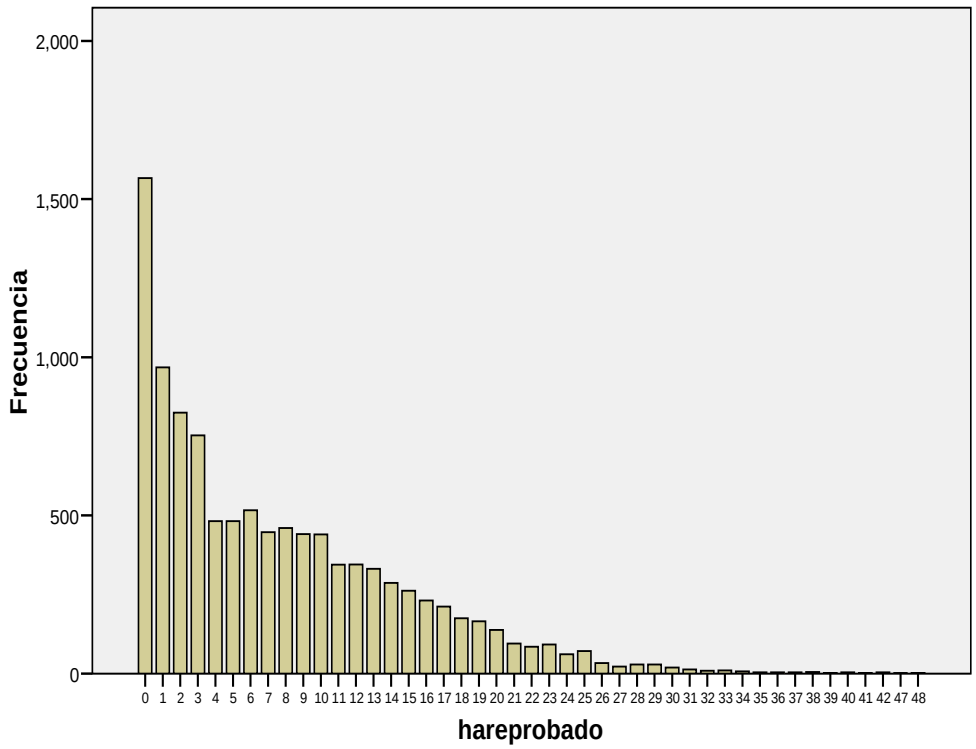


Figura 4.182. Gráfica del atributo *hareprobado* por frecuencias.

En esta gráfica se puede observar que la mayoría de estos alumnos tienen pocas materias reprobadas. En la gráfica de *Pareto* dentro del 80% de los casos más trascendentes se muestra que a lo mucho reprueban 14 materias la cual indica un poco más de 3 semestres de atraso. Aún así pueden seguir adelante sin ningún problema para lograr terminar sus estudios.

En este caso se puede concluir que, para que un alumno tenga éxito para terminar sus estudios, debe tener un promedio superior a 7 y tener a lo mucho atraso en materias reprobadas de no más de 14 materias ya que existen materias con seriación que pueden aumentar dicho atraso y que ocasionará que se termine el tiempo establecido para reinscripción lo cual hace que se desaliente el alumno a seguir adelante.

Tal vez se podrá decir que dicha información no resulta novedosa e interesante ya que es obvio tener un buen promedio para poder terminar la carrera. Pero si se compara a estos alumnos con los que desertaron, se encontrará que ellos reprueban más de dos ocasiones una materia y que no en una sola de ellas se presenta este comportamiento. *Vease Figura 4.183*.

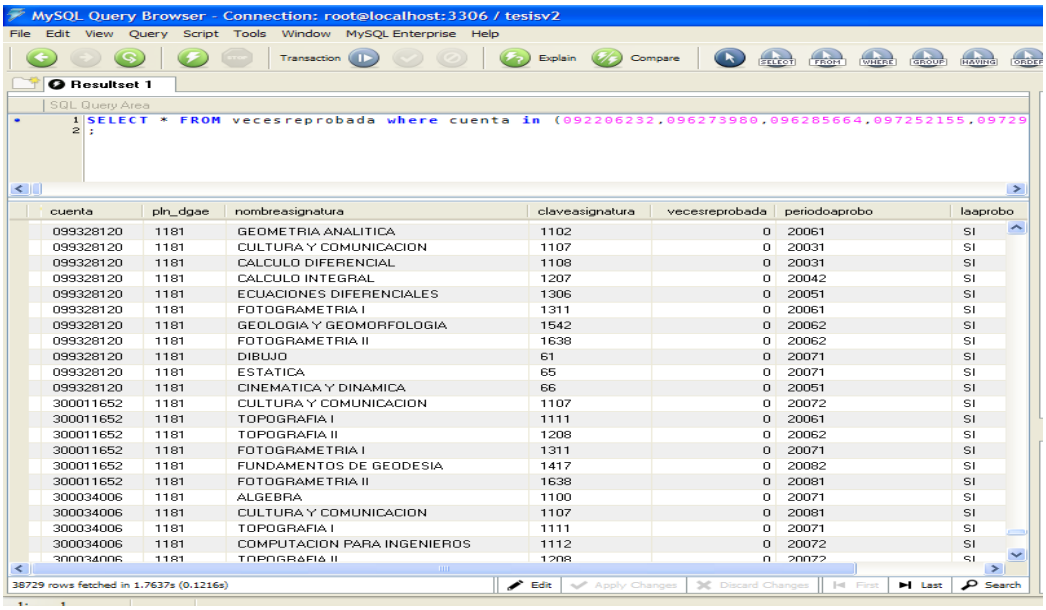


Figura 4.183. Visualización de la tabla *vecesreprobada*.

También se observa que los alumnos que cambiaron de plan de estudios tenían el mismo patrón de conducta manteniéndose dentro de los rangos de promedio y de materias reprobadas de tal forma que pudieran concluir todas las materias.

Conclusiones del capítulo cuatro

En el capítulo cuatro se realizó la parte práctica del estudio, cuya tarea fue ardua desde el principio (recolección y limpieza de los datos, generación de vistas minables), pasando por la búsqueda y evaluación e interpretación de los modelos. Asimismo se muestran las dificultades que se presentan al momento de realizar la limpieza y qué decisiones se tomaron para resolverlas, cómo se acomodaron los datos y la importancia de programar procedimientos como un método automatizado de limpieza en grandes tablas de datos.

Para el caso de las predicciones se trabajo únicamente con el algoritmo de clasificación IBk por tener un buen desempeño. En cambio con las redes neuronales el desempeño fue bajo para este caso.

CONCLUSIONES

• En el *capítulo uno* se presenta los conceptos necesarios para entender qué es y para qué sirve la minería de datos, las aplicaciones en las que se puede utilizar, así como los tipos de datos con los que se puede trabajar. Toda esta información de los conceptos básicos se obtuvo de los libros que se leyeron. Además nos aportaron las grandes aplicaciones de la minería (todo su potencial). Definitivamente se considera a la minería de datos una herramienta muy útil que proporciona la información (nuevo conocimiento útil y novedoso) que se necesita para apoyar a la toma de decisiones.

• En el *capítulo dos* se presenta el proceso del KDD el cual lleva al descubrimiento de conocimiento útil y novedoso. Se explica desde la recopilación de la información, cómo esta se trata y cómo se obtienen los modelos o patrones de comportamiento que llevan finalmente al descubrimiento de dicho conocimiento que funge como un apoyo adicional en la toma de decisiones.

Esta información se obtuvo de los libros citados en las referencias así como de asesorías por parte de especialistas en el ramo. Aporta el procedimiento que se debe seguir para lograr el objetivo principal que es la de encontrar nuevo conocimiento.

• En el *capítulo tres* se muestran algoritmos que se utilizan en la minería de datos. Para el caso de reglas y patrones, se eligió el árbol de decisión (el cual está dentro del bosque aleatorio) por ser fácil de utilización y entendimiento. Para el caso de las predicciones, se mostraron dos algoritmos distintos que se pueden usar en las predicciones. Uno es el algoritmo de clasificación *IBk* basado en la cercanía de vecinos y, el otro, la red neuronal, basada en la estructura cerebral humana.

Se determinó considerar estos dos algoritmos dado que en *rapidminer* existen éstos y, por tanto, se investigó en la literatura y fuentes bibliográficas sobre su uso, su eficiciencia, cómo funcionan; después de esto, se consideraron muy factibles para que se obtuvieran buenos modelos a partir de nuestros datos. Para determinar cuál es el mejor para nuestro estudio, se llevó a cabo con la ayuda de las validaciones (las que

determinan qué tan bien *aprende* el modelo generado por cada uno de estos dos algoritmos).

• En el *capítulo cuatro* se realizó la parte práctica del estudio, cuya tarea fue ardua desde el principio (recolección y limpieza de los datos, generación de vistas minables), pasando por la búsqueda y evaluación e interpretación de los modelos. Asimismo se muestran las dificultades que se presentan al momento de realizar la limpieza y qué decisiones se tomaron para resolverlas, cómo se acomodaron los datos y la importancia de programar procedimientos como un método automatizado de limpieza en grandes tablas de datos.

Para el caso de las predicciones se trabajo únicamente con el algoritmo de clasificación IBk por tener un buen desempeño. En cambio con las redes neuronales el desempeño fue bajo para este caso.

→ Dado el estudio de KDD que se realizó en la Facultad de Ingeniería para determinar y/o encontrar patrones de conducta de los alumnos que desertan y los que concluyen con éxito, lo siguiente:

Tomando como referencia a los planes 1994 y 2006 como factor trascendente en el desempeño académico, que:

- Sí es importante una renovación constante de los planes de estudio para beneficio en la actualización de conocimiento mismo, debido a que hoy en día el mercado laboral así lo requiere para el uso de nuevas tecnologías. Sin embargo, esto no está íntimamente relacionado con la deserción ya que, aunque se cambie varias veces, no ayuda a la disminución de la deserción.

- Lo anterior se debe a que se encontraron que hubieron alumnos que, aunque se cambiaran de plan de estudios, no ayudó a que concluyeran su carrera.

Respecto a la seriación entre materias se encontró que:

- En muchos de los casos de deserción, sí era un factor importante y trascendente el aprobar o no materias con seriación, ya que ellas implican un cuello de botella empezando desde materias de ciencias básicas y materias de semestres de quinto, sexto y séptimo semestre. Esto es porque ocasionan un atraso inminente y mucho más si se llegan a reprobar varias de estas materias seriadas y varias veces.

También se realizó un estudio de predicción el cual se llevó a cabo tomando en cuenta el desempeño de generaciones desde la 1994 hasta la 2005 obteniendo que:

- Un dato histórico o bien sabido, las generaciones anteriores a la 1999, fueron las que presentaron el mayor número de deserción; asimismo al analizar se puede observar que esto se debe a la huelga que se desató en el año 1999, donde se cursaba el semestre 1999-2.

- Por otra parte se llegó a construir un modelo para predecir si un alumno tiene altas o bajas posibilidades de terminar satisfactoriamente la carrera. Ello ayudaría a dar una mejor orientación escolar al alumnado.

Respecto a las reglas encontradas por los árboles se puede decir que:

- El número aceptable de reprobación de materias es aproximadamente 14, siempre y cuando la mayoría de ellas no estén seriadas, ya que permite la conclusión aunque no oportuna pero sí dentro del tiempo establecido para la terminación de la carrera.

- En las reglas con mayor número de deserción, se encontró que los alumnos de estos casos, tenían un atraso inminente desde los primeros semestres, debido a la reprobación de más de una ocasión de alguna materia; lo que agravaba más su situación era que algunas de ellas presentaban seriación con una o más materias. Este patrón de conducta se presentaba en la mayoría de los casos.
- Respecto a las reglas obtenidas por alumnos con buen desempeño, se encontró que el patrón era tener un promedio en un rango de 7.00 a 8.00, y un número de materias reprobadas menores a 14.

Con el fin de mejorar este estudio se recomienda lo siguiente:

Debido a que la base de datos se encontraba incompleta por así decirlo ya que no contaba con los registros semestrales de la relación entre grupos y profesor, ni tampoco se encuentra en la base los datos de los alumnos (socio-económicos). Por lo que no se pudieron tomar en cuenta estos parámetros para asociarse con la deserción entre los alumnos.

Por ejemplo si se hubiera tenido en los registros del alumno, la materia, el grupo, horario y profesor, se hubiera podido buscar si existe una relación, ya sea entre el profesor, o si influye el horario, que aumente el índice de reprobación, el cuál por ende ocasiona la deserción.

También seria interesente conocer si existe una relación socio-económica, con el rendimiento y la deserción de los alumnos.

Conocer si influyen estos parámetros en la deserción podría utilizarse para:

- Otorgamiento de estímulos económicos, ya que hay personas que no obtienen la calificación requerida para hacer acreedores a una. Pero la minería podría ayudar a seleccionar aquellas personas que en realidad están interesadas en estudiar y que sus medios económicos no les permite dedicarle todo su empeño al estudio, por lo cual pueden dejar inconcluso sus estudios.

- También se podría realizar la evaluación del desempeño de los profesores, esto con la finalidad de ayudar y generar argumentos para los estímulos económicos de los profesores.

- Con respecto al factor horario, se evaluaría si es factible impartir los cursos o materias en ciertos horarios, esto debido a la capacidad humana que se tiene para al aprendizaje.

- Otro factor que podría estudiarse es el geográfico, ya que este es un factor importante para el buen rendimiento y desempeño de los alumnos, se podrían encontrar un círculo geográfico factible para que el tiempo que se invierte en ir y venir desde casa a la escuela, no desgastara demasiado al alumno, y con ello no se afectara el rendimiento.

Esto y otras cosas se podrían estudiar en un futuro para realizar una mejora a la situación académica que presenta hoy en día la facultad de ingeniería. Y porque no pensar en toda la UNAM.

Se sabe que se realizan estadísticas para realizar algunos estudios sobre aprovechamiento tanto de los alumnos como el de los profesores, pero la Minería es más allá de estas estadísticas, ya que esta tiene incluye el estudio estadístico, pero la ventaja es que se puede visualizar y relacionar los atributos, con algo especifico que se quiera conocer.

REFERENCIAS

[1]http://en.wikipedia.org/wiki/Data_mining, Conceptos de minería de datos.
Última consulta en Junio del 2009. Utilizado en el capítulo uno.

[2]http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/MineriaDatosBressan.htm
Trabajo monográfico de adscripción realizado por Griselda E. Bressán.
Conceptos de minería de datos. Última consulta en Junio del 2009.
Utilizado en el capítulo uno.

[3]http://msdn2.microsoft.com/es-es/library/ms175595.aspx
Conceptos de minería de datos. Última consulta en Junio del 2009.
Utilizado en el capítulo uno y dos.

[4]Willi Klösgen**.** *Handbook of Data Mining and Knowlede Discovery*. Oxford University, 1ra edición, Oxford 2002. Págs 33, 288-296.

[5]José Hernández Orallo, Ma. José Ramírez Quintana, César Ferri Ramírez. *Introducción a la Minería de Datos*. Pearson – Prentice Hall. Madrid, 2004. 1ra edición Págs. 10,12,13,15,43,65,68,24.

[6]David Hand, Heikki Mannila, Padhraic Smyth.*"Principles of Data Mining"*. MIT Press, 1ra edición, USA, 2001. Págs 3,141-182, 347-352.

[7]http://www.monografias.com/trabajos/datamining/datamining.shtml
Conceptos de minería de datos. Última consulta en Junio del 2009.
Utilizado en el capítulo uno.

[8] http://www.answermath.com/data-mining/mineria-de-datos-3-aplicaciones.htm
Aplicaciones de la minería de datos. Última consulta en Junio del 2009.
Utilizado en el capítulo uno.

[9]César Pérez López, *"DATA MINING Soluciones con Enterprise Miner"*, Ra-Ma, Primera edición Madrid, 2006. Págs. 10, 170, 173, 174, 177, 179, 207, 208, 251.

[10]http://en.wikipedia.org/wiki/Cross-validation_(statistics)
Concepto de validación cruzada. Última consulta en Junio del 2009.

Utilizado en el capítulo dos.

[11] http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos
Conceptos de modelos. Última consulta en Junio del 2009.
Utilizado en el capítulo tres.

[12]Han Kamber, *Data Mining Concepts and Techniques*. Morgan Kaufmann Publishers, Primera edición, USA, 2001. Págs 279, 303, 304.

[13]Rhonda Delmater. *"Data Mining Explained"*, Digital Press, Primera edición, USA 2001, Pág. 212.

[14]Westphal Christopher, Data Mining Solutions, Wiley Computer Publishing, Primera edición, USA 1998, Pág. 213.

[15]http://www.slideshare.net/pierluca.lanzi/machine-learning-and-data-mining-13-nearest-neighbor-and-bayesian-classifiers
Concepto del algoritmo del vecino más cercano. Última consulta en Junio del 2009.
Utilizado en el capítulo tres.

[16] Tamraparni Dasu, Theodore Jonson. *Exploratory Data Mining and Data Cleaning*. Wiley. 1ra edición, New Jersey, 2003. Págs. 18,1-81, 99-123, 162-175.

[17] Michalis Vazirigiannis, Maria Halkid, Dimitrios Gunopulos. *Uncertainty Handling and Quality Assessment in Data Mining*. Springer. 1ra edición, London, 2003. Págs. 11-25, 43-61.

[18] Ian H. Witten, Eibe Frank. *Data Mining. Practical Machine Learning tools and Techniques with Java Implementations*. Morgan Kaufmann Publishers. 1ra edición, USA, 2000. Págs. 4-54, 57-116, 119-132, 141-144.

[19] Kishan Mehrotra, Chilukuri K. Mohan, Sanjay Ranka. *Elements of Artificial Neural Networks*. The MIT Press. 1ra edición, Cambridge, Massachusetts, 1997. Págs. 16-39, 43-56, 65-106.

[20] B. Martín del Brío, A. Sanz Molina. *Redes Neuronales y Sistemas Difusos*. Ra-ma. 2da edición, Madrid, 1997. Págs. 243-268.

[21] Willi Klösgen**.** *Handbook of Data Mining and Knowlede Discovery*. Oxford University, 1ra edición, Oxford 2002. Págs 288-296.

[22] Sitio web para consultar las claves de planes de estudio de la Facultad de Ingeniería. http://www.dgae-siae.unam.mx/www_eqv.php?plt=011. Última consulta en Junio del 2009.

[23] Sitio web para consultar los programas de planes de estudios 2006 de la Facultad de Ingeniería: http://www.ingenieria.unam.mx/revplanes/planes2006/. Última consulta en Junio del 2009.

[24] Rapidminer: http://rapid-i.com. Sitio web del software de minería de datos. Última consulta en Junio del 2009.

[25] SPSS : http://www.spss.com/. Sitio web del paquete de análisis estadístico. Última consulta en Junio del 2009.

[26] Herramientas de MySQL. Sitio web desde el cual se puede bajar todo el software relacionado con *MySQL*. Última consulta en Junio del 2009. http://download.softagency.net/MySQL/Downloads/MySQLGUITools/

[27] Principio de Pareto. Sitio web en donde se explica dicho principio. Última consulta en Junio del 2009. http://es.wikipedia.org/wiki/Principio_de_Pareto

[28] Neural Market Trends. Sitio web en donde se puede encontrar varios ejemplos usando *Rapidminer*. Hágase clic en la sección *tutorials*. Última consulta en Junio del 2009. http://www.neuralmarkettrends.com

[29] Análisis ROC. Sitio web en donde se explica cómo interpretar una curva *ROC*. Última consulta en Junio del 2009. http://gim.unmc.edu/dxtests/ROC2.htm

[30] OLAP – OLTP. Sitio web en donde se explica las diferencias de estos conceptos. Última consulta en Junio del 2009.
http://www.sinnexus.com/business_intelligence/olap_vs_oltp.aspx

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

GLOSARIO

Algoritmo.

Lista bien definida, ordenada y finita de operaciones que permite hallar la solución a un problema.

Base de datos.

Es una coleccion de archivos interrelacionados, son creados con un manejador de bases de datos. Su contenido engloba la informacion concerniente de una organizacion de tal manera que los datos esten disponibles para los usuarios.

KDD (*Knowledge Discovery in Databases*).

Es el proceso de descubrir conocimiento útil y novedoso dentro de los datos.

Minería de datos.

Consiste en la extracción normalmente no trivial de información que reside de manera implícita en los datos. Dicha información es previamente desconocida y podrá resultar útil como apoyo en la toma de decisiones.

Modelo.

En este contexto, es el resultado del proceso de generar una representación abstracta, conceptual, gráfica, visual, física, matemática de fenómenos, sistemas o procesos a fin de analizar, describir, explicar, simular o predecir dichos fenómenos, sistemas o procesos.

Patrón.

Es un conjunto de reglas clasificadas según sus características en común.

Procedimiento.

Es un subprograma compuesto de sentencias SQL (lenguaje de consulta en bases de datos).

Vista minable.

Es una tabla que integra todos los datos ya ordenados, limpios y seleccionados. Dicha tabla está lista para llevar a cabo en ella la minería de datos.

APÉNDICE I

Software utilizado.

Todo el software que se usó fue bajo ambiente *Windows XP*. Para el caso del manejador de Base de Datos y para la Minería de Datos, se pueden usar en Linux.

A continuación se da una breve descripción del software y las razones por las cuales se utilizaron para los fines de la presente tesis.

Manejador de Bases de Datos: MySQL.

Para ir almacenando los datos se utilizó el manejador de bases de datos *MySQL* versión 5.0.45. De hecho se utilizaba una versión anterior. Se decidió cambiar de versión ya que en la anterior existían unos errores o *bugs* que impedían ejecutar unos procedimientos. Al principio creíamos que era un error nuestro en la programación que no podíamos detectar. Después de investigar en los foros, supimos que era un error en esa versión anterior de *MySQL* y que desde la versión 5.0.45 ya estaba solucionado.

Usamos *MySQL* porque permite ir almacenando los datos sin necesidad de establecer una llave primaria, sin necesidad de cuidar la integridad de los datos, permite la *desnormalización* o la repetición de columnas en varias tablas. Esto sonaría descabellado al momento de implementar una Base de Datos. Sin embargo, para la minería de datos esto es necesario. Ya que si, en vez de repetir las columnas, establecemos llaves foráneas, al momento de llevar a cabo la transformación de los datos a través de los procedimientos, los tiempos de ejecución de los mismos serían mucho mayores porque la computadora tendría primero que encontrar las columnas de miles de datos y luego realizar el procedimiento y otra vez volver a relacionar. Computacionalmente sería muy costoso. También, el tener las columnas repetidas facilita el análisis sin tener que hacer consultas más largas relacionando tablas cada vez que queramos hacer un distinto análisis.

MySQL es un software de uso libre y cualquier duda que exista o problema se puede consultarlo en los foros. También cuenta con la posibilidad de programar procedimientos. Esto último fue vital para la transformación de nuestros datos de manera automática.

Instalamos *MySQL* directamente junto con el servidor apache y PHP bajando e instalando desde Internet el *AppServ 2.5.9* para *Windows*.

En algunas situaciones usamos hasta *PHP* cuando necesitamos obtener cierta información de la Base de Datos (una consulta SQL) para una gran cantidad de números de cuenta específicos. Es decir, bajo la sintaxis *WHERE cuenta IN* (09753.. , 0972332, …). Si se tiene que teclear más de 100 números de cuenta, hacerlo a mano sería muy laborioso. Entonces con *PHP* se programa otra consulta *SQL* la cual obtiene sólo los números de cuenta deseados y se despliegan separados por coma. Todo esto se copia y se transfiere a nuestra consulta original y así evitamos teclear cada número de cuenta.

Por ejemplo, el *script* o el código *PHP* quedaría como sigue:

```
<?php
   mysql_connect("localhost", "root", "MiContraseña") or die(mysql_error());
   mysql_select_db("tesisV2") or die(mysql_error());

  $result = mysql_query("SELECT cuenta FROM alumnodesercion4
                         WHERE terminomaterias = 'SI'");

        while($row = mysql_fetch_assoc($result))
             {
              echo ($row['cuenta'].",");
             }
?>
```

http://www.mysql.com

http://www.appservnetwork.com/

Minería de datos: Rapidminer.

Rapidminer es un software en versión libre y en versión empresarial (o versión comercial con costo para su uso) el cual contiene una gran variedad de algoritmos para el análisis de los datos. En principio, es difícil aprender a utilizarlo. Antes de empezar a hacer la minería, primero hay que armar un árbol de procesos u operadores que irán llevando a cabo las fases de nuestra minería; como por ejemplo: cargar los datos, aplicar el algoritmo deseado, hacer una validación, guardar el modelo, exportar los datos a un archivo, etc.

A pesar de que existen algunos tutoriales o manuales en la misma página de *Rapidminer*, no son suficientes; sin embargo, los creadores de *Rapidminer* ofrecen cursos en Estados Unidos y en Alemania a un alto costo. Entonces tuvimos que aprender nosotros mismos a base de ensayo y error. Asimismo existe un sitio en Internet que ayuda a aprender a utilizar el programa el cual se llama *Neural Market Trends*.

Rapidminer es un programa muy versátil y poderoso con el que se puede llevar a cabo prácticamente cualquier proyecto de Minería de Datos tan grande como se quiera, además de tener la posibilidad de automatizar los proyectos e incorporarlos a un Sistema de Información.

http://www.rapidminer.com
http://www.neuralmarkettrends.com/

Para cargar los datos:

Los datos se dieron en un archivo *SQL* (la tabla de los historiales académicos). Las demás tablas se dieron en archivo con extensión *.xls*. El gran problema fue cargar la tabla *historias* la cual pesaba casi 17 MB, y en nuestras computadoras el *software* usado para cargar los datos no podía cargar todos los datos de una sola vez cuando esta tabla contenía más de un millón de renglones. Cargar estos datos a nuestro manejador de bases de datos fue una tarea desafiante, tardada y computacionalmente costosa.
Se decidió partir este archivo (que contenía la tabla *historias*) usando un programa gratuito que se llama *GSplit File Splitter*. Creímos que partiendo el archivo en cinco

partes podría cargarse. No fue posible, y así se fue probando hasta que salieron 28 archivos cuyo tamaño era de 5 MB cada uno, pero aún así, cargar cada archivo resultó ser muy tardado (en promedio poco más de una hora por cada archivo).

También hubo situaciones en que, como resultado de la Minería de Datos o ejecución de los procedimientos, obteníamos los resultados sólo en archivos de *Excel* o en formato de separación por coma por lo que teníamos que cargar los datos de *Excel* a *MySQL* o simplemente para poder cargar las tablas de la Base de Datos de la Facultad. Para esto, usamos el programa *Excel MySQL Import Export & Convert Software*; el cual se instala automáticamente en *Excel* a manera de macro y puede usarse fácilmente para cargar los datos a *MySQL*. No es software libre.

http://www.gdgsoft.com/gsplit/
http://www.sobolsoft.com/excelmysql/

Paquete estadístico: SPSS 15.0

Para obtener las estadísticas y explorar los datos, se utilizó el programa SPSS el cual es un paquete estadístico muy completo y que puede manejar una mayor cantidad de datos que *Excel*. Asimismo, posee una mayor gama de gráficas que se pueden obtener de una manera rápida y sencilla. Si no se sabe utilizar el programa, SPSS incluye en la instalación un tutorial o manual para aprender a utilizarlo. No es un software libre.

http://www.spss.com/

APÉNDICE II

A continuación se presenta una miniguía de cómo ir actualizando los datos con el fin de ir alimentando el modelo y así poder dar seguimiento a los cambios.

Cada semestre se recaba las calificaciones que obtuvieron los alumnos. Para este caso basta que se proporcione la tabla historias del semestre que acaba de pasar, es decir, *where periodo=200xy*, donde *x* es el año y *y* es el número de semestre. Estos datos deben cargarse a la tabla *historias* de nuestro depósito de datos. Se procede asimismo a ejecutar los doce procedimientos nuevamente y luego encontrar los patrones y actualizar los modelos con *Rapidminer* tal y como se ha descrito en el presente trabajo.

ANEXO

En el presente anexo se incluye los códigos fuente completos de los procedimientos que se programaron así como las tablas de datos que son largas.

• ***Tabla 4.1***

status_plan	plan_dgae	Carrera
1	0365	ing. civil 1994
1	0371	ing. de minas y metalurgista 1994
1	0376	ing. Geologo 1994
1	0381	ing.petrolero 1994
1	0386	ing.topografo y geodesta 1994
1	0403	ing. Geofisico 1994
1	0408	ing. en computacion 1994
1	0410	ing. en telecomunicaciones 1994
1	0411	ing.mecanico 1994
1	0412	ing industrial 1994
1	0413	ing. Electrico electronico 1994
1	0722	ing. Geologo-geologia petrolera 1994
1	0723	ing. Geologo-geologia minera 1994
1	0724	ing. Geologo-geotecnia 1994
1	0725	ing. Geologo-geohidrologia 1994
1	0726	ing. Geologo -geologia ambiental 1994
1	0785	ing. Geofisico-explotacion petrolera 1994
1	0786	ing. Geofisico-explotacion minera 1994
1	0787	ing. Geofisico-sismologia1994
1	0788	ing. Geofisico-geotecnia 1994
1	0789	ing. Geofisico-ambiental 1994
1	0790	ing. Geofisico-ciencias atmosfericas 1994
1	0791	ing. Geofisico-geohidrologia 1994
1	0797	ing. en telecom mod-procesa digital de señales 1994
1	0798	ing. en telecom mod-radiocominicaciones 1994
1	0799	ing. en telecom mod-redes de comunicaciones 1994
1	0800	ing. en telecom mod-servicios y sistemas 1994
1	0801	ing. en telecom mod-disp de micro y opticos 1994
1	0805	ing.mecanico mod-diseño1994
1	0806	ing.mecanico mod-manofactura y material 1994
1	0807	ing.mecanico mod-ter y mejora amb 1994
1	0808	ing.mecanico mod-mecatronica 1994
1	0809	ing industrial mod-produccion1994
1	0810	ing industrial mod-admon y sistemas
1	0811	ing. elec electr md-electronica 1994
1	0812	ing. elec electr md-sistemas digitales 1994
1	0813	ing. elec electr md-electronica para comunicaciones 1994

1	0814	ing. elec electr md-energia electrica 1994
1	1096	ing. elec electr md-biomedicas 1994
1	1097	ing. en telecom mod-biomedicas 1994
1	1098	ing.mecanico mod-biomedicas 1994
1	1111	ing.mecatronico 2004
2	1181	ing.geomatico 2006
2	1182	ing. civil 2006
2	1183	ing. de minas y metalurgista 2006
2	1184	ing. Electrico electronico 2006
2	1185	ing. Electrico electronico-electronica 2006
2	1186	ing. Electrico electronico-control y robotica 2006
2	1187	ing. Electrico electronico-biomedicas 2006
2	1188	ing. Electrico electronico-potencia 2006
2	1189	ing. Electrico electronico-de sistemas energeticos 2006
2	1190	ing. en computacion 2006
2	1191	ing. en computacion- ingenieria de hardwer 2006
2	1192	ing. en computacion-redes y seguridad 2006
2	1193	ing. en computacion-bases de datos 2006
2	1194	ing. en computacion-ingenieria de software 2006
2	1195	ing. en computacion- sistemas inteligentes y comp grafica 2006
2	1196	ing. en computacion- biomedicas 2006
2	1197	ing. en telecomunicaciones 2006
2	1198	ing. en telecomunicaciones mod-redes de telecom 2006
2	1199	ing. en telecomunicaciones mod-señ y sist radiocom 2006
2	1200	ing. en telecomunicaciones mod-tec radiofrec opt 2006
2	1201	ing. en telecomunicaciones mod-adminis y normali 2006
2	1202	ing. Geofisico 2006
2	1203	ing. Geofisico-explotacion minera 2006
2	1204	ing. Geofisico-explotacion-sismologia 2006
2	1205	ing. Geofisico-ciencias atmosfericas 2006
2	1206	ing. Geofisico-hidrogeologia 2006
2	1207	ing. Geofisico-explotacion petrolera 2006
2	1208	ing. Geofisico-geotecnia 2006
2	1209	ing. Geofisico-ambiental 2006
2	1210	ing. Geologo 2006
2	1211	ing. Geologo-geologia petrolera 2006
2	1212	ing. Geologo-geologia geotecnia 2006
2	1213	ing. Geologo-minera 2006
2	1214	ing. Geologo-hidrologia y geologia ambiental 2006
2	1215	ing industrial 2006
2	1218	ing.mecanico 2006
2	1223	ing.petrolero 2006
2	1224	ing.mecatronico 2006

Procedimientos.

• *alumnodesercion.sql* (procedimiento 1).

```
DELIMITER $$

DROP PROCEDURE IF EXISTS `tesisv2`.`alumnodesercion` $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `alumnodesercion`()
BEGIN
/*atributos de la tabla alumnodesercion*/
DECLARE cuenta_ VARCHAR(10);
DECLARE causaingreso_ VARCHAR(2);
DECLARE creditos_ INTEGER;
DECLARE periodos_ VARCHAR(5);
DECLARE primerperiodo_ VARCHAR(5);
DECLARE ultimoperiodo_ VARCHAR(5);
DECLARE deserto_ VARCHAR(2);
DECLARE status_plan_ INTEGER;
DECLARE plan_ INTEGER;

/*Variables temporales*/
DECLARE plan_temp VARCHAR(5);
DECLARE creditos_temp INTEGER;
DECLARE creditos_alumno INTEGER;
DECLARE terminomaterias_ VARCHAR(2);
DECLARE ultimo_renglon INT DEFAULT 0;
                /*QUERIES*/
/*1.- Obtener las carreras con sus respectivo número de créditos y plan 94 ó 2006*/
/*Revisar los nulos porque parecen haber registros de plan dgae nulos*/
/*QUERY 1*/
DECLARE fetch_carreras CURSOR FOR
SELECT pln_dgae,creditos,status_plan FROM carreras_planes
ORDER BY pln_dgae;

/*2.- Para una carrera dada, obtener los números de cuenta que existen (de la tabla
historias):*/
/*QUERY 2*/
DECLARE fetch_cuentas CURSOR FOR
SELECT DISTINCT cuenta,causa_ingreso FROM historias
WHERE pln_dgae=plan_temp;

/*3.- Para cada número de cuenta, contar el número de semestres que ha cursado*/
/*QUERY 3*/
DECLARE fetch_num_periodos CURSOR FOR
SELECT COUNT(DISTINCT periodo) FROM historias
WHERE cuenta=cuenta_;

/*4.- Ahora, para ese número de cuenta, obtener su número de créditos y el último semestre
```

```
cursado (max(periodo)) y la causa de ingreso*/
/*QUERY 4*/
DECLARE fetch_creditos_ultperiodo CURSOR FOR
SELECT MIN(DISTINCT h.periodo) AS primerperiodo, MAX(DISTINCT h.periodo) AS
ultimoperiodo,SUM(a.creditos) AS creditos
FROM historias h, asignatura a
WHERE a.clave=h.asignatura
AND h.aprobo='SI' /*analizar después los alumnos que nunca aprobaron alguna materia-->NULL,
verificar que el
ultimo periodo sea efectivamente el ultimo periodo!*/
AND cuenta=cuenta_;

/*5.- Finalmente, determinar si el alumno desertó o no.*/

/*Variable de control*/
DECLARE CONTINUE HANDLER FOR NOT FOUND SET ultimo_renglon=1;
/*procedimiento*/
OPEN fetch_carreras;
ciclo_carreras:LOOP
  FETCH fetch_carreras INTO plan_temp, creditos_temp, status_plan_;
    IF ultimo_renglon=1 THEN
      LEAVE ciclo_carreras;
    END IF;
      OPEN fetch_cuentas;
      ciclo_fetch_cuentas:LOOP
        FETCH fetch_cuentas INTO cuenta_,causaingreso_;
          IF ultimo_renglon=1 THEN
            LEAVE ciclo_fetch_cuentas;
          END IF;
              OPEN fetch_num_periodos;
                FETCH fetch_num_periodos INTO periodos_;
              CLOSE fetch_num_periodos;
              OPEN fetch_creditos_ultperiodo;
FETCH fetch_creditos_ultperiodo INTO
primerperiodo_,ultimoperiodo_,creditos_alumno;
              CLOSE fetch_creditos_ultperiodo;
              /*determinando si desertó o no Considerando ultimo periodo, creditos, etc*/
                IF creditos_alumno>=(creditos_temp-3) THEN
                  SET deserto_='NO';
                ELSEIF ultimoperiodo_='20091' THEN
                  SET deserto_='NO';
                ELSE
                  SET deserto_='SI';
                END IF;
              /*DETERMINANDO QUÉ PLAN ES, si es 1=1994, si es 2=2006*/
                IF status_plan_=1 THEN
                  SET plan_=1994;
                ELSEIF status_plan_=2 THEN
                  SET plan_=2006;
                END IF;
```

```
                /*Llenando los datos en la tabla:*/
                  INSERT INTO alumnodesercion
(cuenta,causa_ingreso,pln_dgae,plan,creditos,periodos,primerperiodo,ultimoperiodo,deserto)
VALUES(cuenta_,causaingreso_,plan_temp,plan_,creditos_alumno,periodos_,primerperiodo_,ultimoperiodo_,deserto_);
        END LOOP ciclo_fetch_cuentas;
      CLOSE fetch_cuentas;
    SET ultimo_renglon=0;
END LOOP ciclo_carreras;
CLOSE fetch_carreras;
END $$ DELIMITER ;
```

• *deserción_null.sql* (procedimiento 2).

```
DELIMITER $$

DROP PROCEDURE IF EXISTS `tesisv2`.`desercion_null` $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `desercion_null`()
BEGIN
/*atributos de la tabla o vista minable alumnodesercion*/
DECLARE cuenta_ VARCHAR(10);
DECLARE indice_ INTEGER;
DECLARE ultimoperiodo_ VARCHAR(5);
DECLARE primerperiodo_ VARCHAR(5);
DECLARE deserto_ VARCHAR(2);

/*Variables temporales*/
DECLARE ultimo_renglon INT DEFAULT 0;

 /*QUERIES*/
/*Obtener los números de cuenta que son NULOS en ultimo periodo (tienen cero créditos)*/
DECLARE nulos CURSOR FOR
SELECT ad.cuenta, MAX(h.periodo), MIN(h.periodo),ad.indice
FROM historias h,alumnodesercion ad
WHERE h.cuenta=ad.cuenta
AND ad.creditos IS NULL
GROUP BY ad.cuenta;

/*Variable de control*/
DECLARE CONTINUE HANDLER FOR NOT FOUND SET ultimo_renglon=1;



/*procedimiento*/
OPEN nulos;
ciclo_nulos:LOOP
        FETCH nulos INTO cuenta_,ultimoperiodo_,primerperiodo_,indice_;
          IF ultimo_renglon=1 THEN
            LEAVE ciclo_nulos;
```

```
          END IF;
                /*determinando si sigue con vida o no, según el último período*/
                IF ultimoperiodo_='20091' THEN
                  SET deserto_='NO';
                ELSE
                  SET deserto_='SI';
                END IF;
                /*Llenando los datos faltantes y correctos a la tabla:*/
                  UPDATE alumnodesercion SET
ultimoperiodo=ultimoperiodo_,primerperiodo=primerperiodo_,creditos=0,deserto=deserto_
                  WHERE indice=indice_;
        END LOOP ciclo_nulos;
      CLOSE nulos;

END $$

DELIMITER ;
```

• *ultimoperiodo_check.sql* (procedimiento 3).

```
DELIMITER $$

DROP PROCEDURE IF EXISTS `tesisv2`.`ultimoperiodo_check` $$
CREATE PROCEDURE `tesisv2`.`ultimoperiodo_check` ()
BEGIN
/*procedimiento que corrige el último semestre cursado*/
/*declarando las variables*/
DECLARE cuenta_ VARCHAR(10);
DECLARE periodomax_ VARCHAR(5);

/*Declarando la variable de control*/
DECLARE ultimo_renglon INT DEFAULT 0;

/*declarando los queries*/
DECLARE maximo CURSOR FOR
SELECT DISTINCT cuenta, MAX(periodo) AS ultimoperiodo FROM historias
GROUP BY cuenta;

/*Variable de control*/
DECLARE CONTINUE HANDLER FOR NOT FOUND SET ultimo_renglon=1;

/*procedimiento*/
/*sustituir todos los últimos semestres a todas las cuentas
DESPUÉS VERIFICAR EN EL CASO DE LOS QUE SÍ DESERTARON PARA CORREGIR A MANO CON QUERY*/
OPEN maximo;
ciclo_maximo:LOOP
  FETCH maximo INTO cuenta_,periodomax_;
    IF ultimo_renglon=1 THEN
      LEAVE ciclo_maximo;
```

```
    END IF;
          /*actualizando la columna ultimoperiodo de alumnodesercion*/
         UPDATE alumnodesercion SET ultimoperiodo=periodomax_
         WHERE cuenta=cuenta_;
        SET ultimo_renglon=0;
END LOOP ciclo_maximo;
CLOSE maximo;

END $$

DELIMITER ;
```

• *Terminomaterias.sql* (procedimiento 4).

```
DELIMITER $$

DROP PROCEDURE IF EXISTS `tesisv2`.`terminomaterias` $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `terminomaterias`()
BEGIN
/*Declaramos las variables a utilizar*/
DECLARE cuenta_ VARCHAR(10);
DECLARE creditos_ INTEGER;
DECLARE deserto_ VARCHAR(2);
DECLARE terminomaterias_ VARCHAR(2);
DECLARE indice_ INTEGER;
DECLARE plan_temp VARCHAR(5);
DECLARE creditos_temp INTEGER;
DECLARE ultimo_renglon INT DEFAULT 0;
DECLARE contador INT DEFAULT 0;
/*Declarando el query por el cual obse tiene las carreras del plan 2006*/
DECLARE fetch_carreras CURSOR FOR
SELECT pln_dgae,creditos FROM carreras_planes;/*todas las carrearas*/
/*declarando el query con el cual obse tiene las cuentas por carrera*/
DECLARE fetch_data CURSOR FOR
SELECT DISTINCT ac.cuenta,ac.creditos,ac.deserto,ac.terminomaterias,ac.indice
FROM alumnodesercion ac, historias h
WHERE ac.cuenta=h.cuenta
AND h.pln_dgae = plan_temp;
 /*claves de la carrera*/
/*Declaramos la bandera que dirá al ciclo cuándo se ha terminado de encontrar datos*/
DECLARE CONTINUE HANDLER FOR NOT FOUND SET ultimo_renglon=1;
/*abrimos el query (en este caso, el cursor).*/
OPEN fetch_carreras;
ciclo_carreras:LOOP
FETCH fetch_carreras INTO plan_temp, creditos_temp;
  IF ultimo_renglon=1 THEN
        LEAVE ciclo_carreras;
  END IF;
```

```
      OPEN fetch_data;
      SET contador=0;
      ciclo_fetch_data:LOOP
        FETCH fetch_data INTO cuenta_,creditos_,deserto_,terminomaterias_,indice_;
          IF ultimo_renglon=1 THEN
            LEAVE ciclo_fetch_data;
          END IF;
          SET contador=contador+1;
        /*lo que se quiere hacer. En este caso se determinará con IF y ELSE*/
        /*si el alumno ya terminó sus materias o no según la deserción
        y el número de créditos (determinados por cada carrera).*/
          IF deserto_='NO' AND creditos_>=(creditos_temp-3) THEN
            SET terminomaterias_='SI';
          ELSEIF deserto_='NO' AND creditos_<(creditos_temp-3) THEN
            SET terminomaterias_='NO';
          ELSEIF deserto_='SI' THEN
            SET terminomaterias_='NO';
          END IF;
          /*Después de clasificar si se terminó o no las materias se procede a actualizar el
          resultado en la tabla de la base de datos*/
          UPDATE alumnodesercion SET terminomaterias=terminomaterias_
          WHERE indice = indice_;
          END LOOP ciclo_fetch_data;
      CLOSE fetch_data;

      SET ultimo_renglon=0;
END LOOP ciclo_carreras;
CLOSE fetch_carreras;
END $$

DELIMITER ;
```

• *duplicidad.sql* (procedimiento 5).

```
 DELIMITER $$

DROP PROCEDURE IF EXISTS `tesisv2`.`duplicidad` $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `duplicidad`()
BEGIN

DECLARE cuentas VARCHAR(10);
```

```
DECLARE ultimo_renglon INT DEFAULT 0;

DECLARE cuentas_duplicadas CURSOR FOR
SELECT cuenta
FROM alumnodesercion
GROUP BY cuenta
HAVING ( COUNT(cuenta) > 1 );

DECLARE CONTINUE HANDLER FOR NOT FOUND SET ultimo_renglon=1;
OPEN cuentas_duplicadas;
ciclo_cuentas:LOOP
FETCH cuentas_duplicadas INTO cuentas;
IF ultimo_renglon=1 THEN
  LEAVE ciclo_cuentas;
END IF;
UPDATE alumnodesercion SET duplicidad=1
WHERE cuenta=cuentas;
END LOOP ciclo_cuentas;
CLOSE cuentas_duplicadas;

END $$

DELIMITER ;
```

• *eliminar_duplicados.sql* (procedimiento 6).

```
DELIMITER $$

DROP PROCEDURE IF EXISTS `tesisv2`.`eliminar_duplicados` $$
CREATE PROCEDURE `tesisv2`.`eliminar_duplicados` ()
BEGIN
/*procedimiento que sirve para eliminar las cuentas
duplicadas dejando sólo una (la que tenga más registros
del plan dgae en cuestión*/

/*Declarando las variables*/
DECLARE cuenta_ VARCHAR(10);
DECLARE ocurrencias_ INT;
DECLARE plndgae_ VARCHAR(4);

/*Declarando la variable de control*/
DECLARE ultimo_renglon INT DEFAULT 0;

/*Declarando los queries*/
/*obteniendo las cuentas*/
DECLARE cuentas CURSOR FOR
SELECT DISTINCT cuenta FROM alumnodesercion
```

```
WHERE duplicidad<>0
ORDER BY cuenta;

/*parac cada número de cuenta obtener los diferentes planes
dgae que tiene registrados y contar cuántas ocurrencias hay*/
DECLARE contar CURSOR FOR
SELECT DISTINCT pln_dgae, COUNT(pln_dgae) AS ocurrencias FROM historias
WHERE cuenta=cuenta_
GROUP BY pln_dgae
ORDER BY ocurrencias DESC LIMIT 1;

/*Variable de control*/
DECLARE CONTINUE HANDLER FOR NOT FOUND SET ultimo_renglon=1;

/*procedimiento*/
OPEN cuentas;
ciclo_cuentas:LOOP
  FETCH cuentas INTO cuenta_;
    IF ultimo_renglon=1 THEN
      LEAVE ciclo_cuentas;
    END IF;
      OPEN contar;
      ciclo_contar:LOOP
        FETCH contar INTO plndgae_,ocurrencias_;
          IF ultimo_renglon=1 THEN
            LEAVE ciclo_contar;
          END IF;
            /*como el primer plan será el que más ocurrencias tenga
            se eliminarán los demás registros*/
            DELETE FROM alumnodesercion WHERE cuenta=cuenta_
            AND pln_dgae NOT IN (plndgae_);

         END LOOP ciclo_contar;
      CLOSE contar;
    SET ultimo_renglon=0;
END LOOP ciclo_cuentas;
CLOSE cuentas;

END $$

DELIMITER ;
```

• *generaciones.sql* (procedimiento 7).

```
DELIMITER $$

DROP PROCEDURE IF EXISTS `tesisv2`.`generaciones` $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `generaciones`()
```

```
BEGIN
/*procedimiento para obtener las generaciones
de los alumnos*/
/*Declarando las variables*/
DECLARE primerperiodo_ VARCHAR(5);
DECLARE generacion_ VARCHAR(4);
DECLARE cuenta_ VARCHAR(9);
/*declarando la variable de control*/
DECLARE ultimo_renglon INT DEFAULT 0;

/*Declarando los queries*/
DECLARE cuentas CURSOR FOR
SELECT cuenta,primerperiodo FROM alumnodesercion;

/*Declarando la variable de control*/
DECLARE CONTINUE HANDLER FOR NOT FOUND SET ultimo_renglon=1;

OPEN cuentas;
ciclo_cuentas:LOOP
  FETCH cuentas INTO cuenta_, primerperiodo_;
    IF ultimo_renglon=1 THEN
      LEAVE ciclo_cuentas;
    END IF;
    IF primerperiodo_='19831' THEN
      SET generacion_='1983';
    ELSEIF primerperiodo_='19862' THEN
      SET generacion_='1986';
    ELSEIF primerperiodo_='19901' THEN
      SET generacion_='1990';
    ELSEIF primerperiodo_='19931' THEN
      SET generacion_='1993';
    ELSEIF primerperiodo_='19932' THEN
      SET generacion_='1993';
    ELSEIF primerperiodo_='19941' THEN
      SET generacion_='1994';
    ELSEIF primerperiodo_='19942' THEN
      SET generacion_='1994';
    ELSEIF primerperiodo_='19951' THEN
      SET generacion_='1995';
    ELSEIF primerperiodo_='19952' THEN
      SET generacion_='1995';
    ELSEIF primerperiodo_='19961' THEN
      SET generacion_='1996';
    ELSEIF primerperiodo_='19962' THEN
      SET generacion_='1996';
    ELSEIF primerperiodo_='19971' THEN
      SET generacion_='1997';
    ELSEIF primerperiodo_='19972' THEN
      SET generacion_='1997';
    ELSEIF primerperiodo_='19981' THEN
```

```
  SET generacion_='1998';
ELSEIF primerperiodo_='19982' THEN
  SET generacion_='1998';
ELSEIF primerperiodo_='19991' THEN
  SET generacion_='1999';
ELSEIF primerperiodo_='19992' THEN
  SET generacion_='1999';
ELSEIF primerperiodo_='20001' THEN
  SET generacion_='2000';
ELSEIF primerperiodo_='20002' THEN
  SET generacion_='2000';
ELSEIF primerperiodo_='20003' THEN
  SET generacion_='2000';
ELSEIF primerperiodo_='20011' THEN
  SET generacion_='2001';
ELSEIF primerperiodo_='20012' THEN
  SET generacion_='2001';
ELSEIF primerperiodo_='20021' THEN
  SET generacion_='2002';
ELSEIF primerperiodo_='20022' THEN
  SET generacion_='2002';
ELSEIF primerperiodo_='20031' THEN
  SET generacion_='2003';
ELSEIF primerperiodo_='20032' THEN
  SET generacion_='2003';
ELSEIF primerperiodo_='20041' THEN
  SET generacion_='2004';
ELSEIF primerperiodo_='20042' THEN
  SET generacion_='2004';
ELSEIF primerperiodo_='20051' THEN
  SET generacion_='2005';
ELSEIF primerperiodo_='20052' THEN
  SET generacion_='2005';
ELSEIF primerperiodo_='20061' THEN
  SET generacion_='2006';
ELSEIF primerperiodo_='20062' THEN
  SET generacion_='2006';
ELSEIF primerperiodo_='20071' THEN
  SET generacion_='2007';
ELSEIF primerperiodo_='20072' THEN
  SET generacion_='2007';
ELSEIF primerperiodo_='20081' THEN
  SET generacion_='2008';
ELSEIF primerperiodo_='20082' THEN
  SET generacion_='2008';
ELSEIF primerperiodo_='20091' THEN
  SET generacion_='2009';
END IF;
    /*llenando la columna generación*/
  UPDATE alumnodesercion SET generacion=generacion_
```

```
      WHERE cuenta=cuenta_;
END LOOP ciclo_cuentas;
CLOSE cuentas;
END $$

DELIMITER ;
```

• *cuantasreprobadas.sql* (procedimiento 8).

```
DELIMITER $$

DROP PROCEDURE IF EXISTS `tesisv2`.`cuantasreprobadas` $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `cuantasreprobadas`()
BEGIN
/*Procedimiento para contar cuántas veces en TOTAL se han reprobado
materias a lo largo de la carrera*/
/*Declarando las variables*/
DECLARE hareprobado_ INTEGER;
DECLARE cuenta_ VARCHAR(10);
/*Declarando la variable de control*/
DECLARE ultimo_renglon INT DEFAULT 0;
/*Declarando los queries:*/
/*Seleccionar los números de cuenta de la tabla historias*/
DECLARE cuentas CURSOR FOR
SELECT DISTINCT cuenta FROM alumnodesercion;
/*WHERE hareprobado IS NULL;/*modifiqué aquí*/
/*Para cada número de cuenta, contar cuántas materias
 ha aprobado el alumno*/
DECLARE veces CURSOR FOR
SELECT COUNT(asignatura) AS 'reprobadas'
FROM historias
WHERE cuenta=cuenta_
AND aprobo='NO';

/*Variable de control*/
DECLARE CONTINUE HANDLER FOR NOT FOUND SET ultimo_renglon=1;

/*procedimiento*/
OPEN cuentas;
ciclo_cuentas:LOOP
  FETCH cuentas INTO cuenta_;
    IF ultimo_renglon=1 THEN
      LEAVE ciclo_cuentas;
    END IF;
      OPEN veces;
      ciclo_veces:LOOP
        FETCH veces INTO hareprobado_;
          IF ultimo_renglon=1 THEN
```

```
            LEAVE ciclo_veces;
          END IF;
      END LOOP ciclo_veces;
       CLOSE veces;
            /*Llenando la columna vecesreprobada de alumnodesercion*/
          UPDATE alumnodesercion SET hareprobado=hareprobado_
          WHERE cuenta=cuenta_;
        SET ultimo_renglon=0;
END LOOP ciclo_cuentas;
CLOSE cuentas;
END $$
DELIMITER ;
```

• *cuantasaprobadas.sql* (procedimiento 9).

```
DELIMITER $$

DROP PROCEDURE IF EXISTS `tesisv2`.`cuantasaprobadas` $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `cuantasaprobadas`()
BEGIN
/*Procedimiento para contar cuántas veces en TOTAL se han aprobado
materias a lo largo de la carrera*/
/*Declarando las variables*/
DECLARE haaprobado_ INTEGER;
DECLARE cuenta_ VARCHAR(10);
/*Declarando la variable de control*/
DECLARE ultimo_renglon INT DEFAULT 0;

/*Declarando los queries:*/
/*Seleccionar los números de cuenta de la tabla historias*/
DECLARE cuentas CURSOR FOR
SELECT DISTINCT cuenta FROM alumnodesercion;
/*WHERE haaprobado IS NULL; /*Aquí modifiqué*/
/*Para cada número de cuenta, contar cuántas materias
 ha aprobado el alumno*/
DECLARE veces CURSOR FOR
SELECT COUNT(DISTINCT asignatura) AS 'aprobadas'
FROM historias
WHERE cuenta=cuenta_
AND aprobo='SI';
/*Variable de control*/
DECLARE CONTINUE HANDLER FOR NOT FOUND SET ultimo_renglon=1;
/*procedimiento*/
OPEN cuentas;
ciclo_cuentas:LOOP
  FETCH cuentas INTO cuenta_;
    IF ultimo_renglon=1 THEN
```

```
      LEAVE ciclo_cuentas;
    END IF;
      OPEN veces;
      ciclo_veces:LOOP
        FETCH veces INTO haaprobado_;
          IF ultimo_renglon=1 THEN
            LEAVE ciclo_veces;
          END IF;
      END LOOP ciclo_veces;
       CLOSE veces;
            /*Llenando la columna vecesreprobada de alumnodesercion*/
          UPDATE alumnodesercion SET haaprobado=haaprobado_
          WHERE cuenta=cuenta_;
        SET ultimo_renglon=0;
END LOOP ciclo_cuentas;
CLOSE cuentas;
END $$

DELIMITER ;
```

- *promedio_termino.sql* (procedimiento 10)

```
DELIMITER $$

DROP PROCEDURE IF EXISTS `tesisv2`.`promedio_termino` $$
CREATE PROCEDURE `tesisv2`.`promedio_termino` ()
BEGIN
/*Este procedimiento determina el promedio de aquellos que terminaron
todas sus materias. Se aconseja crear primero la columna promedio float*/
/*Declarando las variables*/
DECLARE cuenta_ VARCHAR(10);
DECLARE promedio_ FLOAT;

/*Declarando la variable de control*/
DECLARE ultimo_renglon INT DEFAULT 0;

/*Declarando los queries:*/
/*Seleccionar los números de cuenta de aquellos que terminaron todas
sus materias*/
DECLARE cuentas CURSOR FOR
SELECT DISTINCT cuenta FROM alumnodesercion
WHERE terminomaterias='SI';
/*Para cada una de estas cuentas, obtener el promedio*/
DECLARE promedio CURSOR FOR
SELECT AVG(calificacion) AS promedio FROM historias
WHERE cuenta=cuenta_
AND calificacion NOT IN ('NP','AC','RE')
AND aprobo='SI';
```

```
/*Variable de control*/
DECLARE CONTINUE HANDLER FOR NOT FOUND SET ultimo_renglon=1;

/*procedimiento*/
OPEN cuentas;
ciclo_cuentas:LOOP
  FETCH cuentas INTO cuenta_;
    IF ultimo_renglon=1 THEN
      LEAVE ciclo_cuentas;
    END IF;
      OPEN promedio;
      ciclo_promedio:LOOP
        FETCH promedio INTO promedio_;
          IF ultimo_renglon=1 THEN
            LEAVE ciclo_promedio;
          END IF;
          /*Llenando la columna promedio*/
          UPDATE alumnodesercion SET promedio=promedio_
          WHERE cuenta=cuenta_;
      END LOOP ciclo_promedio;
      CLOSE promedio;
      SET ultimo_renglon=0;
END LOOP ciclo_cuentas;
CLOSE cuentas;
END $$

DELIMITER ;
```

- *promedio_des_cursan.sql* (procedimiento 11).

```
DELIMITER $$

DROP PROCEDURE IF EXISTS `tesisv2`.`promedio_des_cursan` $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `promedio_des_cursan`()
BEGIN
/*este procedimiento determina el promedio
para aquellos que ya desertaron o los que están cursando*/

/*Pasos
1) Seleccionar los números de cuenta
2) seleccionar distintamente la materia
3)para cada materia seleccionar aquel registro cuyo
semestre sea el más reciente.
Ahora de esas materias, obtener el promedio sin
tomar en cuenta las calificaciones nominales,
sólo las numéricas
4)actualizar los promedios obtenidos*/
```

```
/*Declarando las variables*/
DECLARE cuenta_ VARCHAR(10);
DECLARE promedio_ FLOAT(3,2);
DECLARE clave_asignatura_ VARCHAR(4);
DECLARE periodo_max_ VARCHAR(5);
DECLARE asignatura_ VARCHAR(5);
DECLARE calificacion_ INTEGER;
DECLARE calif INTEGER;
DECLARE contador INTEGER;
/*Declarando la variable de control*/
DECLARE ultimo_renglon INT DEFAULT 0;

/*Declarando los queries:*/
/*1) Seleccionar los números de cuenta*/
DECLARE cuentas CURSOR FOR
SELECT DISTINCT cuenta FROM alumnodesercion
WHERE terminomaterias='NO'
AND creditos>0;

/*2)Seleccionar sus materias distintamente
AQUI PUEDE SALIR 0 ROWS FETCHED*/
DECLARE materias CURSOR FOR
SELECT DISTINCT asignatura, MAX(periodo) AS periodo, MAX(calificacion) AS calificacion FROM
historias
WHERE cuenta=cuenta_
AND calificacion NOT IN ('NP','AC','RE')
GROUP BY asignatura;

/*Variable de control*/
DECLARE CONTINUE HANDLER FOR NOT FOUND SET ultimo_renglon=1;

/*procedimiento*/
OPEN cuentas;
ciclo_cuentas:LOOP
  FETCH cuentas INTO cuenta_;
    IF ultimo_renglon=1 THEN
      LEAVE ciclo_cuentas;
    END IF;
      SET calif=0;
      SET contador=0;
      OPEN materias;
      ciclo_materias:LOOP
        FETCH materias INTO asignatura_, periodo_max_,calificacion_;
          IF ultimo_renglon=1 THEN
            LEAVE ciclo_materias;
          END IF;
              SET calif=calif+calificacion_;
              SET contador=contador+1;
       END LOOP ciclo_materias;
```

```
       CLOSE materias;
      SET promedio_=(calif/contador);
            /*Llenando la columna promedio*/
          UPDATE alumnodesercion SET promedio=promedio_
          WHERE cuenta=cuenta_;
        SET ultimo_renglon=0;
END LOOP ciclo_cuentas;
CLOSE cuentas;
END $$
DELIMITER ;
```

- *promedio_zero_creds.sql* (procedimiento 12).

```
DELIMITER $$

DROP PROCEDURE IF EXISTS `tesisv2`.`promedio_zero_creds` $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `promedio_zero_creds`()
BEGIN
/*Este procedimiento es para calcular el promedio
de aquellos que tienen CERO créditos,
es decir, ninguna materia pasada*/
/*Declarando las variables*/
DECLARE cuenta_ VARCHAR(10);
DECLARE promedio_ FLOAT(3,2);
DECLARE clave_asignatura_ VARCHAR(4);
DECLARE periodo_max_ VARCHAR(5);
DECLARE asignatura_ VARCHAR(5);
DECLARE calificacion_ INTEGER;
DECLARE calif INTEGER;
DECLARE contador INTEGER;
/*Declarando la variable de control*/
DECLARE ultimo_renglon INT DEFAULT 0;

/*Declarando los queries:*/
DECLARE cuentas CURSOR FOR
SELECT DISTINCT cuenta FROM alumnodesercion
WHERE terminomaterias='NO'
AND creditos=0;

/*2)Seleccionar sus materias distintamente
AQUI PUEDE SALIR 0 ROWS FETCHED*/
DECLARE materias CURSOR FOR
SELECT DISTINCT asignatura, MAX(periodo) AS periodo, MAX(calificacion) AS calificacion FROM
historias
WHERE cuenta=cuenta_
AND calificacion NOT IN ('NP','AC','RE')
GROUP BY asignatura;
```

```
/*Variable de control*/
DECLARE CONTINUE HANDLER FOR NOT FOUND SET ultimo_renglon=1;

/*procedimiento*/
OPEN cuentas;
ciclo_cuentas:LOOP
  FETCH cuentas INTO cuenta_;
    IF ultimo_renglon=1 THEN
      LEAVE ciclo_cuentas;
    END IF;
      SET calif=0;
      SET contador=0;
      OPEN materias;
      ciclo_materias:LOOP
        FETCH materias INTO asignatura_, periodo_max_,calificacion_;
          IF ultimo_renglon=1 THEN
            LEAVE ciclo_materias;
          END IF;
              SET calif=calif+calificacion_;
              SET contador=contador+1;
       END LOOP ciclo_materias;
       CLOSE materias;
       SET promedio_=(calif/contador);
           IF promedio_ IS NULL THEN
              SET promedio_=0;
           END IF;
             /*Llenando la columna promedio*/
          UPDATE alumnodesercion SET promedio=promedio_
          WHERE cuenta=cuenta_;
        SET ultimo_renglon=0;
END LOOP ciclo_cuentas;
CLOSE cuentas;

END $$

DELIMITER ;
```

• *acertod1.sql*

```
DELIMITER $$

DROP PROCEDURE IF EXISTS `tesisv2`.`acertod1` $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `acertod1`()
BEGIN

DECLARE cuenta_ VARCHAR(10);
DECLARE p_deserto_ VARCHAR(2);
```

```
DECLARE ad_deserto_ VARCHAR(2);
DECLARE acerto_ VARCHAR(2);
DECLARE indice_ INTEGER;

DECLARE ultimo_renglon INT DEFAULT 0;


DECLARE cuentas CURSOR FOR
SELECT p.indice,p.cuenta,p.desertara,ad.deserto FROM predictedd1d p,alumnodesercion ad
WHERE p.cuenta=ad.cuenta;


OPEN cuentas;
ciclo_cuentas:LOOP
  FETCH cuentas INTO indice_,cuenta_,p_deserto_,ad_deserto_;
    IF ultimo_renglon=1 THEN
      LEAVE ciclo_cuentas;
    END IF;
      IF ad_deserto_=p_deserto_ THEN
         SET acerto_='SI';
      ELSE
         SET acerto_='NO';
      END IF;

          UPDATE predictedd1d SET acerto=acerto_
          WHERE indice=indice_;
END LOOP ciclo_cuentas;
CLOSE cuentas;
END $$

DELIMITER ;
```

• *acertod2.sql*

```
DELIMITER $$

DROP PROCEDURE IF EXISTS `tesisv2`.`acertod2` $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `acertod2`()
BEGIN

DECLARE cuenta_ VARCHAR(10);
DECLARE p_deserto_ VARCHAR(2);
DECLARE ad_deserto_ VARCHAR(2);
DECLARE acerto_ VARCHAR(2);
DECLARE indice_ INTEGER;

DECLARE ultimo_renglon INT DEFAULT 0;
```

```
/*verificando si terminomaterias*/
DECLARE cuentas CURSOR FOR
SELECT p.indice,p.cuenta,p.terminara,ad.terminomaterias FROM predictedd2t p,alumnodesercion ad
WHERE p.cuenta=ad.cuenta;


OPEN cuentas;
ciclo_cuentas:LOOP
  FETCH cuentas INTO indice_,cuenta_,p_deserto_,ad_deserto_;
    IF ultimo_renglon=1 THEN
      LEAVE ciclo_cuentas;
    END IF;
      IF ad_deserto_=p_deserto_ THEN
         SET acerto_='SI';
      ELSE
         SET acerto_='NO';
      END IF;

          UPDATE predictedd2t SET acerto=acerto_
          WHERE indice=indice_;
END LOOP ciclo_cuentas;
CLOSE cuentas;
END $$

DELIMITER ;
```

Código del árbol de decisión *random forest* o bósque aleatorio 1.

Tree 1

```
promedio <= 7: SI {SI=5800, NO=2978}
promedio > 7
|   hareprobado <= 21
|   |   haaprobado <= 3
|   |   |   hareprobado <= 3.500
|   |   |   |   haaprobado <= 2
|   |   |   |   |   promedio <= 7.835
|   |   |   |   |   |   promedio <= 7.415: NO {SI=0, NO=9}
|   |   |   |   |   |   promedio > 7.415
|   |   |   |   |   |   |   promedio <= 7.500: SI {SI=3, NO=2}
|   |   |   |   |   |   |   promedio > 7.500: NO {SI=1, NO=6}
|   |   |   |   |   promedio > 7.835: SI {SI=46, NO=12}
|   |   |   |   haaprobado > 2
|   |   |   |   |   hareprobado <= 2
|   |   |   |   |   |   promedio <= 7.900
|   |   |   |   |   |   |   promedio <= 7.290: NO {SI=0, NO=44}
|   |   |   |   |   |   |   promedio > 7.290
|   |   |   |   |   |   |   |   promedio <= 7.330: SI {SI=2, NO=2}
|   |   |   |   |   |   |   |   promedio > 7.330: NO {SI=1, NO=56}
|   |   |   |   |   |   promedio > 7.900: NO {SI=6, NO=30}
|   |   |   |   |   hareprobado > 2: SI {SI=5, NO=0}
|   |   |   hareprobado > 3.500
|   |   |   |   hareprobado <= 4.500
|   |   |   |   |   promedio <= 7.550: NO {SI=10, NO=18}
|   |   |   |   |   promedio > 7.550: SI {SI=36, NO=15}
|   |   |   |   hareprobado > 4.500
|   |   |   |   |   promedio <= 7.450
|   |   |   |   |   |   promedio <= 7.330: SI {SI=28, NO=12}
```

```
|   |   |   |   |   |   promedio > 7.330: NO {SI=1, NO=4}
|   |   |   |   |   promedio > 7.450: SI {SI=165, NO=31}
|   |   haaprobado > 3: NO {SI=1548, NO=9390}
|   hareprobado > 21
|   |   terminomaterias = NO
|   |   |   hareprobado <= 25.500
|   |   |   |   promedio <= 7.345
|   |   |   |   |   promedio <= 7.290
|   |   |   |   |   |   promedio <= 7.250
|   |   |   |   |   |   |   promedio <= 7.240: SI {SI=51, NO=50}
|   |   |   |   |   |   |   promedio > 7.240: NO {SI=1, NO=5}
|   |   |   |   |   |   promedio > 7.250: SI {SI=13, NO=7}
|   |   |   |   |   promedio > 7.290: NO {SI=4, NO=10}
|   |   |   |   promedio > 7.345
|   |   |   |   |   promedio <= 8.020
|   |   |   |   |   |   promedio <= 7.370: SI {SI=13, NO=6}
|   |   |   |   |   |   promedio > 7.370
|   |   |   |   |   |   |   promedio <= 7.620: SI {SI=55, NO=31}
|   |   |   |   |   |   |   promedio > 7.620
|   |   |   |   |   |   |   |   promedio <= 7.700: NO {SI=13, NO=18}
|   |   |   |   |   |   |   |   promedio > 7.700: SI {SI=41, NO=23}
|   |   |   |   |   promedio > 8.020: SI {SI=19, NO=5}
|   |   |   hareprobado > 25.500
|   |   |   |   hareprobado <= 37
|   |   |   |   |   hareprobado <= 29.500
|   |   |   |   |   |   promedio <= 7.510
|   |   |   |   |   |   |   promedio <= 7.465: SI {SI=91, NO=29}
|   |   |   |   |   |   |   promedio > 7.465
|   |   |   |   |   |   |   |   promedio <= 7.475: NO {SI=0, NO=3}
|   |   |   |   |   |   |   |   promedio > 7.475: SI {SI=5, NO=4}
|   |   |   |   |   |   promedio > 7.510
|   |   |   |   |   |   |   promedio <= 7.960: SI {SI=68, NO=8}
|   |   |   |   |   |   |   promedio > 7.960
|   |   |   |   |   |   |   |   promedio <= 8.010: NO {SI=3, NO=4}
|   |   |   |   |   |   |   |   promedio > 8.010: SI {SI=17, NO=3}
|   |   |   |   |   hareprobado > 29.500: SI {SI=303, NO=48}
|   |   |   |   hareprobado > 37
|   |   |   |   |   hareprobado <= 42.500: SI {SI=171, NO=10}
|   |   |   |   |   hareprobado > 42.500
|   |   |   |   |   |   promedio <= 8.075: SI {SI=423, NO=4}
|   |   |   |   |   |   promedio > 8.075
|   |   |   |   |   |   |   promedio <= 8.090: NO {SI=1, NO=1}
|   |   |   |   |   |   |   promedio > 8.090: SI {SI=20, NO=0}
|   |   terminomaterias = SI: NO {SI=0, NO=618}
```

Tree 2

```
hareprobado <= 4: NO {SI=720, NO=5254}
hareprobado > 4
|   periodos <= 3: SI {SI=2743, NO=1280}
|   periodos > 3
|   |   hareprobado <= 20.500
|   |   |   hareprobado <= 10.500: NO {SI=637, NO=2406}
|   |   |   hareprobado > 10.500
|   |   |   |   periodos <= 5: SI {SI=772, NO=729}
|   |   |   |   periodos > 5
|   |   |   |   |   periodos <= 7: NO {SI=569, NO=612}
|   |   |   |   |   periodos > 7
|   |   |   |   |   |   hareprobado <= 17.500
|   |   |   |   |   |   |   hareprobado <= 12
|   |   |   |   |   |   |   |   periodos <= 15: NO {SI=91, NO=415}
|   |   |   |   |   |   |   |   periodos > 15: SI {SI=13, NO=8}
|   |   |   |   |   |   |   hareprobado > 12: NO {SI=350, NO=893}
|   |   |   |   |   |   hareprobado > 17.500: NO {SI=270, NO=437}
|   |   hareprobado > 20.500
|   |   |   hareprobado <= 29
|   |   |   |   periodos <= 7: SI {SI=245, NO=239}
|   |   |   |   periodos > 7
|   |   |   |   |   hareprobado <= 26
|   |   |   |   |   |   periodos <= 8: SI {SI=128, NO=67}
|   |   |   |   |   |   periodos > 8
|   |   |   |   |   |   |   periodos <= 9: SI {SI=94, NO=58}
```

```
|   |   |   |   |   |   |   periodos > 9
|   |   |   |   |   |   |   |   periodos <= 10: SI {SI=86, NO=58}
|   |   |   |   |   |   |   |   periodos > 10: NO {SI=296, NO=336}
|   |   |   |   |   hareprobado > 26
|   |   |   |   |   |   periodos <= 12: SI {SI=187, NO=67}
|   |   |   |   |   |   periodos > 12
|   |   |   |   |   |   |   periodos <= 18: SI {SI=98, NO=91}
|   |   |   |   |   |   |   periodos > 18: NO {SI=11, NO=21}
|   |   |   hareprobado > 29
|   |   |   |   hareprobado <= 40.500
|   |   |   |   |   periodos <= 17: SI {SI=656, NO=230}
|   |   |   |   |   periodos > 17
|   |   |   |   |   |   hareprobado <= 35
|   |   |   |   |   |   |   hareprobado <= 33
|   |   |   |   |   |   |   |   periodos <= 21: SI {SI=37, NO=25}
|   |   |   |   |   |   |   |   periodos > 21: NO {SI=5, NO=10}
|   |   |   |   |   |   |   hareprobado > 33: NO {SI=13, NO=32}
|   |   |   |   |   |   hareprobado > 35: SI {SI=66, NO=40}
|   |   |   |   hareprobado > 40.500
|   |   |   |   |   periodos <= 19
|   |   |   |   |   |   periodos <= 9: NO {SI=4, NO=5}
|   |   |   |   |   |   periodos > 9: SI {SI=545, NO=79}
|   |   |   |   |   periodos > 19
|   |   |   |   |   |   hareprobado <= 48
|   |   |   |   |   |   |   hareprobado <= 43
|   |   |   |   |   |   |   |   periodos <= 21: SI {SI=11, NO=6}
|   |   |   |   |   |   |   |   periodos > 21: NO {SI=8, NO=11}
|   |   |   |   |   |   |   hareprobado > 43: SI {SI=47, NO=22}
|   |   |   |   |   |   hareprobado > 48
|   |   |   |   |   |   |   hareprobado <= 62
|   |   |   |   |   |   |   |   periodos <= 26: SI {SI=143, NO=37}
|   |   |   |   |   |   |   |   periodos > 26: NO {SI=6, NO=8}
|   |   |   |   |   |   |   hareprobado > 62: SI {SI=113, NO=20}
```

RuleModel 1

```
if plan <= 1994 and terminomaterias = NO then SI  (7066 / 191)
if plan <= 1994 and terminomaterias = SI then NO  (0 / 1738)
if plan > 1994 and generacion <= 2001 and terminomaterias = NO
then SI  (437 / 8)
if plan > 1994 and generacion <= 2001 and terminomaterias = SI then NO
(0 / 134)
if plan > 1994 and generacion > 2001 and periodos <= 14.500 and
creditos <= 0 and generacion <= 2,008.500 then SI  (278 / 49)
if plan > 1994 and generacion > 2001 and periodos <= 14.500 and
creditos <= 0 and generacion > 2,008.500 then NO  (0 / 116)
if plan > 1994 and generacion > 2001 and periodos <= 14.500 and
creditos > 0 and generacion <= 2002 and terminomaterias = NO and
periodos <= 6.500 then SI  (16 / 0)
if plan > 1994 and generacion > 2001 and periodos <= 14.500 and
creditos > 0 and generacion <= 2002 and terminomaterias = NO and
periodos > 6.500 then NO  (181 / 217)
if plan > 1994 and generacion > 2001 and periodos <= 14.500 and
creditos > 0 and generacion <= 2002 and terminomaterias = SI then NO
(0 / 198)
if plan > 1994 and generacion > 2001 and periodos <= 14.500 and
creditos > 0 and generacion > 2002 and haaprobado <= 1 and generacion
<= 2008 and periodos <= 2 then SI  (91 / 5)
if plan > 1994 and generacion > 2001 and periodos <= 14.500 and
creditos > 0 and generacion > 2002 and haaprobado <= 1 and generacion
<= 2008 and periodos > 2 and generacion <= 2,007.500 then SI  (72 /
43)
if plan > 1994 and generacion > 2001 and periodos <= 14.500 and
```

```
creditos > 0 and generacion > 2002 and haaprobado <= 1 and generacion
<= 2008 and periodos > 2 and generacion > 2,007.500 then NO  (8 / 74)
if plan > 1994 and generacion > 2001 and periodos <= 14.500 and
creditos > 0 and generacion > 2002 and haaprobado <= 1 and generacion
> 2008 then NO  (0 / 246)
if plan > 1994 and generacion > 2001 and periodos <= 14.500 and
creditos > 0 and generacion > 2002 and haaprobado > 1 then NO  (804 /
10477)
if plan > 1994 and generacion > 2001 and periodos > 14.500 then SI
(11 / 0)

correct: 21171 out of 22460 training examples.
```