TESIS DE LICENCIATURA EN CIENCIAS DE LA COMPUTACIÓN

ÁRBOLES DE DECISIÓN, UNA TÉCNICA DE DATA MINING

DESDE UNA PERSPECTIVA INFORMATICA Y ESTADISTICA

Tesista
Noemí Lorena Matsudo
L.U.:439/91

Directora
Dra. Ana Silvia Haedo
Profesora adjunta
Departamento de Computación

Universidad Nacional de Buenos Aires
Facultad de Ciencias exactas y Naturales
Departamento de Computación

A mis padres, por su esfuerzo
A la Dra. Ana Haedo, por aceptar
la dirección de ésta Tesis

Agradecimientos

A la Dra. María Carolina Monard profesora de la Universidad de Saõ Paulo, Campus Saõ Carlo, y a su alumna Claudia Aparecida Martins, por el asesoramiento brindado en el aprendizaje de máquina supervisado.

A la Lic. Clyde Trabuchi y al Lic. Augusto Hosz, miembros del I.N.D.E.C. *(Instituto Nacional de Estadística y Censo)*, por brindarme toda la información necesaria sobre la E.P.H. *(Encuesta Permanente de Hogares)*.

Contenido

Resumen

Data mining combina métodos y herramientas de al menos tres áreas: aprendizaje de máquina (A.M.), estadística y base de datos.

La relación entre aprendizaje de máquina, estadística y *data mining* es un poco obvia, las tres áreas apuntan a localizar regularidades, patrones o conceptos en los datos.

El interés de este trabajo fue analizar y aplicar una de las técnicas de *data mining*, árboles de decisión, a través de dos enfoques diferentes.

Se estudiaron los árboles de decisión que construye el aprendizaje de máquina, así como los obtenidos mediante un nuevo enfoque estadístico que utiliza *datos simbólicos*.

En aprendizaje de máquina la tarea de un inductor (o programa de aprendizaje), es generar una buena clasificación a partir de un conjunto de datos ya clasificados, para luego clasificar nuevos datos no etiquetados. Uno de los programas de aprendizaje, es el árbol de decisión, éste construye un árbol a partir de un conjunto de datos clasificados y genera, *hojas* que indican clases y *nodos* que especifican el test que se llevará a cabo sobre un valor de atributo, con una rama y un subárbol para cada uno de los resultados posibles del test.

El análisis de *datos simbólicos* intenta resumir los datos de una gran base de acuerdo a algún concepto oculto para extraer nuevos conocimientos. Estos conceptos, solamente pueden ser descriptos por tipos más complejos de datos que se llaman *datos simbólicos*, cuyos valores pueden ser: subconjunto de categorías, intervalos (ej. mínimo y máximo de ventas diarias) o distribuciones de frecuencias (ej. distribuciones del ingreso de diferentes ciudades). Además, este enfoque permite definir reglas y taxonomías entre diferentes variables.

Nosotros, analizamos con profundidad la construcción de éstos árboles de decisión, y utilizamos estos métodos para analizar los datos recolectados por la *Encuesta Permanente de Hogares (E.P.H.)*, que es un programa nacional e intercensal que se desarrolla en el *Instituto Nacional de Estadística y Censo (I.N.D.E.C.)*. Esta encuesta releva información socioeconómica de diferentes lugares del país. Mediante estos métodos tratamos de encontrar patrones o regularidades en los datos, con el fin de establecer las características habitacionales de los diferentes lugares del país. Analizamos las ventajas y desventajas de ambas implementaciones como así también las conclusiones.

Abstract

Data mining combines methods and tools of at least three areas: machine learning, statistic and data bases.

The relationship between machine learning, statistic and data mining is quite obvious, the three areas aim to discover important regularities, patterns or concepts in the data.

The main object of this work was to analyze and apply decision trees, which is one *data mining* 's technique, through two different focuses.

On one hand we studied decision trees built by machine learning, while on the other hand we studied a new statistic focus that use *symbolic data.*

In machine learning the inductor 's (or learning program 's) task, consists in generating a good classification method, from a set of already classified data, that we can apply afterward to new non classified cases. The decision trees' method, builds a tree from a classified set of data generating *leaves* that indicate classes and *nodes* that specify a test applied to attribute values with a branch and a subtree for each possible test result.

On the other hand, the analysis of symbolic data try to summarize these data in terms of the underline concepts in order to extract new knowledge from them. These concepts can only be described by more complex type of data which we call symbolic data it value can be, subsets of categories, intervals (e.g. minimum and maximum daily sales) or frequency distributions (e.g. income distribution in different countries). Beside, it allows to define rules and taxonomies between different variables.

We have deeply analyzed the construction of these decision trees and we used these methods to analyze, data obtained from the *Permanent Surveys 'Home (E.P.H.)* which is a national and intercensus program developed by the *National Institute of Statistic and Census (I.N.D.E.C.).* This survey gives social and economic information from different region of the country. Through these methods we look forward to find patterns or regularities of data that allow to classify the habitants of the different points in the country. Finally, in the last chapter, we show the advantages and disadvantages of both implementations as well the conclusion.

Introducción

La explosión de la disponibilidad de información en la sociedad moderna, lleva a que la mayoría de las organizaciones tengan una gran base de datos conteniendo información accesible y potencialmente rica.

El desarrollo de nuevas técnicas para encontrar la información requerida a partir de una enorme cantidad de datos es uno de los principales desafíos de hoy para el desarrollo de software. El crecimiento del volumen de información es la razón de la dificultad para encontrar el significado de lo que queremos buscar.

Se podría hacer una analogía con la búsqueda de tesoros en las minas "*mining*". En ella, se debe remover gran cantidad de escombros antes de encontrar el oro o diamante. Con una computadora se puede encontrar automáticamente la "*información-diamante*" entre las toneladas de "*datos-escombros*" de una gran base de datos.[ADR/97]

Consideramos entonces al ***data mining*** un área muy atractiva e interesante de investigar.

Objetivo

Analizaremos un conjunto de datos mediante el proceso de *KDD (Knowledge Discovery Data).* Este proceso se utiliza para descubrir conocimiento nuevo en una gran base de datos; como se verá en el *Capítulo I*, el *data mining* es parte de este proceso.

Existen varias técnicas utilizadas en *data mining* para encontrar patrones o relaciones entre los datos; nosotros nos concentraremos en los *árboles de decisión* analizaremos los árboles que se construyen mediante la *clasificación simbólica supervisada* y mediante el *análisis de datos simbólicos*, el primero es un método utilizado en el *Aprendizaje de Máquina* (AM) mientras que el segundo es un nuevo enfoque del *análisis de datos estadístico.* Se compararán los resultados de ambos métodos analizando sus ventajas y desventajas.

Los datos serán analizados utilizando ambos métodos, esos datos fueron recolectados por la *Encuesta Permanente de Hogares* (EPH). Dicha encuesta es un programa nacional e intercensal que se desarrolla en el INDEC desde 1972, conjuntamente con las *Direcciones Provinciales de Estadística* (DPE) desde 1974.

Esta encuesta releva actualmente información socioeconómica de 28 aglomerados urbanos del país, nosotros analizaremos solamente los aglomerados (32 y 33) correspondientes al Gran Buenos Aires del mes de Octubre del '98.

El objetivo general de la EPH consiste en caracterizar a la población:

- desde el punto de vista demográfico.
- por su participación en la producción de bienes y servicios.
- por su participación en la distribución del producto social.

Capítulo I

Descubrir conocimiento a partir de los datos - KDD (Knowledge Discovery Data)

1.1 Concepto del KDD

Existe una confusión acerca del término de *data mining* y *KDD (Knowledge Discovery Data)* algunos autores los ven como sinónimos. En la primera conferencia internacional de *KDD* en Montreal en 1995, se propuso que el término de *KDD* sea empleado para describir todo el proceso de extracción de conocimiento de los datos. En este contexto, conocimiento significa relaciones y patrones entre los elementos de datos. El término de *data mining* sería usado exclusivamente para el paso de descubrimiento del *KDD*.

El *KDD* es "la extracción no trivial de conocimiento, implícito en los datos, previamente desconocido y potencialmente útil". Entonces el conocimiento debe ser nuevo, no obvio y útil. El *KDD* no es una técnica nueva, es un campo multidisciplinario de investigación, donde contribuyen: *Aprendizaje de máquina*, *estadística*, *tecnología de las bases de datos*, *sistemas expertos* y *visualización de los datos* [ADR/97]. Ver Figura 1.

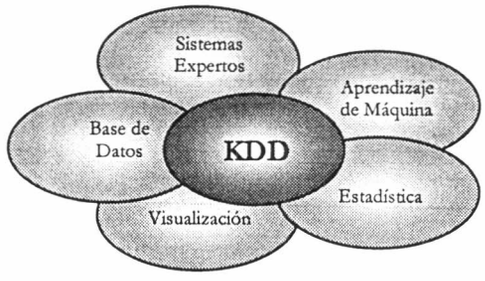


Figura 1: *Data mining* es un campo multidisciplinario

1.2 Proceso de KDD

El proceso de *KDD* es un proceso de soporte de decisión en el cual se buscan patrones de información en los datos.

Este proceso consta de seis pasos:

1. Selección de los datos
2. Limpieza
3. Enriquecimiento
4. Codificado
5. Análisis Exploratorio - *Data mining*
6. Información

El 5to. Paso, *data mining*, es la fase de descubrimiento real y es el punto que analizaremos con mayor detalle. Quizá la presentación de la metodología da la impresión que hay una trayectoria lineal a través del proceso, y no es así, se trata de un proceso dinámico en el que, estando en cualquier paso se puede retroceder una o más fases. Por ejemplo, estando en la fase de codificado o *data mining*, podría considerarse que la fase de limpieza está incompleta, o podrían descubrirse nuevos datos y usar estos para enriquecer el conjunto de datos existente.

1.2.1 Selección de datos

Una vez que se formularon los requerimientos de información, el paso siguiente es coleccionar y seleccionar los datos que se necesitan. No siempre es una tarea fácil reunir esta información en una base de datos centralizada, ya que esto puede llevar a conversiones de bajo nivel.

Por ejemplo, las conversiones de archivos planos a tablas relacionales o de sistemas relacionales a sistemas jerárquicos. También, se puede encontrar que los datos operacionales que se usan en diferentes partes de la organización varía en la calidad: algunos departamentos mantienen bases de datos de alta calidad conteniendo información que es vital para sus operaciones mientras que otros tienen un pequeño conjunto de datos construído especialmente *ad hoc*. Algunas bases de datos se actualizan día a día, y otras tienen información de varios años. Las diferentes bases de datos usadas en varias partes de la organización pueden usar técnicas completamente diferentes para identificar sus registros.

1.2.2 Limpieza

Una vez que se recolectaron los datos, el siguiente paso es la limpieza. Probablemente no se tiene conciencia de la cantidad de suciedad que existe en los datos. Es bueno invertir tiempo en examinar los datos para tener una idea de las posibilidades de extraer información, quizá en la práctica, es difícil si se tienen grandes conjuntos de datos. Si la base de datos es muy grande es aconsejable seleccionar algunos registros al azar y analizarlos para tener una vaga idea de lo que se tiene.

Hay varios tipos de procesos de limpieza, algunos son ejecutados por adelantado mientras que otros sólo se invocan después que la suciedad se encuentra, en la etapa de codificado o descubrimiento (*data mining*).

Un caso importante son los registros duplicados, es necesario eliminarlos, a veces se generan por error de tipeo, cambios de información (por ejemplo domicilio), o porque las personas brindan información errónea deliberadamente. Otro caso interesante son los datos inconsistentes por ejemplo, fechas del tipo 01-01-01.

1.2.3 Enriquecimiento

Una vez efectuada la limpieza de datos, se puede enriquecer la base de datos. En algunos países obtener información adicional es un aspecto comercial, por ejemplo, se puede obtener información adicional sobre varios sujetos (como edad, ingresos, cantidad de créditos, etc.).

1.2.4 Codificado

Si para algunos individuos de la base de datos, falta algún tipo de información, por ejemplo, posesión de auto o casa, es necesario hacer un análisis de las posibles consecuencias antes de tomar la decisión de borrar dichos registros. Si esta información se distribuye aleatoriamente sobre la base de

datos, remover todos los registros con falta de información no afectará el tipo de agrupamiento a encontrar. Por otro lado, es posible que haya alguna conexión casual entre la falta de información de cierto tipo y el individuo, en este caso remover los registros afectarán los tipos de patrones a encontrar.

Además de ésto, la etapa de codificado sirve para dar mayor performance a los datos. A veces la base de datos está demasiado detallada para usarla como entrada en el algoritmo de reconocimiento de patrones. Por ejemplo, la fecha de nacimiento es una información muy detallada, podría guardarse información de clase de edades separadas en rango de diez años. Otro ejemplo es la dirección, quizá solo es necesario el código postal.

1.2.5 Data mining

Data mining combina métodos y herramientas de al menos tres áreas: *aprendizaje de máquina*, *estadística* y *bases de datos*.

La relación entre *aprendizaje de máquina*, *estadística* y *data mining* es un poco obvia, las tres áreas apuntan a localizar regularidades importantes, patrones o conceptos de datos empíricos.

Los métodos de *aprendizaje de máquina* forman parte del *data mining*. *árboles de decisión* o *reglas de inducción* son unos de los componentes de varios algoritmos de *data mining*.

En *estadística* desde hace varios años se investiga el *Análisis Exploratorio de Datos (EDA)*. El *EDA* y el *KDD* tienen muchos puntos y métodos en común.

Nos concentraremos en uno de los métodos de aprendizaje de máquina, llamado *clasificación simbólica supervisada* en los Capítulos II y III. Además investigamos un nuevo enfoque estadístico, que es el *análisis de datos simbólico* en los Capítulos IV y V.

1.2.6 Información

El paso de información combina dos funciones diferentes:

- Análisis de los resultados de algoritmos de reconocimiento de patrón.
- Aplicaciones de los resultados de los algoritmos de reconocimiento de patrón a nuevos datos.

Se puede dar la información usando herramientas de consulta tradicional para base de datos (SQL). Aunque ahora, aparecieron nuevas técnicas de visualización de datos, desde simples diagramas de puntos mostrando diferentes agrupaciones en dos dimensiones hasta complejos entornos interactivos conteniendo información del conjunto de datos.

Capítulo II

A.M. (Aprendizaje de Máquina)

2.1 Aprendizaje de Máquina

El aprendizaje de máquina o (AM) es la automatización de un proceso de aprendizaje, como resultado del aprendizaje, se construyen reglas que están basadas en las observaciones de un entorno.

La inducción es la inferencia de información a partir de los datos y el aprendizaje inductivo es el proceso de construir un modelo donde el entorno o la base de datos es analizada con vista a encontrar patrones, los objetos son agrupados en clases y se definen reglas dentro de esas clases para poder predecir las clases de nuevos objetos.

Existen dos estrategias importantes en el aprendizaje inductivo, el *aprendizaje supervisado* y el *aprendizaje no supervisado.*

En el primero existe un experto que ayuda al sistema a construir un modelo, este experto, define las clases y da ejemplos de cada clase, el sistema tratará de encontrar las características comunes de los ejemplos pertenecientes a una misma clase y poder predecir así la clase de los futuros objetos. Un ejemplo podría ser, el analista de una empresa de riesgo crediticio, que desea saber cuando un cliente es capaz o no de pagar un crédito, el analista puede construir un modelo a partir de sus datos históricos teniendo en él los datos de los clientes que pagaron o no sus créditos, un buen modelo le permitirá entender mejor a sus clientes y predecir en un futuro que clientes pagarán sus créditos.

En el caso del *aprendizaje no supervisado* no se definen las clases de los ejemplos, el sistema observa los casos y reconoce los patrones por si mismo estableciendo las clases, es decir, que el sistema es capaz de inferir las clases a partir de los datos. El ejemplo en este caso sería, la decisión de una compañía de crear una segmentación del mercado de clientes para poder entender y vender mejor, el sistema utilizará entonces una base de datos con información de sus clientes y creará por si solo una segmentación del mercado.

2.2 Aprendizaje de Máquina Supervisado

Nos concentraremos en el *aprendizaje supervisado.* En los sistemas de aprendizaje de máquina supervisado se ingresa un conjunto de instancias de entrenamiento donde cada instancia se describe por un vector de valores y una clase etiquetada. La tarea del algoritmo de inducción es aprender a clasificar correctamente nuevas instancias no etiquetadas. Para clases discretas el problema se conoce como *clasificación* y para clases continuas se llama *regresión.*

Como habíamos mencionado, nuestro interés es encontrar las características comunes de las familias que pertenecen a una misma región o aglomerado, por lo tanto estudiaremos en mayor detalle la *clasificación simbólica supervisada.* El término *simbólica* indica que la clasificación puede ser leída y comprendida por un humano. El término *supervisada,* indica que algún proceso, ha clasificado previamente las instancias en el conjunto de entrenamiento. Finalmente, como se dijo antes, el término *clasificación* denota el hecho de que la clase es discreta.

La Figura 2, muestra la jerarquía de aprendizaje, los nodos sombreados conducen al *aprendizaje de clasificación supervisado.*

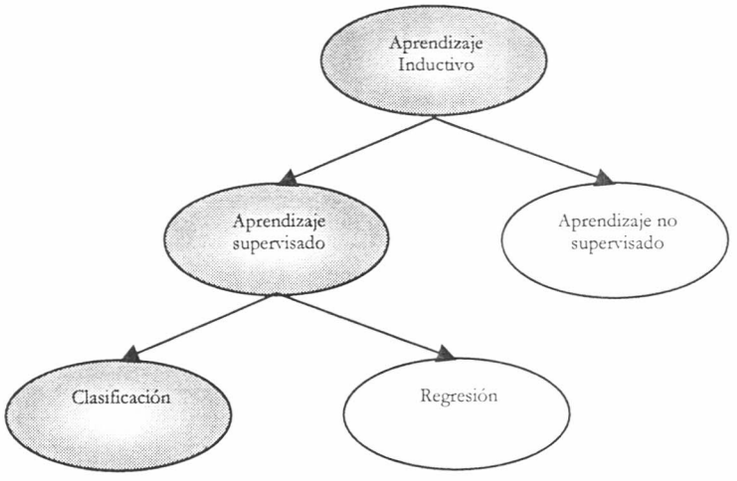


Figura 2: La jerarquía de aprendizaje

2.3 Conceptos básicos en el aprendizaje de máquina supervisado

2.3.1 Características del conjunto de datos

Un conjunto de datos es un conjunto de instancias (también llamadas ejemplos, casos o registros) clasificadas. Es decir, todas las instancias tienen un atributo especial llamado *clase*, el cual describe el fenómeno de interés, o lo que queremos que el algoritmo aprenda.

Un atributo, (a veces llamado campo) describe aspectos de una instancia. Normalmente hay dos tipos de atributos: nominal (ej. color: rojo, verde, azul) y continuo (ej. peso, edad) estos se usan cuando hay un orden sobre los valores.

En el mundo real es común trabajar con datos imperfectos. Los datos imperfectos podrían ser derivados de los procesos que generan, adquieren o transforman datos, a veces también pueden ser clases etiquetadas incorrectamente (por ejemplo, instancias con los mismos valores que pertenecen a clases diferentes). En todos estos casos se puede decir que hay *ruidos* en los datos.

Generalmente, en la mayoría de los inductores hay un valor importante que significa *desconocimiento*. Este valor, debe ser diferente del resto por ejemplo el "0" en los continuos o el "blanco" en los nominales. En muchos inductores este valor se representa por "?".

La mayoría de los inductores asumen que los atributos que describen las instancias son suficientemente *relevantes* para aprender la tarea asignada. Un atributo es *irrelevante* si hay una descripción completa y consistente de las clases a ser aprendidas que no usa este atributo. Es importante elegir atributos con capacidad de predicción. Por ejemplo, atributos con bajo poder de predicción, podrían ser, color de pelo, color de ojos, número de hijos, si la tarea es predecir la enfermedad mientras que, temperatura, examen de espalda, examen de piel, serían atributos de mayor predicción.

Un conjunto de datos está separado en dos conjuntos disjuntos, el conjunto de *entrenamiento* el cual se usa para aprender el concepto y el *conjunto de testeo*, se usa para medir la efectividad del concepto aprendido.

La Tabla 1 muestra un conjunto de datos T con n instancias y m atributos. La fila i indica la i-ésima instancia ($i = 1, 2,...,n$), la columna j indica el j-ésimo atributo ($j = 1, 2,...,m$) y la entrada x_{ij} se refiere al valor de la i-ésima instancia para el j-ésimo atributo.

	X_1	X_2	...	X_m	Y
T_1	x_{11}	x_{12}	...	x_{1m}	y_1
:	:	:	:	:	:
T_n	x_{n1}	x_{n2}	..	x_{nm}	y_n

Tabla 1: Formato de un conjunto de datos

2.3.2 Características del algoritmo de aprendizaje de máquina supervisado

Veamos ahora como trabaja un algoritmo de aprendizaje de máquina supervisado, dado un conjunto de *entrenamiento*, la salida del algoritmo es un clasificador tal que, dada una nueva instancia, éste predice su clase *Y*. Formalmente, una instancia es una par *(X,f(X))* donde *X* es la entrada y *f(X)* es la salida. La tarea de un inductor es, dado un conjunto de instancias, inducir una función *h* que aproxime *f*. En este caso *h* se llama una hipótesis sobre *f*.

Como la función *f* es actualmente desconocida, hay muchas posibles elecciones para *h*, pero sin conocimiento extra, no tenemos forma de elegir entre esas hipótesis. La preferencia de algunas sobre otras, simplemente por una cuestión de consistencia se llama *tendencia*.

Otro factor importante es la varianza de un clasificador, la cual mide cuánto fluctúa un algoritmo de aprendizaje para diferentes *conjuntos de entrenamientos*.

Algunos clasificadores son inestables, en el sentido que pequeñas perturbaciones en el conjunto de *entrenamiento*, o en su construcción, puede resultar en grandes cambios en el clasificador. Los clasificadores inestables se caracterizan por su alta varianza y baja tendencia mientras que los estables tienen baja varianza y alta tendencia.

2.3.3 Reglas

Una forma de expresar los resultados en los algoritmos de aprendizaje de máquina es a través de reglas. Una regla tiene la forma:

$$Si \text{ <complejo> } entonces \text{ <clase} = C_i\text{>}$$

Donde C_i pertenece a un conjunto de posibles valores de *k* clases $\{C_1, C_2, \ldots, C_k\}$. La parte *<complejo>* también se llama condición de la regla y *<clase = Ci>* se llama conclusión de la regla. Las reglas de aprendizaje son usualmente *consistentes* y *completas*.

Un <complejo> es una disyunción de conjunciones de la forma:

$$X_i \, op \, Valor$$

Donde X_i es un atributo, *op* es un operador {=, <, >, *etc.*} y *Valor* es un valor constante del atributo X_i.

Es posible tener una combinación lineal de atributos (para atributos continuos) de la forma

$$C_1 \times X_1 + C_2 \times X_2 + \ldots + C_m \times X_m \, op \, Valor$$

Donde C_i es una constante, X_i es un atributo continuo *op* es un operador y *Valor* es un valor constante.

Las instancias que satisfacen la parte *<complejo>* de la regla, se dice que son *cubiertas* por la regla. Las instancias que satisfacen ambos la *<complejo>* y *<clase* = C_i*>* se llaman *positivamente cubiertas* por la regla. Por otro lado las que satisfacen la parte *<complejo>* pero *<clase* ≠ C_i *>* se dicen *negativamente cubiertas*. Ver Tabla 1.

Los sistemas de aprendizaje por asociación encuentran reglas de implicación conjuntivas o *reglas de asociación* de la forma:

Si <complejo1> entonces <complejo2>

donde no hay atributos comunes entre *complejo1* y *complejo2*. No hay una explícita definición de clase y algunos atributos se pueden usar como parte de la conclusión de la regla. Por ejemplo:

Si X_3 *y* X_5 *entonces* X_1 *y* X_2

Observar que una regla de asociación es una generalización de una regla convencional definida previamente. Los criterios usados para evaluar reglas de asociación se pueden usar para evaluar reglas convencionales (Tabla 2). Los sistemas de asociación convencional encuentran reglas de asociación que satisfacen algún criterio mínimo.

Instancias que satisfacen...	**Son...**
<complejo1>	Cubiertas por la regla
<complejo1> y <complejo2>	Positivamente cubierta por la regla
<complejo1> y no <complejo2>	Negativamente cubierta por la regla

Tabla 2: Definiciones cubiertas por las reglas

Se usan dos criterios para evaluar las reglas, *soporte* y *confianza*. Sea n el número total de instancias de entrenamiento, n_1 el número de instancias que satisfacen *<complejo1>* y n_{12} el número de instancias que satisfacen ambos *<complejo1>* y *<complejo2>*. Entonces se define:

$$Soporte = \frac{n_{12}}{n}$$

$$Confianza = \frac{n_{12}}{n_1}$$

2.3.4 Medidas de error y eficiencia

Una medida muy usada es la tasa de error de un clasificador h denotada por *ce(h)* generalmente la tasa de error se obtiene por la ecuación $ce(h) = 1/n \sum || y_i \neq h(x_i) ||$. Compara cada instancia etiquetada con el valor de la predicción, ||E|| retorna un 1 si E es verdadero y 0 en otro caso. El complemento de la tasa de error, la efectividad del clasificador, se denota por *ca(h) = 1-ce(f)*.

En problemas de regresión, el *predictor de error (pe)* se puede estimar calculando la distancia entre el verdadero valor y el predicho. Generalmente se utiliza el error cuadrático medio *(mse)* y la distancia media absoluta *(mad)*.

Cuando se extrae una hipótesis de los datos es posible que el clasificador sea muy específico para el conjunto de entrenamiento. Como el conjunto de entrenamiento es solamente una muestra de todas las instancias posibles, se pueden generar clasificadores que mejoren la performance sobre estos conjuntos de entrenamientos mientras decrece la performance sobre otras instancias fuera de este

conjunto de entrenamiento. En este caso decimos que el clasificador *sobreajusta* el conjunto de entrenamiento.

Por otro lado es posible que se den pocas instancias representativas al sistema de aprendizaje. Si se generan predictores que mejoran pobremente la performance del conjunto de entrenamiento como también sobre el conjunto de testeo, se dice que el predictor *subajusta* el conjunto de entrenamiento.

Una forma estándar de eliminar el ruido y el *sobreajuste* en árboles de decisión es el podado. Hay básicamente dos aproximaciones al podado (*pruning*).

pre-podado (pre-pruning): significa que durante la generación de hipótesis algunas instancias del conjunto de entrenamiento son ignoradas para que la hipótesis final no clasifique correctamente todas las instancias del entrenamiento.

pos-podado (pos-pruning): significa que primero se genera la hipótesis que explica todo el conjunto de entrenamiento. Luego la hipótesis se generaliza eliminando algunas partes, como la eliminación de ramas del árbol de decisión o algunas condiciones en reglas de inducción.

Una vez inducida una hipótesis se evalúa por su *completitud*, si ésta clasifica todas sus instancias, y consistencia si clasifica correctamente las instancias. Entonces, dada una hipótesis, ésta se puede considerar:

1. *completa y consistente*
2. *incompleta y consistente*
3. *completa e inconsistente*
4. *incompleta e inconsistente*

Una matriz de confusión ofrece una medida de la efectividad del modelo de clasificación. Cada elemento $M(C_i, C_j)$ de esta matriz indica el número de instancias que actualmente pertenece a la verdadera clase Ci, pero fueron predichas como pertenecientes a la clase C_j.

$$M(C_i, C_j) = \sum \| h(x) = C_j \|$$

Clase	**Predecir C_1**	**Predecir C_2**	**...**	**Predecir C_k**
C_1 verdadera	$M(C_1, C_1)$	$M(C_1, C_2)$	...	$M(C_1, C_k)$
C_2 verdadera	$M(C_2, C_1)$	$M(C_2, C_2)$	...	$M(C_2, C_k)$
...	...	...	...	...
C_k verdadera	$M(C_k, C_1)$	$M(C_k, C_2)$	...	$M(C_k, C_k)$

Tabla 3: Matriz de confusión

El número de predicciones correctas para cada clase cae en la diagonal principal $M(C_i,C_j)$ de la matriz. Todos los otros elementos $M(C_i, C_j)$, para $i \neq j$ son el número de instancias mal clasificadas.

Por supuesto, el clasificador ideal debería tener todas las entradas excepto su diagonal igual a cero ya que no debería tener errores.

Dada una regla, una instancia y una clase, hay cuatro posibilidades:

1. La instancia satisface <*complejo*> y esta clase es la misma que la predicha por la conclusión <*clase* = C_i>.
2. La instancia satisface <*complejo*> y esta clase **no** es la misma que la predicha por la conclusión <*clase* = C_i>.

3. La instancia **no** satisface <*complejo*> y esta clase es la misma que la predicha por la conclusión <*clase* = C_i>.
4. La instancia **no** satisface <*complejo*> y esta clase **no** es la misma que la predicha por la conclusión <*clase* = C_i>.

La tasa de error *ce(h)* y su complemento *ca(h)*, son dos de las métricas más usadas para evaluar la performance de los sistemas de aprendizaje.

Si se considera el problema de clasificar dos clases, etiquetadas como '+' y '-', cuando hay solamente dos clases los dos posibles errores se denominan *falso positivo* y *falso negativo*. La Tabla 3 ilustra la matriz de confusión para el problema de clasificación de dos clases donde *Vp* es el nú.mero de ejemplos positivos clasificados correctamente y *Fn* es el número de instancias positivas clasificadas incorrectamente de un total de *n=(Vp + Fn + Fp + Vn)* instancias.

Pueden ocurrir cuatro situaciones:

1. La instancia pertenece a la clase C^+ y es predicha por el clasificador como clase C^+ en este caso esta instancia es *verdadera positiva.*
2. La instancia pertenece a la clase C^- y es predicha por el clasificador como clase C^- en este caso esta instancia es *verdadera negativa.*
3. La instancia pertenece a la clase C^- y es predicha por el clasificador como clase C^+ en este caso esta instancia es *falsa positiva.*
4. La instancia pertenece a la clase C^+ y es predicha por el clasificador como clase C^- en este caso esta instancia es *falsa negativa.*

Clase	Predecir C^+	Predecir C^-	Tasa de error de Clase	Tasa de error Total
	Verdadera positiva	Falsa negativa		
Verd. C^+	Vp	Fn	Fn/(Vp+Fn)	
				(Fp+Fn)/n
	Falsa positiva	Verdadera negativa	Fp/(Fp+Vn)	
Verd. C^-	Fp	Vn		

Tabla 4: Performance en la clasificación de dos clases

$$Valor\ de\ predicción\ C^+ = \frac{V_p}{V_p + F_p}$$

$$Valor\ de\ predicción\ C^- = \frac{V_n}{V_n + F_n}$$

$$Valor\ de\ verdad\ C^+ = \frac{V_p}{V_p + F_n}$$

$$Valor\ de\ verdad\ C^- = \frac{V_n}{F_p + V_n}$$

$$Efectividad = \frac{Vp + Vn}{n}$$

Un resultado común es el desbalanceo de las clases. Por ejemplo, si suponemos la proporción de instancias de cada clase – (C_1, C_2, C_3) = (99%, 0.25%, 0.75%). Un clasificador simple que predice siempre la clase C_1 debería tener una eficiencia del 99%. Esto podría ser indeseable si las clases minoritarias, tienen información importante, por ejemplo, C_1: pacientes normales, C_2: pacientes con enfermedad A, C_3: pacientes con enfermedad B.

Cuando se trabaja con conjuntos de datos desbalanceados es deseable usar una medida de performance además de eficiencia.

2.3.5 Podado *(Pruning)*

Como mencionamos anteriormente, el *pruning* es utilizado para evitar el *sobreajuste* que se produce al construir los árboles de decisión, reduciendo la complejidad del árbol mientras da una mejor performance al árbol original. Si bien existe dos tipos de *pruning*, *prepruning* y *pospruning*, profundizaremos en este punto el *pospruning*.

El *prepruning* suele tener una desventaja, es posible que conjunciones de testeos sean la mejor forma de particionar el árbol, pero cada uno de esos testeos no distingan suficientemente bien a las instancias, y el *prepruning* hace que esas conjunciones de testeos no aparezcan en el árbol.

Un tipo de pruning es el que está basado en el error, para simplificar el árbol de decisión se descartan uno o más subárboles y se reemplazan éstos por hojas, la clase que se le asocia a la hoja es la de mayor frecuencia. Se analizan los nodos más bajos del árbol que no sean hojas, si reemplazar ese subarbol por una hoja o por su rama más usada baja la predicción de error, entonces se poda el árbol, la predicción de error del árbol entero se verá afectada, ya que el índice de error del árbol decrece si el índice de error de alguno de sus subárboles es reducido.

Es claro, que el índice de error sobre el conjunto de entrenamiento sobre el cual el árbol fue construído, no sirve para predecir el índice de error ya que cualquier podado incrementaría dicho índice.

Una técnica es utilizar un nuevo conjunto de casos distinto del conjunto de entrenamiento, la estimación de error que se obtiene de ellos no está influenciada. Aunque esta técnica suele tener una desventaja si los datos son escasos, ya que podría hacer que el conjunto de entrenamiento no sea lo suficientemente grande para construir el árbol de decisión.

2.3.6 Construcción de los conjuntos de datos para estimar el error

Existen varios paradigmas para estimar el error de los algoritmos de clasificación.

1.- **Holdout:** Separa de los datos un porcentaje fijo de instancias p para entrenamiento y $(1-p)$ para testeo, el valor de p debería ser mayor a $\frac{1}{2}$. Un valor típico a usar es $p = 2/3$.

2.- **Random:** Genera varios inductores, tomando I subconjuntos aleatorios, el error es el promedio de los errores derivado por cada predictor. Esto puede producir una mejor estimación del error que el anterior punto.

3.- **r-fold cross validation:** Se dividen aleatoriamente las instancias en r mutuamente excluyentes particiones, de aproximadamente igual tamaño. Las instancias de los $(r-1)$ bloques son usadas como conjunto de entrenamiento y el predictor extraído se testea sobre el bloque remanente, este proceso es repetido r veces, para cada corrida se toma un conjunto de testeo diferente. El error que produce este proceso es el promedio de error de los r bloques.

4.- **r-fold stratified cross validation:** Se dividen aleatoriamente las instancias como en el *r-fold cross-validation* pero se mantiene la distribución de las clases según el conjunto de datos original.

2.3.7 Test de evaluación de algoritmos

En este punto, mostramos una de las formas de evaluar los algoritmos. Dado un algoritmo, y los errores producidos para cada una da las corridas de ese algoritmo calculamos la media, varianza y desvío estándar como:

Sea $pe(h_i)$, $i = 1, 2, ..., r$ el error producido por la i-ésima corrida. Y, sea A el algoritmo

$$media(A) = \frac{1}{r}\sum_{i=1}^{r} pe(h_i)$$

$$\mathrm{var}(A) = \frac{1}{r}\left[\frac{1}{r-1}\sum_{i=1}^{r}(pe(h_i) - media(A))^2\right]$$

$$ds(A) = \sqrt{\mathrm{var}(A)}$$

En general, el error se representa como su *media* seguido por el símbolo "±" seguido por su desvío estándar.

Cuando comparamos diferentes inductores, sobre el mismo dominio, el desvío estándar puede ser visto como una imagen de la robustez de los algoritmos: si el error sobre diferentes conjuntos de testeos con el clasificador construído sobre diferentes conjuntos de entrenamientos es muy diferente sobre un test a otro el algoritmo no es robusto al cambiar el conjunto de entrenamiento tomado del mismo conjunto de datos.

Para determinar cuándo la diferencia entre dos algoritmos es significante o no, se realiza el siguiente test. Supongamos que A_p es el algoritmo propuesto y A_s es el algoritmo estándar, se calculan los siguientes valores

$$media(A_s - A_p) = media(A_s) - media(A_p)$$

$$ds(A_s - A_p) = \sqrt{\frac{ds(A_s)^2 - ds(A_p)^2}{2}}$$

$$ad(A_s - A_p) = \frac{media(A_s - A_p)}{ds(A_s - A_p)}$$

Si $ad(A_s\text{-}A_p) > 0$ entonces A_p tiene mejor performance que A_s. Si $ad(A_s\text{-}A_p) \geq 2$ entonces A_p tiene mejor performance que A_s con un nivel de confianza de 95%. Por otro lado si $ad(A_s\text{-}A_p) \leq 0$ entonces A_s tiene mejor performance que A_p y si $ad(A_s\text{-}A_p) \leq -2$ entonces A_s tiene mejor performance que A_p con un nivel de confianza de 95%.

Capítulo III

Software *See5*

3.1 Software

El software que utilizamos para procesar los datos con métodos de aprendizaje de máquina supervisado fue el *See5*[1].

Tanto el *See5* bajo *Windows* como su contraparte bajo *Unix C5.0,* son versiones superiores del *C4.5*. Analizamos el funcionamiento del *C4.5* que no difiere demasiado del *See5,* aunque este último es más rápido y utiliza menos memoria.

3.2 C4.5 *[QUI/93]*

C4.5 es un programa de clasificación, que desciende del *ID3* el cual analiza numerosos registros ya clasificados y construye un modelo inductivamente, mediante la generalización de los casos específicos.

3.2.1 Características de los datos

Descripción de los atributos: Los datos a ser analizados deben estar en "archivos planos", toda la información acerca de un objeto se debe poder expresar mediante una colección fija de propiedades o atributos. Los valores de cada atributo pueden ser numéricos o discreto pero los atributos usados para describirlo no deben variar de un caso a otro, por ejemplo, es difícil imaginar un atributo que describe la historia médica completa, porque cada historia varía en el tipo y cantidad de información contenida en ella.

Clases predefinidas: Las categorías para las cuales los casos son asignados, deben estar establecidas de antemano, esto es aprendizaje *supervisado*, en contraste con el aprendizaje *no supervisado* en el cual los grupos se encuentran mediante el análisis.

Clases discretas: Las clases deben estar claramente delimitadas, un caso debe o no pertenecer a una clase, y debe haber muchos más casos que clases.

Datos suficientes: La generalización inductiva procede por la identificación de patrones en los datos. Para poder distinguir los casos coincidentes se utilizan *test estadísticos*, deben existir casos suficientes para permitir a esos *test* ser efectivos. La cantidad de datos requeridos depende de la cantidad de atributos, clases y complejidad del modelo de clasificación, a medida que aumenta, más datos serán necesarios para construir un modelo de clasificación

[1] Este software fue obtenido de la página de Quinlan http://www.rulequest.com/see5-win.html

confiable. Un modelo simple se puede identificar a veces usando pocos casos, pero un modelo de clasificación detallada en general requiere de cientos o miles de *casos de entrenamiento.*

Modelo de clasificación lógica: El programa construye solamente, clasificadores que se pueden expresar como *árboles de decisión* o *reglas de producción.* Estas formas restringen la descripción de una clase a una expresión lógica cuyas primitivas son sentencias acerca de los valores de atributos particulares.

3.2.2 Arbol de decisión (C4.5)

El árbol de decisión que construye el *C4.5* contiene:

- Hojas que indican clases.
- Nodos que especifican algún test a ser llevado a cabo sobre un valor singular de atributo, con una rama y un subárbol para cada posible resultado del test.

Un árbol de decisión se puede usar para clasificar un caso comenzando desde su raíz y moviéndose a través de él hasta llegar a una hoja.

3.2.2.1 Construcción del árbol de decisión

La idea original se remonta a fines de 1950 en el trabajo realizado por Hoveland and Hunt, *Experimentos en Inducción* [HUN/66] que describen experimentos con varias implementaciones de *sistemas de aprendizaje de conceptos.* Otras investigaciones, han llegado independientemente a un método similar. [FRI/77] expone los fundamentos del famoso sistema *CART* [BRE/84]. Esa idea es también tomada por el *ID3* [QUI/79], [QUI/83], [QUI/86].

3.2.2.1.1 Dividir y conquistar

La idea del método de *Hunt* para construir un árbol de decisión de un *conjunto de entrenamiento* T es simple:

Sean $\{C_1, C_2, \ldots, C_k\}$ clases, hay tres posibilidades.

1. T contiene una o más instancias, todas pertenecientes a una misma clase C_j, en este caso, el árbol de decisión para T es una hoja identificando la clase C_j.
2. T no contiene instancias. Otra vez, en esta situación el árbol de decisión es una hoja pero la clase asociada con la hoja debe estar determinada. Por ejemplo, podría ser determinada de acuerdo al conocimiento del dominio, tales como la clase mayoritaria.
3. T contiene instancias que pertenecen a varias clases. En este caso la idea es redefinir T en T_i subconjuntos de entrenamientos disjuntos.

Es muy importante la elección adecuada del *test* que divide a T en nuevos subconjuntos, lo ideal es construir particiones con la menor cantidad de clases posibles, de manera que el árbol final sea pequeño.

Por qué no explorar todos los posibles árboles de decisión y seleccionar el más simple?

Desafortunadamente el problema de encontrar el menor árbol de decisión de un conjunto de entrenamiento es *NP-completo* [HAY/76]. Por lo tanto una vez que un *test* ha sido seleccionado para particionar el conjunto de entrenamiento, la elección es hecha en concreta y las consecuencias de elecciones alternativas no son exploradas.

Ejemplo 1: ***Construcción del árbol de decisión a partir de un conjunto de datos***

Se divide sucesivamente el *conjunto de entrenamiento* hasta que todos los casos de cada subconjunto pertenezcan a una misma clase. La Tabla 5 muestra un ejemplo con un *conjunto de entrenamiento*[2] pequeño.

Estudio	Edad	Cant. Hogar	Sexo	Clase
Primario	25	3	Mujer	Ocupado
Primario	30	4	Mujer	Desocupado
Primario	35	4	Hombre	Desocupado
Primario	22	4	Hombre	Desocupado
Primario	19	3	Hombre	Ocupado
Secundario	71	4	Mujer	Desocupado
Secundario	65	3	Mujer	Desocupado
Secundario	75	4	Hombre	Ocupado
Secundario	68	4	Hombre	Ocupado
Secundario	70	4	Hombre	Ocupado
Universitario	22	4	Mujer	Ocupado
Universitario	33	3	Hombre	Ocupado
Universitario	14	3	Mujer	Ocupado
Universitario	31	3	Hombre	Ocupado

Tabla 5: Ejemplo de un conjunto de entrenamiento.

Dado que no todos los casos pertenecen a la misma clase, el algoritmo de *dividir y conquistar* los separa en subconjuntos.

Al testar el *Estudio*, el último grupo contiene solamente casos de *Ocupado*, pero el primer y segundo grupo contienen aún casos mezclados. El primer grupo se subdivide además por *Cant_Hogar* $\leq$ *3* y *Cant_Hogar* > *3*. El segundo grupo se divide por *Sexo* = *Mujer* y *Sexo* = *Hombre*.

El árbol de decisión que da el *See5* es:

```
ESTUDIO = Universitario: Ocupado (4)
ESTUDIO = Primario:
:...CANT_HOGAR <= 3: Ocupado (2)
:   CANT_HOGAR > 3: Desocupado (3)
ESTUDIO = Secundario:
:...SEXO = Hombre: Ocupado (3)
    SEXO = Mujer: Desocupado (2)
```

3.2.2.1.2 Criterio de ganancia (gain)

El criterio de *gain* es uno de los criterios utilizados para seleccionar un test de particionamiento.

Notaciones:

1. Si S es un conjunto de casos, freq(C_i,S) es el número de casos en S que pertenecen a la clase C_i.
2. |S| el número de casos en S.

La teoría de la información en que se basa este criterio es que la información que lleva un mensaje depende de su probabilidad, y puede ser medida en *bits* como – $\log_2$ de esa probabilidad.

[2] El ejemplo es una muestra de EPH (Encuesta Permanente de Hogares)

Entonces, por ejemplo, si hay ocho mensajes equiprobables, la información llevada por alguno de ellos es $-\log_2(1/8) = 3$ bits.

Si se selecciona un caso aleatorio de un conjunto S de casos y se anuncia que éste pertenece a alguna clase Cj. Este mensaje tiene probabilidad

$$\frac{freq(C_j, S)}{|S|}$$

y entonces la información que lleva es

$$-\log_2\left(\frac{freq(C_j, S)}{|S|}\right) \ bits$$

para encontrar la información de un mensaje sumamos todas las clases en proporción a sus frecuencias en S

$$info(S) = -\sum_{j=1}^{k}\left(\frac{freq(C_j, S)}{|S|}\right) \times \log_2\left(\frac{freq(C_j, S)}{|S|}\right) \ bits$$

Cuando se aplica al conjunto de casos de entrenamientos, *info (T)* mide la cantidad promedio de información necesitada para identificar la clase de un caso en T (esta cantidad también es conocida como entropía de un conjunto T).

Después de T ha sido particionada de acuerdo con los n resultados de un test X, definimos $info_x$.

$$info_X(T) = \sum_{j=1}^{n} \frac{|T_i|}{|T|} \times info(T_i)$$

la cantidad $gain(X) = \inf o(T) - \inf o_X(T)$ mide la información que se gana particionando T de acuerdo al test X., el criterio de *gain* selecciona un test que maximice esta información de *gain.*

Ejemplo 2: ***Calcular el criterio de gain***

Calculamos la información de *gain* para los datos dados en el *Ejemplo 1*

```
Info(T) = -9/14*log2(9/14)-5/14*log2(5/14) = 0.940
```

Recordamos que esto representa el promedio de información que se necesita para identificar la clase de un caso en T.

Si usamos el atributo *Estudio* para dividir T en tres subárboles, el resultado es el siguiente:

```
Infox(T) =   4/14*(-4/4*log2(4/4) - 0/4*log2(0/4))
           + 5/14*(-2/5*log2(2/5) - 3/5*log2(3/5))
           + 5/14*(-3/5*log2(3/5) - 2/5*log2(2/5))
           = 0,694

gain(X) = 0.940-0.694 = 0.246
```

Supongamos que en lugar de dividir *T* por el atributo *Estudio* lo hubiésemos dividido por *Sexo*. Esto hubiera separado a *T* en dos subconjuntos.

```
Info_x(T) =   6/14*(-3/6*log2(3/6)-3/6*log2(3/6))
            + 8/14*(-6/8*log2(6/8)-2/8*log2(2/8))
          = 0.892

gain(X) = 0.940-0.892 = 0.048
```

con lo cual el criterio de *gain* toma el atributo *Estudio* y no *Sexo* para separar *T*.

3.2.2.1.3 Criterio de índice de ganancia - gain ratio

A pesar de que el criterio de *gain* da buenos resultados este tiene una gran deficiencia. Por ejemplo, si se tiene un campo *Identificación* que contiene un valor diferente para cada uno de los casos, al particionar en subconjuntos teniendo en cuenta el criterio de *gain*, cada subconjunto contendría un único caso, y $info_x(T) = 0$, con lo cual *gain(X)* es maximal para todos los subconjuntos. Desde el punto de vista de la predicción, esta división es poco útil.

Esto se puede rectificar por un tipo de normalización.

$$splitinfo(S) = -\sum_{j=1}^{n} \left(\frac{|T_i|}{|T|} \right) \times log_2 \left(\frac{|T_i|}{|T|} \right) \ bits$$

Esto representa la información potencial que se genera dividiendo *T* en *n* subconjuntos, entonces,

```
gain ratio(X) = gain(X)/splitinfo(X)
```

Para el caso anterior, si hay k clases, el numerador *gain(X)* es como mucho $log_2(k)$, y el denominador, por el otro lado es $log_2(n)$ donde *n* es el número de casos de entrenamiento ya que todos los casos tienen un único resultado. Si el número de entrenamiento es más grande que el número de clases. Entonces *splitinfo(X)* tendría un valor pequeño.

Ejemplo 3: ***Calcular el criterio de gain ratio***

Para el caso del test sobre *Educación* es

```
Splitinfo(X)     = -5/14*log2(5/14)-4/14*log2(4/14)-5/14*log2(5/14)
                 = 1.577 bits

gain(X)          = 0.246
gain ratio(X)    = 0.246/1.577 =  0.156
```

3.2.2.1.4 Posibles test

C4.5 contiene tres tipos de test diferentes:

- El test "standard" sobre atributos discretos, con un resultado y rama para cada valor posible de ese atributo.

- Un test más complejo, basado sobre un atributo discreto, para el cual los posibles atributos se asignan a grupos de números variables.
- Si el atributo A es continuo, se usa un test binario con resultados $A \leq Z$ y $A>Z$, basado en comparaciones sobre el valor de A contra umbrales de valor Z.

3.2.2.1.5 Test sobre atributos continuos

Si el test debe ser aplicado sobre un atributo A continuo los casos de entrenamiento de T primero son ordenados sobre el valor del atributo A. Hay solamente un número finito de esos valores, si $\{v_1, v_2, v_3, \ldots, v_m\}$ son los valores. Hay *m-1* posibles separaciones de A en dos subconjuntos $\{v_1, \ldots, v_i\}$ y $\{v_{i+1}, \ldots, v_m\}$ con un umbral entre v_i y v_{i+1}, es común elegir el punto medio de cada intervalo como el umbral representativo.

$$\frac{v_i + v_{i+1}}{2}$$

C4.5 elige el valor más alto esto asegura que todos los valores de los umbrales que aparecen en el árbol están realmente en los datos.

Parece costoso examinar los *m-1* casos de umbrales, pero cuando los casos han sido ordenados, esto lleva una sola pasada.

3.2.2.1.6 Valores de atributos desconocidos

A veces los valores de los datos tienen errores. Esto podría ser porque el valor no es relevante en algún caso particular, o podría haber ocurrido un error de tipeo.

Es evidente que un test no puede dar información a cerca de los miembros de una clase cuyos valores del atributo testeado son desconocidos.

Si T es el conjunto de entrenamiento y X un test basado en algún atributo A, y suponemos que el valor de A es conocido en una fracción F de los casos en T. *Info(T)* y $info_x(T)$ son calculados como antes excepto que solo los casos con valores conocidos de A son tomados en cuenta. La definición de *gain* puede ser modificada a

```
gain(x)     = probabilidad A conocida*(info(T)-infox(T))
            + probabilidad A desconocida*0
            = F x (info(T) - infox(T))
```

de manera similar es alterado *splitinfo(X)*.

3.2.2.1.7 Particionamiento del conjunto de entrenamiento

Si el test X con resultados O_1, O_2, O_n que es finalmente elegido tiene resultados desconocidos sobre algunos de los casos, el concepto de particionamiento debe ser generalizado.

Cuando un caso de T con resultado conocido O_i es asignado al subconjunto T_i, esto indica que la probabilidad que el caso pertenezca al subconjunto T_i es 1 y a otros subconjuntos es 0. Cuando el resultado es desconocido, solamente una sentencia probabilística débil puede ser hecha. Asociamos con cada caso para cada subconjunto T_i un peso representando la probabilidad de que el caso pertenezca a cada subconjunto.

Si el caso tiene un resultado conocido, este peso es 1; si el caso tiene un resultado desconocido el peso es sólo la probabilidad de que el resultado sea O_i.

Ejemplo 4: ***Calcular criterio de gain y gain ratio con valores desconocidos***

Supongamos que modificamos los casos anteriores para *Estudio* = *Secundario* colocando un valor desconocido (denotado por "?").

Estudio	Edad	Cant. Hogar	Sexo	Clase
Secundario	71	4	Mujer	Desocupado
Secundario	65	3	Mujer	Desocupado
Secundario	75	4	Hombre	Ocupado
Secundario	68	4	Hombre	Ocupado
?	70	4	Hombre	Ocupado

Tabla 6: Ejemplo de casos de entrenamiento con datos desconocidos

Tenemos las siguientes frecuencias si nos restringimos a los 13 casos restantes:

Estudio	Ocupado	Desocupado	Total
Primario	2	3	5
Secundario	3	0	3
Universitario	3	2	5
Total	8	5	13

Tabla 7: Frecuencias sin el caso desconocido

$$info(T) = -8/13*\log_2(8/13)-5/13*\log_2(5/13)$$
$$= 0.961 \text{ bits}$$

$$info_x(T) = 5/13*(-2/5*\log_2(2/5)-3/5*\log_2(3/5))$$
$$+ 3/13*(-3/3*\log_2(3/3)-0/3*\log_2(0/3))$$
$$+ 5/13*(-3/5*\log_2(3/5)-2/5*\log_2(2/5))$$
$$= 0.7474 \text{ bits}$$

$$gain(X) = 13/14*(0.961-0.747)$$
$$= 0.199 \text{ bits}$$

El *splitinfo(x)* es determinada por el conjunto de entrenamiento completo:

$-5/14*\log_2(5/14) = 0.5305$ para Primario.
$-3/14*\log_2(3/14) = 0.476$ para Secundario.
$-5/14*\log_2(5/14) = 0.5305$ para Universitario.
$-1/14*\log_2(1/14) = 0.27$ para "?"

$$splitinfo(X) = 0.5305+0.476+0.5305+0.27 = 1.807 \text{ bits}$$

$$gain\ ratio = 0.199/1.807 = 0.110$$

Cuando los 14 casos son particionados por este test, los 13 casos para el cual los valores de *Estudio* son conocidos, no presentan problemas. El caso remanente es asignado a todos los bloques de la partición (*Primario, Secundario, Universitario*) con peso *5/13, 3/13, 5/13* respectivamente.

Enfocamos sobre el caso *Primario*:

Estudio	Edad	Cant. Hogar	Sexo	Clase	Peso
Primario	25	3	Mujer	Ocupado	1
Primario	30	4	Mujer	Desocupado	1
Primario	35	4	Hombre	Desocupado	1
Primario	22	4	Hombre	Desocupado	1
Primario	19	3	Hombre	Ocupado	1
?	*70*	*4*	*Hombre*	*Ocupado*	*5/13*

Tabla 8: Pesos correspondientes

Estos son divididos en

Cant_Hogar ≤ 33 Ocupados, *0 Desocupados*
Cant_Hogar ≥ 35/13 Ocupados, *2 Desocupados*

El primer subconjunto contiene casos de una misma clase. El segundo aún contiene casos de dos clases pero el programa no puede encontrar un test que mejore esta situación.

El árbol de decisión tiene la siguiente estructura:

Nivel de estudio = primario:
Edad < 45: *Ocupados (2.0)*
Edad ≥ 45: *Desocupados (2.4/0.4)*
Nivel de estudio = universitario: ocupado (5.4)
.....

Las hojas poseen un valor (N) o (N/E), N es el número fraccional de casos que alcanzan la hoja, E es el número de casos de esa hoja que pertenecen a otra clase.

Veamos que pasa cuando el árbol es usado para clasificar un caso con:
- *Educación = primaria,*
- *Edad = desconocida,*
- *Cant. Hogar = 4,*
- *Sexo = Mujer*

Para clasificar este caso se elige la rama de *Primario*, pero luego no se conoce la *Edad*
- Si la *Edad* fuera **menor** que *45*, podría clasificarse como *Ocupado.*
- Si la *Edad* fuera **mayor** o igual que *45*, podría ser clasificado como *Desocupado* con probabilidad de 2/2.4 (83%) y *Ocupado* con probabilidad de 0.4/3.4(11%).

3.2.2.2 Agrupar atributos

Cuando se tienen atributos discretos, muchas veces es conveniente agruparlos para que el resultado del algoritmo sea más eficiente.

Cómo dijimos anteriormente, al seleccionar un atributo discreto como criterio de separación, se genera una rama diferente por cada posible valor del atributo, esto puede llegar a traer la apertura de gran cantidad de ramas, como consecuencia la cantidad de datos contenidos en algunas ramas pueden llegar a ser insuficientes para la detección de patrones.

Por otro lado, el criterio de *gain ratio*, mide la ganancia de información que produce la separación de ese atributo, y ese valor disminuye a medida que el número de subconjuntos aumenta.

Un ejemplo sería si tenemos un atributo que indica un elemento químico, agruparlos según la familia (halógenos y no metales), según la temperatura de cada elemento (sólido, líquido, o gaseoso) serían algunas de las posibilidades, algunas de las agrupaciones son irrelevantes para el clasificador, la elección más conveniente debe ser hecha de antemano.

3.2.2.3 Podado del árbol

Una característica del *C4.5*, que no posee todos los algoritmos de árboles de decisión es el *podado del árbol con complejidad de costo minimal.* Esto lo hace de la siguiente forma si T es un árbol de decisión usado para clasificar n ejemplos en el conjunto de entrenamiento C. Sea E el conjunto de los mal clasificados de tamaño n. Si $l(T)$ es el número de hojas en T la complejidad del costo de T es:

$$R_\alpha = R(T) + \alpha * l(T)$$

Donde $R(T) = m/n$ es el error estimado de T, si α es el costo de cada hoja, R_α es una combinación lineal de su error estimado y su penalidad por esa complejidad, si α es pequeño la penalidad por tener un gran número de hojas es pequeña y T será grande. Si se convierte algún subárbol S en una hoja. El nuevo árbol T_α clasificará k más ejemplos mal pero contendrá $l(S) - 1$ hojas menos. El costo de complejidad de T_α es el mismo que T si

$$\alpha = \frac{k}{n(l(s)-1)}$$

Hay un único subárbol T_α el cuál minimiza $R_\alpha(T)$ para algún valor de α todos los otros subárboles tienen un costo de complejidad igual o más alto.

Para $T_0=0$, podemos encontrar el subárbol tal que α es el de arriba. Si T_1 es el árbol. Hay entonces una secuencia $T_1 \supset T_2 \supset T_3 \ldots$ para generar T_{i+1} de T_i, examinamos cada nodo *no hoja* del subárbol de T_i y encontramos el mínimo valor de α. El mejor árbol es seleccionado de esta serie de árboles con el error de clasificación que no exceda un error esperado sobre algún conjunto de testeo.

3.2.2.3 Diferencias con otros algoritmos de árbol de decisión

La principal diferencia entre los distintos algoritmos de decisión radica en el criterio que se utiliza para seleccionar el atributo y la forma en que se abrirá el árbol en las diferentes ramas.

A continuación mencionamos las características de otros árboles de decisión:

CART, (Classification And Regression Tree) es un algoritmo de árbol de decisión binario, que cuál tiene exactamente dos hojas para cada nodo interno. Si se considera un problema con dos clases (*A y B*), y un nodo que tiene 100 ejemplos, 50 de cada clase, el nodo tiene una impureza[3] máxima. Si se puede encontrar una separación de los datos en dos subconjuntos una que contenga 40 de la clase *A* y 5 de la clase *B*, el otro 10 de la clase *A* y 45 de la clase *B*. Entonces intuitivamente se redujo la impureza. La impureza sería totalmente removida si se pudiera encontrar una separación que produzca dos subgrupos uno con 50 de la clase *A* y 0 de la clase *B*, el otro con 0 de la clase *A* y 50 de la clase *B* el índice de *Gini* formaliza esta idea.

$$Gini(c) = 1 - \sum_j p_j^2$$

[3] Un nodo es puro cuando todos sus casos corresponden a la misma clase.

donde p_j es la probabilidad de la clase *j* en *c*. Para cada posible separación se suma la impureza de los subgrupos y se elige la separación con la máxima reducción en la impureza.

Para atributos continuos y ordenados CART considera todas las posibles separaciones. Para *n* valores del atributo evaluará *n-1* separaciones. Para atribtos categóricos CART examina todas las posibles separaciones binarias. Para *n* valores de atributos habría 2^{n-1} separaciones.

Cal5 utiliza para seleccionar el atributo de separación una medida de discriminación, *"quotient"*

$$quotient(N) = \frac{A^2}{A^2 + D^2}$$

donde A es la desviación estándar de casos en N y D es el valor medio del cuadrado de las distancias entre las clases. El atributo con menor valor de quotient es elegido como el mejor para la separación de ese nodo.

Un ejemplo sería si tenemos un atributo que indica un elemento químico, agruparlos según la familia (halógenos y no metales), según la temperatura de cada elemento (sólido, líquido, o gaseoso) serían algunas de las posibilidades, algunas de las agrupaciones son irrelevantes para el clasificador, la elección más conveniente debe ser hecha de antemano.

3.2.2.3 Podado del árbol

Una característica del *C4.5*, que no posee todos los algoritmos de árboles de decisión es el *podado del árbol con complejidad de costo minimal.* Esto lo hace de la siguiente forma si T es un árbol de decisión usado para clasificar n ejemplos en el conjunto de entrenamiento C. Sea E el conjunto de los mal clasificados de tamaño n. Si $l(T)$ es el número de hojas en T la complejidad del costo de T es:

$$R_\alpha = R(T) + \alpha * l(T)$$

Donde $R(T) = m/n$ es el error estimado de T, si α es el costo de cada hoja, R_α es una combinación lineal de su error estimado y su penalidad por esa complejidad, si α es pequeño la penalidad por tener un gran número de hojas es pequeña y T será grande. Si se convierte algún subárbol S en una hoja. El nuevo árbol T_α clasificará k más ejemplos mal pero contendrá $l(S) - 1$ hojas menos. El costo de complejidad de T_α es el mismo que T si

$$\alpha = \frac{k}{n(l(s) - 1)}$$

Hay un único subárbol T_α el cuál minimiza $R_\alpha(T)$ para algún valor de α todos los otros subárboles tienen un costo de complejidad igual o más alto.

Para $T_0=0$, podemos encontrar el subárbol tal que α es el de arriba. Si T_1 es el árbol. Hay entonces una secuencia $T_1 \supset T_2 \supset T_3 \ldots$ para generar T_{i+1} de T_i, examinamos cada nodo *no hoja* del subárbol de T_i y encontramos el mínimo valor de α. El mejor árbol es seleccionado de esta serie de árboles con el error de clasificación que no exceda un error esperado sobre algún conjunto de testeo.

3.2.2.3 Diferencias con otros algoritmos de árbol de decisión

La principal diferencia entre los distintos algoritmos de decisión radica en el criterio que se utiliza para seleccionar el atributo y la forma en que se abrirá el árbol en las diferentes ramas.

A continuación mencionamos las características de otros árboles de decisión:

CART, (Classification And Regression Tree) es un algoritmo de árbol de decisión binario, que cuál tiene exactamente dos hojas para cada nodo interno. Si se considera un problema con dos clases (*A y B*), y un nodo que tiene 100 ejemplos, 50 de cada clase, el nodo tiene una impureza[3] máxima. Si se puede encontrar una separación de los datos en dos subconjuntos una que contenga 40 de la clase *A* y 5 de la clase *B*, el otro 10 de la clase *A* y 45 de la clase *B*. Entonces intuitivamente se redujo la impureza. La impureza sería totalmente removida si se pudiera encontrar una separación que produzca dos subgrupos uno con 50 de la clase *A* y 0 de la clase *B*, el otro con 0 de la clase *A* y 50 de la clase *B* el índice de *Gini* formaliza esta idea.

$$Gini(c) = 1 - \sum_j p_j^2$$

[3] Un nodo es puro cuando todos sus casos corresponden a la misma clase.

donde p_j es la probabilidad de la clase j en c. Para cada posible separación se suma la impureza de los subgrupos y se elige la separación con la máxima reducción en la impureza.

Para atributos continuos y ordenados CART considera todas las posibles separaciones. Para n valores del atributo evaluará $n-1$ separaciones. Para atribtos categóricos CART examina todas las posibles separaciones binarias. Para n valores de atributos habría 2^{n-1} separaciones.

Cal5 utiliza para seleccionar el atributo de separación una medida de discriminación, *"quotient"*

$$quotient(N) = \frac{A^2}{A^2 + D^2}$$

donde A es la desviación estándar de casos en N y D es el valor medio del cuadrado de las distancias entre las clases. El atributo con menor valor de quotient es elegido como el mejor para la separación de ese nodo.

Capitulo IV

Análisis de datos simbólicos

4.1 Concepto

El *análisis de datos simbólicos* intenta resumir los datos de una gran base en término de su concepto oculto para extraer nuevos conocimientos. Estos conceptos sólo pueden ser descriptos por tipos más complejos de datos llamados *datos simbólicos*, ellos contienen variación interna y son estructurados. Los métodos estadísticos de análisis de datos estándar (exploratorios, representaciones gráficas, agrupamiento, análisis factorial,...etc) se extienden a estos datos simbólicos.

Los *datos simbólicos* conducen a tablas de datos más complejas llamadas *tablas de datos simbólicos* porque cada celda de la tabla no necesariamente contiene un simple valor categórico o cuantitativo, sino varios valores que pueden ser evaluados y relacionados mediante reglas y taxonomías lógicas. Por ejemplo, una celda puede contener un intervalo o una distribución. Se define el *Análisis de Datos Simbólicos (ADS)* como una extensión del análisis de datos estándar a las tablas de datos simbólicos.

4.2 Definiciones

Atributos: Los atributos en la *tabla de datos simbólicos*, generalmente no son valores simples se usan para describir un conjunto de unidades llamados *individuos*. Las filas son descripciones simbólicas de esos individuos. Los tipos de datos pueden ser:

a) *Un simple valor cuantitativo:*
Por ejemplo, si el *"peso"* es una variable y *w* es un individuo, *peso(w) = 3.5*.

b) *Un simple valor categórico:*
Por ejemplo, *ciudad(w) = Buenos Aires*.

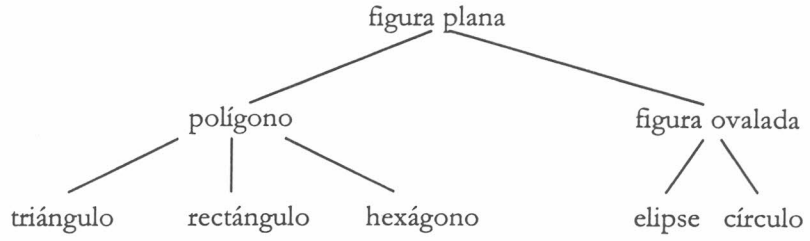
c) *Un conjunto de valores ó categoría (variable multivaluada):*
Por ejemplo, *peso(w) = {3.5, 2.1, 5}* significa que el peso de *w* puede ser de 3.5 ó 2.1 ó 5. Notar que a) y b) son casos especiales de c)

d) *Un intervalo:*
Por ejemplo, *peso(w) = [3, 5]* significa que el peso de *w* varía en el intervalo *[3, 5]*.

e) *Un conjunto de valores con pesos asociados:*
Por ejemplo, en la forma de un histograma. Notar que a), b) y c) son casos especiales de e) donde los pesos son todos 1.

Relaciones entre variables: en el análisis de datos simbólico además podemos definir dependencias entre las variables. Se utilizan comúnmente dos tipos de relaciones:

a) *Taxonomía de variables con estructura jerárquica de categorías:*
Es posible definir atributos que estén ordenados jerárquicamente
Ejemplo: *"la variable figura puede ser representada con la siguiente taxonomía"*



b) *Variables de dependencias jerárquicas:*
Se dice que una variable es jerárquicamente dependiente de otra si para cada valor que tome ésta hay un conjunto de valores que toma la segunda.
Ejemplos: *"si Y es tipos de artículos y Z = artículos "*
si Y = autos, Z = {Audi, Ford, Citroen, ...}
si Y = computadoras, Z = {Compaq, IBM, ...}"

Un caso especial son las variables *no-aplicables.* A veces no tienen sentido el valor de una variable para determinados valores de otra variable.
Ejemplos: *"si Y = sexo, Z = número de hijos (que hayan nacido de esa persona)"*
si Z = fem., Z = {1, 2, 3, ...}
si Z = masc., Z = {no-aplicable}"

c) *Variables de dependencia lógica:*
Se establece esta dependencia entre dos variables, si los valores de una variable dependen de otra lógica o funcionalmente.
Ejemplos: *"si peso(w) ≤ 55 ⇒ altura (w) ≤ 80"*
"si sexo(w) = fem. ∧ años(w) ≤ 10 ⇒ cantidad de hijos = 0".

4.3 Generación de datos simbólicos

Los datos simbólicos se pueden generar de diferentes formas. Considerando que cada categoría de una variable categórica o de una asociación lógica de variables es un *concepto,* un concepto puede ser por ejemplo, una "ciudad" o un tipo de "empleo" o algo más complejo como una categoría socio-profesional (CSP) asociado con una categoría de edad (E) y una región (R). Obteniendo una nueva variable categórica (V) con un número de categorías dado por |CSP|.|E|.|R| (donde |Y| denota el número de categorías de la variable Y). Cada categoría de esta variable se asocia con un concepto.

Los datos simbólicos también se pueden usar después de un proceso de agrupamiento para describir las propiedades de las clases obtenidas de una forma explicada y sobre eso resumir la gran base de datos describiendo los individuos.

Los datos simbólicos pueden también ser "nativos" en el sentido que ellos resultan del conocimiento experto, cuando describen escenarios de accidentes de tráficos, tipos de emigración, tipos de personas retiradas, etc.

Una importante fuente de objetos simbólicos es provista por bases de datos relacionales, si se quiere estudiar las propiedades de un conjunto de unidades cuya descripción necesita juntar varias relaciones como mostramos en el siguiente ejemplo.

Ejemplo 1: ***Obtener objetos simbólicos a partir de un modelo relacional***

Consideremos dos relaciones FAMILIA, PERSONAS en una base de datos relacional definida de la siguiente forma:

FAMILIA, contiene tres *familias* cada una caracterizada por su *cantidad de personas* y *región* (Ver Tabla 9)

Familia	Cantidad	Región
F1	2	1
F2	2	2
F3	1	1

Tabla 9: Relación FAMILIAS

PERSONAS, contiene cinco registros con la *familia*, la *edad* y el sexo (Ver Tabla 10)

Persona	Familia	Edad	Sexo
P1	F1	25	M
P2	F2	30	M
P3	F3	50	F
P4	F1	29	F
P5	F2	20	F

Tabla 10: Relación PERSONAS

De estas dos relaciones podemos obtener la siguiente tabla de datos. La cual describe a cada familia.

Familia	Edad	Sexo	Cantidad	Peso
F1	[25, 29]	M(½), F(½)	2	1
F2	[20, 30]	M(½), F(½)	2	2
F3	50	F	1	1

Tabla 11: Tabla de datos simbólicos implicada por las dos relaciones previas

Por lo tanto, si se quiere estudiar un conjunto de familias descriptas por las cinco variables asociadas con las dos relaciones previas, es natural llevar al problema de diseñar métodos estadísticos para los siguientes tipos de variables simbólicas:

- *Variables con Intervalos:* Por ejemplo la variable *Sexo*, toma varios valores en la misma celda (Tabla 11)

- *Multivaluadas con peso:* Por ejemplo la variable *Sexo* para familia*SF1*: el peso *F(½)* significa que el 50% de sus integrantes tiene sexo femenino.
- *Reglas:* Además para la Tabla 11, podemos especificar reglas sobre los atributos.
- *Taxonomías:* Podemos establecer un nivel de jerarquía entre los valores de un atributo.

4.4 Objetos simbólicos

Antes de definir lo que es un objeto simbólico daremos la definición de relaciones, descripciones y aserciones.

Relaciones

Se considera una relación R definida sobre un conjunto D y se denota $[d\ R\ d'] \in L$ al resultado de la comparación entre d' y d por R, donde $L = \{verdadero, falso\}$ o $L = [0,1]$.

Si $L = \{verdadero, falso\}$, $[d\ R\ d'] = verdadero$ significa que hay una conexión entre d y d'. En el caso donde $L = [0,1]$, el valor $[d'\ R\ d]$ mide el grado de conexión entre d y d' (es una relación difusa).

Por ejemplo, R puede ser una relación de $\{=, \equiv, \subseteq, \geq, \leq, \Rightarrow\}$.

Descripciones

Si $\mathcal{Y}_j$ es el dominio (espacio de observaciones) de Y_j, con j=1,...,p. Cada elemento $z=(z_1,...,z_p) \in \mathcal{Y}_1 x \mathcal{Y}_2 x \mathcal{Y}_3 ... x \mathcal{Y}_p$ es la *descripción.*

$\mathcal{Y}_j$ puede ser un conjunto de valores simples o multivaluados.

Ejemplo: $z=(100,20,7,8)$ *o* $z=([0,100], [10,20], [4,7], [3,8])$

Aserciones

Una *aserción q* es una conjunción de condiciones.

Ejemplo: $q = [tamaño \leq 100] \wedge [tamaño \geq 50] \wedge [color \in \{rojo, verde\}]$

Formalmente,

$$q = \bigwedge_{v=1}^{r} \left[Y_{jv} R_{jv} z_v \right]$$

la función aq: Ω -> $\{0,1\}$ y está definida por:

$$a_q(u) = \bigwedge_{v=1}^{r} \left[Y_{jv}(u) R_{jv} z_v \right]$$

se llama función de *extensión de mapeo*

el conjunto u de Ω que cumplen con el requerimiento *extensión*

$$Ext(q) = \left\{ u \in \Omega \mid \left[Y_{jv}(u) R_{jv} z_v \right] = 1 \forall v = 1,...,r \right\}$$

Objetos simbólicos

La mayoría de los algoritmos de ADS presentan en la salida la descripción d de una clase C de individuos.

Más formalmente, si Ω es un conjunto de individuos, D un conjunto conteniendo las descripciones de los individuos y las descripciones de clases de individuos, Y un mapeo definido de Ω a

D el cual asocia cada $w \in \Omega$ una descripción $d \in D$ usando un vector de variables Y_i de tipo a) o e) de la sección 3.2 la descripción de un individuo w se llama *descripción individual.* La descripción de una clase C de individuos se llama *descripción intencional.* Por ejemplo la descripción de un accidente de tráfico, o de alguna clase de falla son ejemplos de descripciones intencionales. La descripción d dada por el mapeo Y sobre el conjunto de individuos son usualmente guardados en una tabla de datos simbólicos.

Para definir un objeto simbólico s, se necesita un mapeo a llamado extensión de mapeo, una descripción d (en general, intencional) y una forma de comparar esta descripción individual, basada en una relación R, más formalmente:

Un objeto simbólico s es una tripla $s = (a, R, d)$ donde d es una descripción, R es una relación entre descripciones y a es un mapeo definido de Ω en L dependiendo de R y d.

Básicamente se distinguen dos clases de objetos simbólicos: objetos simbólicos booleanos y objetos simbólicos modales.

4.4.1 Objetos simbólicos booleanos:

Dada una tabla de datos de p variables $Y_1,\ldots Y_p$ de los tipos a) a d), definidos en el punto 3.2, se denota $Y(w)=(Y_1(w)\ldots Y_p(w))$ el vector de variables, y se considera una descripción $d = (d_1,\ldots.d_p) \in D$ junto con p relaciones binarias R sobre el dominio de Y_j. Resulta un objeto simbólico booleano si se asume que la sentencia $[Y(w)\ R\ d] \in L$ puede ser verdadera o falsa.

Sintaxis de objetos simbólicos booleanos en el caso de *aserciones*, una *aserción* es un tipo especial de objetos simbólicos definidos por $s = (a, R, d)$ donde R se define por $[d'\ R\ d] = \wedge_{j=1,\ldots p}[d'_j R_j D_j]$ donde $a(w) = [Y(w)\ R\ d]$ en el caso booleano.

Ejemplo 2: ***Objeto simbólico booleano***

```
p = 2 variables, Y1 = edad e Y2 = CSP
a(w) = [edad(w)⊆{12,20,28}] ∧ [CSP(w)⊆{empleado, trabajador}].
```

La sintaxis de los objetos simbólicos es la siguiente:
Si en la tabla de datos simbólicos originales un individuo w esta descripto por:

```
edad(w)={12, 20} y CSP(w)={empleado}
d1={12, 20, 28}  d2={empleado, trabajador}, d=(d1, d2),
Rj= ⊆(para j = 1, 2)

a(w) = [Y(w) R d]
     = [{12,20}⊆{12, 20, 28}]∧[{empleado}⊆{empleado, trabajador}]
     = verdadero ∧ verdadero = verdadero.
```

4.4.1.1 Extensión de un objeto simbólico booloeano

La extensión de un objeto simbólico s, denotada por $Ext(s)$, es el conjunto de todos los individuos w de Ω con a(w) = verdadero. Esto es idéntico a la extensión de a, denotado por $Extent(a)$. Por lo tanto, tenemos:

```
Ext(s) = Extent(a) = {w ∈ Ω / a(w) = verdadero}
```

4.4.2 Objetos simbólicos Modal

Sea $Y(w)=(Y_1,\ldots,Y_n)$ un vector de variables Y_j llamamos objeto simbólico modal si L es un intervalo tal que $a(w) = [Y(w)\ R\ d] \in L = [0,1]$.

Como en el caso booleano, una *aserción* modal $[d'Rd] = \wedge^*_{i=1..p}[d'_i R_i d_i]$ donde $\wedge^*$ está dada por la definición de mapeo f.

Ejemplo 3: *Objeto simbólico Modal*

Eligiendo `f:[0,1]^p -> L=[0,1]` está completamente definido por:

$a(w)=[edad(w)R_1\{12(0.2),[20,28](0.8)\}]\wedge^*[CPC(w)R_2\{empleado(0.4),trabajador(0.6)\}]$.

En este caso, los pesos (0.2), (0.8), (0.4), (0.6) representan frecuencias pero podrían considerarse casos más generales de pesos como "posibilidades", "creencias", "capacidades".

4.4.2.1 Extensión de un objeto simbólico modal.

Dado un b > 0, la extensión de un objeto simbólico modal es definido por:

$Ext_b(s) = Extent_b(a) = \{w \in O / a(w) >= b\}$.

4.5 Ejemplo de un Análisis de datos simbólicos

El siguiente es un ejemplo de la posible aplicación del *Análisis de Datos Simbólico* a una base donde los hogares, se encuentran caracterizados por sus regiones REGION, el número de habitaciones HAB, el número de baños BAÑOS y su categoría socio-profesional CSP (Tabla 12):

Id.	Región	Hab.	Baños	CSP
11404	Norte	2	1	Baja
11405	Norte	2	1	Media
11406	Norte	1	3	Media
...	...	...	...	...
12112	GBA	1	3	Baja
12112	GBA	2	2	Media
12112	GBA	1	3	Alta

Tabla 12: Tabla de datos estándar para familias

En datos de censos donde hay gran cantidad de datos, se pueden resumir estos describiendo cada región por los hogares. Para esto borramos la primera columna de la Tabla 12 y obtenemos la Tabla 13.

Región	Hab	Baños	CSP
Norte	2	1	Baja
Norte	2	1	Media
Norte	1	3	Media
...	...	...	...
GBA	1	3	Baja
GBA	2	2	Media
GBA	1	3	Alta

Tabla 13: La primera columna con la identificación de la familia ha sido borrada

Ahora podemos describir cada ciudad con el histograma de la categoría de cada variable. En la siguiente tabla de datos simbólicos Tabla 14, cada celda contiene un histograma y no un número cuantitativo o cualitativo como en la tabla de datos estándar.

Región	Hab.	Baños	SPC
Norte	2 (2/3), 1(1/3)	1(2/3), 3(1/3)	...
...	...	...	...
GBA	1(2/3), 2(1/3)	2(2/3), 3(1/3)	...

Tabla 14: Una tabla de datos simbólicos donde cada celda contiene un histograma.

Finalmente, aplicaríamos a esta *tabla de datos simbólicos* (la cual puede ser completada por reglas de asociación y taxonomías de variables) los métodos de ADS implementados, histogramas de cada variable simbólica, clasificación, análisis factorial, visualizaciones gráficas de descripciones simbólicas.

Capítulo V

Software SODAS

5.1 Proceso "TREE"

El software utilizado para procesar los datos simbólicos, es el *SODAS (Symbolic Oficial Data Analysis System),* participaron de este proyecto alrededor de 17 grupos de investigación Europeos y tres Institutos Nacionales de Estadísticas(EUSTAT/España, INE/Portugal, ONS/London).

Varios métodos de análisis pueden ser aplicados a los datos simbólicos, por ejemplo, análisis de componentes principales, gráficos de estrellas, agrupamientos, árboles de decisión, etc.

Nosotros nos concentraremos en el proceso "TREE" (árbol de decisión), para realizar luego una comparación con los árboles de decisión que genera el *See5.*

El algoritmo es propuesto para tratar explícitamente datos probabilísticos. Se considera una población de n objetos particionados en m clases.

El objetivo es describir en forma de un árbol binario, las distintas clases.

5.2 Características de los datos

Consideramos una población $\Omega = \{1,\ldots,n\}$ con n objetos. Ω está particionado en m clases disjuntas $C1,\ldots,Cm$. Cada objeto $k \in \Omega$ está descripto por dos categorías de variables.

- **C:** la variable clase
- $\mathbf{Y_1,\ldots,Y_p}$: p variables explicativas o predictores.

La variable C es una variable nominal definida sobre Ω con m categorías $\{1,\ldots,m\}$, $C(k)$ es el índice de la clase del objeto $k \in \Omega$ y C es un mapeo definido sobre Ω con dominio $\{1,\ldots,m\}$.

Denotamos por $\mathcal{Y}j$, el conjunto de todos los posibles valores para Y_j. Se consideran tres tipos de predictores:

- **Variable nominal:** $\mathcal{Y}j = \{v_1,\ldots,v_{nj}\}$es un conjunto de categorías no ordenadas.
- **Variable cuantitativa discreta u ordinal:** $\mathcal{Y}j = \{v_1,\ldots,v_{nj}\}$ es un conjunto finito de diferentes niveles numéricos. Donde $v_1 < \ldots < v_s < v_{s+1} < \ldots < v_{nj}$ en el caso de cuantitativo discreto, o $v_s <^* v_{s+1}$ $(s=1,\ldots,nj-1)$ donde $<^*$ es un orden completo sobre $\mathcal{Y}j$ en el caso de una variable ordinal.
- **Variable cuantitativa continua:** $\mathcal{Y}j = [v_1,v_{nj}] \subseteq R$ es un intervalo de la línea real.

5.2.1 La tabla de datos simbólicos y la descripción de n objetos

Para cada objeto k es asociada una descripción con la forma de la siguiente aserción.

$a_k = [C(k)=c_k] \wedge [Y_{k1} \sim f_{k1}] \wedge \ldots \wedge [Y_{kp} \sim f_{kp}]$

- **Descripción de la variable de clase C:** $[C(k) = c_k]$
  La clase de un objeto k toma un único valor $C(k) = c_k \in \{1,\ldots m\}$.
- **Descripción del predictor Y_j:** $[Y_j(k) \sim f_{kj}]$
  Para cada cupla (k,Y_j) hay asociada una variable aleatoria $Y_j(k)$, representada por una distribución probabilística o una frecuencia de distribución f_{kj} sobre $\mathcal{Y}j$.

Las descripciones probabilísticas f_{kj} son de dos tipos:

- **Y_j continuo:** La descripción asociada por Y_j a un objeto k es un intervalo $V_{kj} \subset \mathcal{Y}j$. Asumimos implícitamente que, Y_{jk} tiene una distribución de probabilidad uniforme.

- **Y_j discreto:** La descripción asociada con Y_j está dada por una distribución de frecuencia sobre un subconjunto $\{v_{j1},\ldots,v_{j,nj-1}\}$sobre $\mathcal{Y}j$.

5.3 Preguntas binarias

La división del árbol se hace a través de preguntas binarias, en cada paso se dividen los casos evaluando las respuestas binarias (*si o no*) sobre las variables.

Distinguimos dos tipos de preguntas binarias dependiendo si Y_j tiene valores con o sin orden.

- **Casos con orden:**
  Las preguntas binarias son de la forma: $[Y_j \leq c]$; si o no?

- **Casos sin orden:**
  Las preguntas binarias son de la forma:$[Y_j \in V]$; si o no?

La Tabla 15 resume las conexiones entre tipo de variable, su descripción, y la pregunta binaria que es construida.

Variable	Nominal	Ordinal	Continua
Descripción	Distribución	Distribución	intervalo
Pregunta	[Yj ∈ V]?	[Yj ≤ c]?	[Yj ≤ c]?

Tabla 15: Variables y Preguntas.

- **Preguntas de la forma $[Y_j \leq c]$**
  En el caso discreto, si Y_j tiene n_j valores, se construyen n_{j-1} preguntas binarias(donde c, el valor del umbral es elegido entre $v_{j1},\ldots,v_{jnj-1}$
  En el caso continuo, es diferente del anterior, porque un intervalo $[m_{kj}, M_{kj}]$ ofrece más posibilidades para elegir el umbral (2n-1 posibilidades). También se puede elegir el punto medio de los intervalos.

- **Preguntas de la forma $[Y_j \in V]$**
  En este caso se pueden tener $(2^{nj-1}-1)$ preguntas diferentes con n_j elementos diferentes.
  La búsqueda binaria es equivalente a elegir un subconjunto V de Y_j.

5.3.1 Construcción de una nueva tabla de variables binarias

El objetivo de este procedimiento es construir una tabla de la siguiente forma:

	1 ... q ... Q
1	
⋮	⋮
k	... p_{kq} ...
⋮	⋮
n	

donde, $\{1,\ldots,Q\}$ son todas las preguntas binarias posibles.

p_{kq} es la probabilidad (peso) que la descripción Y_{kj} del objeto k se asigne a la propiedad $[Yj \leq c]$ o $[Yj \in V]$, asociado con la pregunta q.

Por convención, se asume que el nodo izquierdo tiene la propiedad $[Y_j \leq c]$ o $[Y_j \in V]$ mientras que los del nodo derecho tienen la propiedad $[Y_j > c]$ o $[Y_j \notin V]$.

p_{kq} = probabilidad que k sea asignado al nodo izquierdo, usando $Y_j(k)$
$1\text{-}p_{kq}$ = probabilidad que k sea asignado al nodo derecho, usando $Y_j(k)$

Cálculo de las probabilidades:

- **si Y_j es continua**
  descripción de k : $[m_{kj}, M_{kj}]$
  pregunta binaria q : $[Y_j \leq c]$

$$p_{kq}\begin{cases}\dfrac{c-m_{kj}}{M_{kj}-m_{kj}} \text{.....si} \quad c \in [m_{kj}, M_{kj}] \\ 0 \text{.....} si \quad c \langle m_{kj} \\ 1 \text{.....} si \quad c \rangle m_{kj}\end{cases}$$

- **si Y_j es ordinal**
  descripción de k : f_{kj}
  pregunta binaria q : $[Yj \leq c]$

$$p_{kq} = \sum_{v \leq c} f_{kj}(v)$$

- **si Y_j es nominal**
  descripción de k : f_{kj}
  pregunta binaria q : $[Yj \in V]$

$$p_{kq} = \sum_{v \in V} f_{kj}(v)$$

5.4 Construcción del árbol de decisión

En cada paso del algoritmo, la estrategia consiste en separar el nodo terminal que tiene el mejor incremento de "información".

Se construye una secuencia de árboles binarios:

$A_1, A_2, \ldots, A_p \ldots, A_{max}$

En la Figura 3, los nodos negros porveen el mejor incremento de "información"

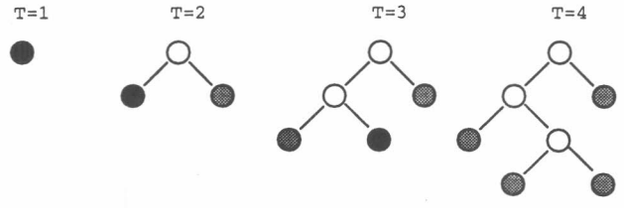


Figura 3: Ejemplo de los primeros cuatro pasos de crecimiento del árbol.

5.4.1 Selección del mejor subárbol

Selecciona un árbol A^* asociado con la "mejor" regla de decisión, la calidad de una regla de decisión se define mediante un *índice de predicción correcta estándar.* El índice asociado con el árbol A_t se denota por $R(A_t)$.

$$A^* = \arg\min_{A \in \{A_1, \ldots, A_{ar1}\}} R(A)$$

Notaciones:

T_A: conjunto de nodos terminales admisibles para el particionamiento de un paso del algoritmo(nodos que no satisfacen la regla de parada del algoritmo)

T_D: conjunto de nodos terminales definitivos durante la fase de crecimiento del árbol (los que satisfacen la regla de parada).

T_C: conjunto de todos los nodos $T_C = T_A \cup T_D$. También será llamado conjunto de nodos terminales recurrentes.

$Q_A(t)$: conjunto de preguntas admisibles a un nodo t.

$Q_C(t)$: conjunto de preguntas candidatas a un nodo t. Son preguntas binarias cuyos nodos no son "tan pequeños" y proveen un "mínimo de información".

$P_t(i)$: probabilidad condicional observando la clase i dentro del nodo t.

$p_k(t)$: probabilidad del objeto k de pertenecer al nodo t.

p_{kt}: probabilidad que k satisfaga la pregunta q.

W(t,q): información que provee la pregunta q sobre el nodo t. Esto es la calidad de la separación binaria producida por la pregunta.

q_t*: mejor pregunta para el nodo t.

q*: mejor pregunta entre todos los nodos terminales en un paso del particionamiento.

l,nl: nodo izquierdo creado por la separación y su tamaño asociado.
r,nr: nodo derecho creado por la separación y su tamaño asociado.

5.4.2 Algoritmo de árbol de decisión general

A continuación se detalla el algoritmo y en el punto siguiente se da la explicación de los pasos(x).

```
Comenzando con el nodo raíz
Mientras (1) (el tamaño del árbol es admisible) y (existe nodos terminales admisibles:
            T_A ≠ 0)
      Para cada nodo t: ∀t ∈ T_A
            Para cada pregunta admisible q del nodo t: ∀q ∈ Q_A(t)
                  (2) Separar t en dos nuevos nodos terminales temporarios: l y r
                  T_C <- T_C \{t}U{l,r}
                  (3) Computar el tamaño de los nodos de l y r: n_l y n_r
                  Si el tamaño de los dos nodos es admisible: (4) (n_l > umbral mínimo) y
                  (n_r > umbral mínimo)
                        (5) Computar la calidad de la separación: W(t,q)
                        Si la calidad es suficientemente grande: W(t,q) > umbral mínimo
                        Entonces q es candidata para el nodo t: q ∈ Q_C(t)
                        Sino: renunciar a la pregunta q
                  Sino: renunciar a la pregnta q
            Fin
            Si no existe pregunta candidata para el nodo t: Q_C(t) = ∅
            Entonces t es un nodo terminal: T_D <- T_D U {t}
            Sino (6) seleccionar la mejor pregunta para el nodo t: q_i*
      Fin
      Si no existe nodo t para separar Q_C(t) = ∅, ∀t ∈ T_A
      Entonces fin del algoritmo
      Sino (7) elegir la mejor separación del mejor nodo: q*
      Actualizar el nodo del árbol:
      Si el tamaño de los nodos hijos l es suficientemente grande y
            l no es un nodo puro entonces T_A <- T_A U {t}
      Entonces: l es un nodo terminal: TD <- TD U {t}
      Si el tamaño de los nodos hijos r es suficientemente grande y
            r no es un nodo puro entonces T_A <- T_A U {r}
      Entonces: l es un nodo terminal: TD <- TD U {r}
      T_C <- T_A + T_D
      (8) Computar la información descriptiva asociada con el nuevo nodo.
      (9) Estimar la proporción de predicción de error R(A) del árbol actual.
Fin
```

5.4.3 Descripción del algoritmo

Tamaño y admisibilidad de los nodos hijos *l, r* con los pesos n_l y n_r (paso (4))

Primero, computamos las probabilidades de un objeto *k* para los nodos hijos *l* y *r* del nodo *t*.

$$p_k(l) = p_k(t) \times p_{kq}$$

$p_k(r) = p_k(t) \times (1-p_{kq})$

Donde p_{kq} fue calculado en la sección 5.3.1 y $p_k(t)$ es la probabilidad de que un objeto k sea miembro de un nodo t. En el primer paso de el algoritmo, tenemos $p_k(t) = 1$ para $k=1,\ldots,n;$ porque todos los individuos pertenecen al nodo raíz, en los pasos siguientes las probabilidades p_{kq} se computan una vez que la mejor separación se eligió. El n_l y el n_r se calculan como:

$$n_l = \sum_{k=1}^{n} p_k(l)$$

$$n_r = \sum_{k=1}^{n} p_k(r)$$

condición de admisión para los tamaños n_l y n_r

la calidad de la separación binaria q para el nodo t es calculada si y solo si los tamaños de los nodos l y r son más grandes que un umbral dado.

$(n_l \geq umbral\ min)$ **y** $(n_r \geq umbral\ min)$ donde el *umbral min* es elegido a priori

Búsqueda de la mejor separación (paso (6) y (7))

Se actualiza el conjunto Tc de los nodos terminales, los dos nodos hijos l y r sustituyen al nodo t que se encuentra bajo estudio, y se calcula la calidad de la separación en base al nuevo conjunto.

$$T_c = T_c \setminus \{t\} \cup \{l,r\}$$

Calidad $W(t,q)$ de una separación q para un nodo t (paso (5))

El criterio de separación general está definido por

$$W(t,q) = \log \prod_{k=1}^{n} \sum_{s \in T_c} p_k(s).P_s(c_k) \qquad (5.1)$$

donde $p_k(s)$ es la probabilidad de que el objeto k sea asignado al nodo s, y $P_s(c_k)$ es la probabilidad que la clase C(k) = c_k de k es observada, condicionalmente sobre el nodo s:

$$P_s(c_k) = P_s(i) \quad si\ k \in Ci$$

O, equivalentemente

$$P_s(C_k) = P_s(1)^{ck1} \times \ldots \times P_s(K)^{ckm}$$

$c_{ki} = 1 \Leftrightarrow k \in C_i$. Sin embargo en (5.1) solamente estamos interesados en la información asociada con los dos nodos hijos, entonces se puede escribir como:

$$W(t,q) = \log \prod_{k=1}^{n} \sum_{s \in T_c} (p_k(l).P_l(c_k) + p_k(r).P_r(c_k) + \lambda_k)$$

donde λ_k es una cantidad fija que sólo depende de $p_k(s)$ y $P_s(c_k)$ asociados con los nodos s que fueron previamente calculados.

$$\lambda_k = \sum_{\substack{s \in T_i \\ s \in \{l,r\}}} p_k(s) * P(c_k)$$

Pl(1),...,Pl(m) y *Pr(1),...,Pr(m)* son estimadas por el algoritmo de EM

Algoritmo de EM para estimar la probabilidad condicional $P_l(i)$, $P_r(i)$

La idea del algoritmo EM es encontrar el valor del parámetro que maximiza *W(t,q)* en forma iterativa. En nuestro caso los parámetros son las siguientes probabilidades condicionales

$P_l^{(h)}(i)$ y $P_r^{(h)}(i)$ i = 1,..., k h = 0,1,2....(índice de iteración)

cada paso de la iteración consiste en repetir dos pasos *Expectación (E) y Maximización (M)*

Paso E: computamos el valor esperado de la probabilidad de que *k* pertenezca al nodo *l* y *r* en la iteración *h* del algoritmo.

$$E^{(h)}(k,l) = \frac{P_l^{(h)}(c_k) * p_k(l)}{P_l^{(h)}(c_k) * p_k(l) + \lambda_k}, \quad k = 1,...,n$$

$$E^{(h)}(k,r) = \frac{P_r^{(h)}(c_k) * p_k(r)}{P_r^{(h)}(c_k) * p_k(r) + \lambda_k}, \quad k = 1,...,n$$

Paso M: calculamos la nueva probabilidad condicional:

$$P_l^{(h+1)}(i) = \frac{\sum_{k=1}^{n} c_{ki} E^{(h)}(k,l)}{\sum_{k=1}^{n} E^{(h)}(k,l)} = \frac{\sum_{k \in Ci} E^{(h)}(k,l)}{\sum_{k=1}^{n} E^{(h)}(k,l)}, \quad i = 1,...,m$$

$$P_r^{(h+1)}(i) = \frac{\sum_{k=1}^{n} c_{ki} E^{(h)}(k,r)}{\sum_{k=1}^{n} E^{(h)}(k,r)} = \frac{\sum_{k \in Ci} E^{(h)}(k,r)}{\sum_{k=1}^{n} E^{(h)}(k,r)}, \quad i = 1,...,m$$

Podemos entonces computar el valor del criterio $W^{(h+1)}(t,s)$ reemplazando la probabilidad condicional por su estimada $P_l^{(h+1)}(i)$ y $P_r^{(h+1)}(i)$.

El algoritmo para cuando:

$W^{(h+1)}(t,q) - W^{(h)}(t,q) < \varepsilon$

Caso inicial:

$$P_l^{(0)}(i) = \frac{\sum_{k=1}^{n} c_{ki}\, p_k}{\sum_{k=1}^{n} p_k(l)} = \frac{\sum_{k \in Ci} p_k(l)}{nl} \quad , \quad i = 1,\ldots,m$$

$$P_r^{(0)}(i) = \frac{\sum_{k=1}^{n} c_{ki}\, p_k}{\sum_{k=1}^{n} p_k(r)} = \frac{\sum_{k \in Ci} p_k(r)}{nr} \quad , \quad i = 1,\ldots,m$$

Selección de la mejor separación

La selección se hace en dos pasos:

- Selección de la mejor pregunta q para cada nodo terminal t: q_t^* (paso (6))
- Selección de la mejor pregunta q_t^* entre todos los nodos terminales t: q^*(paso (7))

Como consecuencia, la mejor separación es aquella que produce el mejor árbol de *(T+1)* nodos.

Descripción de información calculada para cada nuevo nodo t (paso (8)):

Los siguientes tipos de información son calculados para cada nuevo nodo t:

- **Información relacionada a la descripción del nodo:**
  $[Y_j \in V]$, $[Y_j < c]$, *etc*
- **Informacón sobre los individuos**
  1. Probabilidad de ser parte del objeto k para el nodo t: $p_k(t)$
  2. el conjunto de los individuos asignados a t: $nodo(t) = \{k / pk(t) > pk(s),\ \forall s \neq t\}$
- **Información sobre las clases de las particiones dadas a priori**
  Definimos la siguiente que provee información general sobre las K clases a priori sobre el nodo t:

	Tamaño	Prob.	d
clase 1	$n_t(1)$	$p_t(1)$	
...			
clase i	$n_t(i)$	$p_t(i)$	*
...			
clase m	$n_t(m)$	$p_t(m)$	

Tabla 16: Información de las clases

1. $pt(i)$ = probabilidad de la clase i dentro del nodo t.
2. $nt(i)$ = tamaño de la clase i dentro del nodo t: ésto es calculado en base a la lista de objetos previos que fueron asignados al nodo t: $n_t(i) = \{k / k \in nodo(t)\ y\ k \in C_i\}$.

3. La última columna indica la clase mayoritaria de acuerdo a las probabilidades condicionales $[d(t) = i] \Leftrightarrow [P_t(i) > P_t(l), \forall l \neq i]$.
4. Computamos el tamaño total de los nodos $n(t)$ y la probabilidad $p(t)$. Estas cantidades se calculan de la siguiente forma:

$$p(t) = \sum_{k=1}^{n} p_t(t) * p(k) = \frac{1}{n} * \sum_{k=1}^{n} p_k(t)$$

$$n(t) = \sum_{i=1}^{m} n_t(i)$$

Asumimos que cada objeto tiene un peso igual a $p(k)=1/n$.

Condición de parada (paso (1))

- El árbol es demasiado grande: $card(T_C) > umbral\ max$
- No hay más separaciones posibles $T_A = \varnothing$. Esto ocurre cuando los nodos terminales son solamente puros, todos los individuos pertenecen a una única clase, o el tamaño es demasiado pequeño.

Capitulo VI

Limpieza de los datos

6.1 Descripción y análisis de los datos

Aplicaremos las metodologías antes descriptas a la Encuesta Permanente de Hogares (EPH). Ésta consta de dos cuestionarios:

- Un *cuestionario familiar* con datos de la vivienda y características del hogar. La información correspondiente se presenta en el archivo *HOG_BUA.DBF*.
- Un *cuestionario individual* con datos laborales, de ingresos, de educación y de migración de cada uno de los componentes del hogar. La información correspondiente se presenta en el archivo *PER_BUA.DBF*.

En general, no se encontró demasiada suciedad en los datos, el motivo es que la base de datos fue especialmente construida para analizar su contenido, con lo cual, no fueron encontrados errores de tipeo, duplicidad, etc.Sin embargo se tuvieron en cuenta los siguientes puntos:

- Eliminar aquellos atributos con información redundante. Esto es, aquellos atributos cuya información esté contenida en otro.
- Eliminar aquellos atributos que tuvieran poca información. Hay atributos que, para la mayoría de los registros tienen el mismo valor, con lo cual dichos atributos, no aportan demasiada información para el análisis.
- Eliminar aquellos atributos que dieran una información demasiada detallada, ya que, se desea realizar un análisis general.
- Identificar aquellos registros que contenían información inconsistente, y eliminarlos o modificarlos, según la situación.

6.1.1 Archivos a Procesar

Se analizarán los archivos a procesar por el *See5* y por el *Sodas*, para el primer archivo cada registro representará la información de una persona, mientras que para el segundo cada registro representará la información de una familia.

6.1.1.1 Archivo a procesar por el *See5*

Para una mejor comprensión, el archivo será dividido lógicamente en partes, agrupando los campos que correspondan a un mismo tema. En los Anexos I y II se encontrará información detallada de todas las variables.

Campo CLAVE

Nombre	Tipo	Descripción
ID_PERSONA	Continua	Código de identificación de cada persona

Campos con información de la vivienda

Nombre	Tipo	Descripción
H_P01-H_P02	Categórica	Indica el tipo de vivienda y la cantidad de habitaciones de la misma (* 44)
HP06A-HP06B-HP06C-HP06D	Categórica	Indica las características del baño (* 46)
H_P07-H_P08	Categórica	Indica las características de la vivienda (* 46)

Campos con información de la familia

Nombre	Tipo	Descripción
H_R01_CAT	Categórica	Cantidad de personas en el hogar (**47)
H_ITF_CAT	Categórica	Ingreso Total Familiar (**47)
H_IPCF_CAT	Categórica	Ingreso Per Cápita Familiar (**47)
H_MEN14_CAT	Categórica	Cantidad de personas menores de 14 años (**48)
H_CAT60_CAT	Categórica	Cantidad de personas mayores de 60 (**48)
HOGAR1	Categórica	Indica las características del hogar

Campos con información personal

Nombre	Tipo	Descripción
H08	Categórica	Indica la relación de parentezco
H12_CAT	Categórica	Edad de las personas (**48)
H13	Categórica	Sexo de las personas
H14	Categórica	Estado civil de las personas
P11	Categórica	Indica si la persona es jubilado, rentista, estudiante, etc.
ESTADO	Categórica	Indica si la persona es ocupada, desocupada, etc.

Campos con información ocupacional

Nombre	Tipo	Descripción
P17	Categórica	Relación en la ocupación, patrón, obrero, etc.
P18	Categórica	A que se dedica
P18B	Categórica	Tipo de establecimiento, privado, público, etc.
P19	Categórica	Cantidad de personas que trabajan en el lugar
P20_A	Categórica	Datos de la ocupación.
P20_B	Categórica	Datos de la ocupación.
P20_C	Categórica	Datos de la ocupación.
P22_M_CAT	Categórica	Tiempo que hace que está trabajando en el lugar (**51)
P23	Categórica	Condiciones de Trabajo
P24	Categórica	Condiciones de Trabajo

Campos con información de desocupado

Nombre	Tipo	Descripción
P32D	Categórica	Tiempo que hace que está buscando trabajo (**51)
P41	Categórica	Trabajo anterior

Campos con información de ingresos

Nombre	Tipo	Descripción
CODINGRE	Categórica	Declaraciones de ingreso
P47T_CAT	Categórica	Ingreso Total (** 52)

Campos con información de educación

Nombre	Tipo	Descripción
P54	Categórica	Lee y escribe
P55	Categórica	Etapa escolar, si asiste o asistió al colegio
P56	Categórica	Nivel de estudio
P58	Categórica	Finalizó o no el estudio
P58B	Categórica	Año de estudio en el que se encuentra

Campos con información de migraciones

Nombre	Tipo	Descripción
P59	Categórica	Lugar de nacimiento
P59COD	Categórica	Código de provincia o país

Campo correspondiente a la clase

Nombre	Tipo	Descripción
AGLOMERADO	Categórica	Aglomerado o región en donde vive la persona

(*) La variable fue combinada, en la pag. que se indica junto al * se encuentra su explicación.
(**) La variable fue categorizada, en la pag. que se indica junto a los ** se encuentra su explicación.

Cantidad total de registros	Cantidad total de atributos
11911	38

6.1.1.2 Archivo a procesar por el *Sodas*

Dividiremos también lógicamente éste archivo, para su mejor comprensión. En los Anexos I y II se encontrará información detallada de todas las variables.

Campo CLAVE

Nombre	Tipo	Descripción
ID_Familia	Continua	Código de identificación de cada familia

Campos con información de la vivienda

Nombre	Tipo	Descripción
H_P01	Categórica	Indica el tipo de vivienda
H_P02	Categórica	Indica la cantidad de habitaciones de la vivienda (**44)

HP06A	Categórica	Indica si tiene o no baño
HP06B	Categórica	Indica el tipo de desagüe del baño
HP06C	Categórica	Indica el tipo de conexión
HP06D	Categórica	Indica si el uso es exclusivo o compartido
H_P07	Categórica	Indica el régimen de tenencia de la vivienda
H_P08	Categórica	Indica el tipo de material de la vivienda

Campos con información de la familia

Nombre	Tipo	Descripción
H_R01_CAT	Categórica	Cantidad de personas en el hogar (**47)
H_ITF_CAT	Categórica	Ingreso Total Familiar (**47)
H_IPCF_CAT	Categórica	Ingreso Per Cápita Familiar (**47)
H_MEN14_CAT	Categórica	Cantidad de personas menores de 14 años (**48)
H_CAT60_CAT	Categórica	Cantidad de personas mayores de 60 (**48)
HOGAR1	Categórica	Indica las características del hogar

Campos con información personal

Nombre	Tipo	Descripción
H08	Multivaluada	Indica la distribución de parentezco dentro de la familia
H12_CAT	Multivaluada	Indica la distribución de las edad de la familia (**49)
H13	Multivaluada	Indica la distribución de sexo de la familia
H14	Multivaluada	Indica la distribución de estado civil de la familia
P11	Multivaluada	Indica la distribución de jubilados, rentistas, estudiantes, etc. de la familia
ESTADO	Multivaluada	Indica la distribución de ocupados, desocupados, etc. de la familia

Campos con información ocupacional

Nombre	Tipo	Descripción
P17	Multivaluada	Relación en la ocupación, patrón, obrero, etc.
P18	Multivaluada	A que se dedica
P18B	Multivaluada	Tipo de establecimiento, privado, público, etc.
P19	Multivaluada	Cantidad de personas que trabajan en el lugar
P20_A	Multivaluada	Datos de la ocupación.
P20_B	Multivaluada	Datos de la ocupación.
P20_C	Multivaluada	Datos de la ocupación.
P22M_CAT	Multivaluada	Tiempo que hace que está trabajando en el lugar (**51)
P23	Multivaluada	Condiciones de Trabajo
P24	Multivaluada	Condiciones de Trabajo

Campos con información de desocupado

Nombre	Tipo	Descripción
P32D	Multivaluada	Tiempo que hace que está buscando trabajo (**51)
P41	Multivaluada	Trabajo anterior

Campos con información de ingresos

Nombre	Tipo	Descripción
CODINGRE	Multivaluada	Declaraciones de ingreso
P47T_CAT	Multivaluada	Ingreso Total (** 52)

Campos con información de educación

Nombre	Tipo	Descripción
P54	Multivaluada	Lee y escribe
P55	Multivaluada	Etapa escolar, si asiste o asistió al colegio
P56	Multivaluada	Nivel de estudio
P58	Multivaluada	Finalizó o no el estudio
P58B	Multivaluada	Año de estudio en el que se encuentra

Campos con información de migraciones

Nombre	Tipo	Descripción
P59	Multivaluada	Distribución del lugar de nacimiento de la familia
P59COD	Multivaluada	Distribución del código de provincia o país de la familia

Campo correspondiente a la clase

Nombre	Tipo	Descripción
AGLOMERADO	Categórica	Aglomerado o región en donde vive la familia

(*) La variable fue combinada, en la pag. que se indica junto al * se encuentra su explicación.
(**) La variable fue categorizada, en la pag. que se indica junto a los ** se encuentra su explicación.

Cantidad total de registros	Cantidad total de atributos
3567	44

6.1.1 Limpieza del archivo de Familia (HOG_BUA.DBF)

A continuación, se muestra el análisis realizado para cada una de las variables correspondiente al archivo. En el Apéndice A, se encuentra su diccionario o diseño de registro.

- REALIZADA y RAZONUSU: se **eliminan** debido a que sólo se analizarán los hogares en donde se hayan podido realizar las encuestas.

```
REALIZADA       si se realizó la entrevista al hogar
                0 = no
                1  =si

RAZONUSU        causa de no respuesta de vivienda (preg. 22 del cuestionario)
                1 = Vivienda encuestable
                2 = Vivienda NO encuestable
```

Realizada	Razonusu	Cantidad
0	1	452
0	2	440
1		3567

Por lo tanto se analizarán 3567 familias de un total de 4459 familias.

- ONDA y AÑO: se **eliminan** debido a que todos los registros tienen valores '3' y '98' respectivamente.

- AGLOMERADO: se **conserva,** aunque sólo toma los valores 32 (Cdad. Bs. As. Dominio GBA) y 33 (Partidos Dominio GBA).

Onda	Año	Aglomerado	Cantidad
3	98	32	1432
3	98	33	3027

- P01 y P02: se **conservan,** sus valores son combinados para el *See5* en una única variable, para el *Tree* se mantienen separadas. Quedando definida la variable P01-P02 de la siguiente forma

Descripción	P01-P02
Casa, con 1 habitación	**1-1**
Casa, con 2 habitaciones	**1-2**
Casa, con 3 habitaciones	**1-3**
Casa, con 4 habitaciones	**1-4**
Casa, con 5 habitaciones	**1-5**
Casa, con 6 habitaciones	**1-6**
Casa, con 7 habitaciones	**1-7**
Casa, con 8 habitaciones	**1-8**
Casa, con 9 habitaciones	**1-9**
Casa, con 10 habitaciones	**1-10**
Casa, habit. No Sabe/No responde	**1-99**
Departamento, con 1 habitación	**2-1**
Departamento, con 2 habitaciones	**2-2**
Departamento, con 3 habitaciones	**2-3**
Departamento, con 4 habitaciones	**2-4**
Departamento, con 5 habitaciones	**2-5**
Departamento, con 6 habitaciones	**2-6**
Departamento, con 7 habitaciones	**2-7**
Departamento, con 8 habitaciones	**2-8**
Departamento, con 9 habitaciones	**2-9**
Vive en el trabajo, con 1 habitación	**3-1**
Vive en el trabajo, con 2 habitaciones	**3-2**
Inquilino, con 1 habitación	**4-1**
Inquilino, con 2 habitaciones	**4-2**
Inquilino, con 3 habitaciones	**4-3**
Inquilino, con 4 habitaciones	**4-4**
Inquilino, con 8 habitaciones	**4-8**
Hotel o pensión, con 1 habitación	**5-1**
Vivienda no destinada a fines habit., con 1 habitación	**6-1**
Vivienda en villa, con 1 habitación	**7-1**
Vivienda en villa, con 2 habitaciones	**7-2**
Vivienda en villa, con 3 habitación	**7-3**

- P03: se **elimina** en general tiene el mismo valor que P02.

```
P01             tipo de vivienda.
                1 = Casa
                2 = Departamento
                3 = Vivienda en lugar de trabajo
                4 = Inquilinato
                5 = Hotel o pensi6n
                6 = Vivienda no destinada a fines habita.
                7 = Vivienda en villa
                8 = Otro

P02             cantidad de habitaciones que tiene la vivienda (excluido baño y cocina).

P03             habitaciones de uso exclusivo del hogar. (esta variable será eliminada ya
                que posee la misma distribución que P02).
```

P01	P02	P03	Cantidad	
0	0	0	892	Hog. no encuest.
1	1	1	167	
1	2	1	6	
1	2	2	474	
1	3	1	4	
1	3	2	5	
1	3	3	927	
1	4	1	2	
1	4	2	5	
1	4	3	1	
1	4	4	408	
1	5	1	4	
1	5	2	3	
1	5	3	5	
1	5	5	117	
1	6	4	1	
1	6	6	45	
1	7	2	1	
1	7	5	1	
1	7	7	15	
1	8	8	4	
1	10	10	1	
1	99	99	2	
2	1	1	112	
2	2	1	2	
2	2	2	419	

P01	P02	P03	Cantidad	
2	3	1	2	
2	3	2	2	
2	3	3	501	
2	4	1	1	
2	4	2	1	
2	4	4	193	
2	5	5	59	
2	6	3	2	
2	6	6	13	
2	7	7	6	
2	8	8	4	
2	9	9	1	
3	1	1	2	
3	2	2	1	
4	1	1	11	
4	2	2	3	
4	3	3	4	
4	4	4	1	
4	8	4	1	
5	1	1	3	
6	1	1	2	
7	1	1	8	
7	2	2	6	
7	3	1	1	
7	3	2	1	
7	3	3	7	

- P04 y P05: se **eliminan** debido a que el 98% de los hogares tienen luz y agua.

```
P04             instalación de agua.
                0 = no
                1 = si

P05             instalación de electricidad
                1 = sí
                2 = no
```

P04	P05	Cantidad	Frecuencia
1	1	3512	0,98
1	2	2	0,0005
2	1	50	0,014
2	2	3	0.0008

- P06A, P06B, P06C, P06D: se las **conservan**, para el *See5* se combinan en un único atributo que contenga la información de todas ellas P06A_P06B_P06C_P06D (sus valores se encuentran en la tabla).

```
P06A        instalación de baño
            1 = tiene
            2 = no tiene

P06B        el baño tiene
            1 = inodoro con botón o cadena con arrastre de agua
            2 = inodoro sin botón o cadena con arrastre de agua
            3 = letrina

P06C        el desague es
            1 = a red pública o cloaca
            2 = a cámara escéptica u pozo ciego
            3 = solo a pozo ciego

P06D        el baño es de
            1 = de uso exclusivo del hogar
            2 = compartido con otro hogar
```

P06A	P06B	P06C	P06D	Cantidad	P06A_P06B_P06C_P06D
1	1	1	1	1883	**1-1-1-1**
1	1	1	2	29	**1-1-1-2**
1	1	2	1	793	**1-1-2-1**
1	1	2	2	15	**1-1-2-2**
1	1	3	1	293	**1-1-3-1**
1	1	3	2	15	**1-1-3-2**
1	1	9	1	2	**1-1-9-1**
1	2	1	1	14	**1-2-1-1**
1	2	1	2	3	**1-2-1-2**
1	2	2	1	118	**1-2-2-1**
1	2	2	2	7	**1-2-2-2**
1	2	3	1	166	**1-2-3-1**
1	2	3	2	15	**1-2-3-2**
1	3	0	0	87	**1-3-0-0**
2	0	0	0	127	**2-0-0-0**

- P07 y P08: se **conservan**, para el *See5* se reemplazan por una única variable que contiene el valor de ambas (P07_P08).

```
P07         régimen de tenencia de la vivienda
            1 = propietario de la vivienda y el terreno
            2 = propietario de la vivienda solamente
            3 = inquilino o arrendatario de la vivienda
            4 = ocupante con relación de dependencia
            5 = ocupante gratuito
            8 = otros

P08         tipo de materiales de la vivienda.(predominantes de paredes externas)
            1 = mampostería(ladrillo, bloques, paneles, etc.)
            2 = madera
            3 = metal o fibrocemento.(chapas lisas o fibrocemento)
```

```
4 = adobe
5 = chorizo, cartón o desechos
8 = otros
```

P07	P08	Cantidad	P07_P08
1	1	2526	**1-1**
1	2	38	**1-2**
1	3	4	**1-3**
2	1	129	**2-1**
2	2	26	**2-2**
2	3	3	**2-3**
3	1	531	**3-1**
3	2	9	**3-2**

P07	P08	Cantidad	P07_P08
3	3	3	**3-3**
4	1	39	**4-1**
5	1	202	**5-1**
5	2	27	**5-2**
5	3	3	**5-3**
5	8	1	**5-8**
8	1	26	**8-1**

- H_R01(Cantidad de personas en el hogar): se **conserva**. Los valores de sus atributos son discretizados y la nueva variable es H_R01CAT. La siguiente tabla muestra los valores de las variables H_R01 su cantidad y H_R01CAT.

H_R01	Cantidad	H_R01CAT
1	563	**1**
2	1624	**2**
3	2001	**3**
4	2848	**4**
5	2040	**5**
6	1326	**6**
7	623	**7-8**
8	312	**7-8**
9	225	**9-10**
10	130	**9-10**
11	88	**11-12-13-14**
12	48	**11-12-13-14**
13	13	**11-12-13-14**
14	70	**11-12-13-14**

- H_ITF, DECIF, H_IPCF, DECCF, H_MEN14, H_CAT60: se **conservan**.
  Las variables ITF, IPCF, MEN14, CAT60 son discretizadas, a continuación se muestran los cuadros que indican *variable*, *cantidad* y *variable discretizada*.

H_ITF	Cantidad	H_ITFCAT
0	1256	**0**
(0 – 200]	555	**0-200**
(200 - 500]	2374	**200-500**
(500 - 800]	2367	**500-800**
(800 - 1000]	1009	**800-1000**
(1000 - 1500]	1766	**100-1500**
(1500 - 2000]	968	**1500-2000**
(2000 - 4000]	1254	**2000-4000**
(4000 - 17000]	362	**4000-17000**

H_IPCF	Cantidad	H_IPCFCAT
0	1256	**0**
(0 – 100]	3482	**0-100**
(100 - 300]	4632	**100-300**
(300 - 600]	2264	**300-600**
(600 - 1000]	1008	**600-1000**
(1000 - 2000]	424	**1000-2000**
(2000 - 4000]	83	**2000-4000**
(4000 - 9000]	18	**4000-9000**

H_MEN14	Cantidad	H_MEN14CAT
0	5104	**0**
1	2646	**1**
2	2050	**2**
3	1081	**3**
4	514	**4**
5	284	**5**
6	130	**6**
7-8-9	102	**7-8-9**

H_CAT60	Cantidad	H_CAT60CAT
1	2385	**1**
2	927	**2**
3	8599	**3**

La siguiente tabla muestra la relación entre el *Ingreso Total Familiar* y el *Decil de Ingreso Total Familiar.*

ITF Max	DECIF	Cantidad
0	12	1131
0	0	125
250	1	789
400	2	900
500	3	1085
620	4	1081
800	5	1116
1000	6	1136
1280	7	1141
1700	8	1119
2500	9	1183
16800	10	1105

La siguiente tabla muestra la relación entre el *Ingreso Per Cápita Familiar* y el *Decil de Ingreso Per Cápita Familiar.*

Max	DECC	Cantidad
87.5	1	1698
133.64	2	1375
175	3	1192
220	4	1093
280	5	1051
353.33	6	933
450	7	945
608	8	846
950	9	857
0	00	125
8400	10	665
0	12	1131

- R02, R03, R4, R5, R6, DECIND, DECIF_DOM, DECCF_DOM, AREASNUE y PERCEPT: se **eliminan** debido a que no tienen demasiada información.

6.1.2 Limpieza del archivo de Personas (PER_BUA.DBF)

- ONDA y AGLOMERADO: se **eliminan** debido a que se encuentran en el archivo HOG_BUA.DBF.

- H11(Fecha de nacimiento): se **elimina** debido a que contiene detalladamente la información proporcionada por H12 (Años cumplidos). Esta última es discretizada, la siguiente tabla muestra los valores de la nueva variable discreta H12CAT.

H12	Cantidad	H12CAT
0-6		**0-6**
7-12		**7-12**
13-21		**13-21**
22-30		**22-30**
31-50		**31-50**
51-65		**51-65**
66-80		**66-80**
81-150		**90-150**

- H15 (encuesta realizada) y H17 (razón por la que no fue encuestado): se **eliminan** porque todos los individuos tienen valor 1.

- Se analizaron las siguientes variables:

```
P01     Ha trabajado en la semana. ?
        1 = si; 2 = no

P02     Recibe algún pago por su trabajo?
        1 = si; 2 = no

P03     Ha trabajado... ?
        1=...  menos de 15 horas
        2=...  15 o más horas

P04     Aunque no haya trabajado, tenía Ud. alguna ocupación?
        1 = si; 2 = no

P05     No trabajó en su ocup. durante la sem. de referencia por ...
        1 = suspensión (asalariados)
        2 = falta de trabajo (cta. propia)
        3 = enfermedad
        4 = huelga
        5 = vacaciones o licencia
        8 = otros

P06     Es la suspensión?
        1 = menor de 1 mes
        2 = de 1 a 3 meses
        3 = mayor de 3 meses

P07     Ha buscado trabajo en la semana del....al....
        1 = si; 2 = no

P08     No buscó....
        1 = porque no quiere trabajar
        2 = por estar enfermo
        3 = por tener trabajo asegurado
        4 = porque cree no poder encontrarlo en esa semana.
        5 = porque espera contestación de un trabajo futuro
        7 = por causas momentáneas
        8 = por otras razones
```

```
P09             Desearía Ud. trabajar
                1 = si; 2 = no

P10             Desearía trabajar ...
                1 = menos de 15 horas
                2 = 15 o más horas

P11             Es Ud ...
                1 = jubilado o pensionado
                2 = rentista
                3 = estudiante
                4 = ama de casa
                5 = menor de 6 años
                6 = incapacitado
                8 = otros

ESTADO
                1 = ocupado
                2 = desocupado
                3 = inactivo
                0 = desconocido
```

ESTAD	P01	P02	P03	P04	P05	P06	P07	P08	P09	P11	Cantida	
1	1	1	0	0	0	0	0	0	0	0	4480	Trabajadores
1	1	2	2	0	0	0	0	0	0	0	51	Trab. no pagos
1	2	0	0	1	1	1	0	0	0	0	8	Trab. Suspendidos < 1 mes
1	2	0	0	1	3	0	0	0	0	0	40	Trab. Enfermos
1	2	0	0	1	5	0	0	0	0	0	45	Trab. Vacaciones o licenc.
1	2	0	0	1	8	0	0	0	0	0	24	Trab. No trabj en la semana
2	1	2	1	0	0	0	1	0	0	0	3	Desocupado. Trabajaron
2	1	2	1	0	0	0	2	5	0	0	1	Desocupado. Trabajaron
2	2	0	0	1	2	0	1	0	0	0	22	Desocupado. Cta ppia.
2	2	0	0	1	2	0	2	3	0	0	3	Desocupado. Cta ppia
2	2	0	0	1	2	0	2	5	0	0	2	Desocupado. Cta ppia
2	2	0	0	2	0	0	1	0	0	0	643	Desocupado.
2	2	0	0	2	0	0	2	2	0	0	7	Desocupados enfermos
2	2	0	0	2	0	0	2	3	0	0	6	Sin ocupación, trabajo
2	2	0	0	2	0	0	2	4	0	0	9	Desocupado no busca.
2	2	0	0	2	0	0	2	5	0	0	14	Desocupado espera rta.
2	2	0	0	2	0	0	2	7	0	0	17	Desocupado no busca.
3	1	2	1	0	0	0	2	1	0	4	4	Inactivas amas de casa trab <
3	1	2	1	0	0	0	2	1	0	6	1	Inactivos discapacitados trab
3	1	2	1	0	0	0	2	8	1	3	1	Inactivos estudiantes
3	1	2	1	0	0	0	2	8	1	8	1	Inactivos otros
3	2	0	0	1	2	0	2	1	0	3	1	Inactivos estudiantes (trabajó)
3	2	0	0	2	0	0	2	1	0	1	1054	Jubilados.
3	2	0	0	2	0	0	2	1	0	2	14	Rentista.
3	2	0	0	2	0	0	2	1	0	3	2659	Estudiante.
3	2	0	0	2	0	0	2	1	0	4	1398	Ama de casa
3	2	0	0	2	0	0	2	1	0	5	1123	Menor de 6 años
3	2	0	0	2	0	0	2	1	0	6	77	Inactivos discapacitados
3	2	0	0	2	0	0	2	1	0	8	119	Inactivos por otros motivos
3	2	0	0	2	0	0	2	8	1	1	10	Jubilado quiere trab.
3	2	0	0	2	0	0	2	8	1	2	1	Rentista quiere trab.
3	2	0	0	2	0	0	2	8	1	3	9	Estudiante quiere trab.
3	2	0	0	2	0	0	2	8	1	4	30	Ama de casa quiere trab.
3	2	0	0	2	0	0	2	8	1	6	2	Discapacitado quiere trab.
3	2	0	0	2	0	0	2	8	1	8	24	Inactivo Sin información

ESTAD	P01	P02	P03	P04	P05	P06	P07	P08	P09	P11	Cantida	
3	2	0	0	2	0	0	2	8	2	4	3	Inactivo Sin información
3	2	0	0	2	0	0	2	8	2	8	5	Inactivo Sin información

Después de analizar el cuadro se decidió **eliminar** las siguientes variables:

P03: sólo 11 personas han trabajado menos de 15 horas.
P06: el único valor que toma es 1 y solo para 8 individuos.
P07: la variable P08 contiene ésta información en forma más detallada.
P09: pocas veces toma valor distinto de 0 y si lo hace no brinda demasiada información.

Bloque de Ocupados

El total de registros correspondientes al bloque de ocupados, cuyo valor en el atributo ESTADO=1, son 4648. El análisis se realizó sobre estos registros.

- P12: se **elimina**. El 90% de los ocupados tienen una única ocupación.

- P13AUS: se **elimina**. La mayoría de los registros no tienen valor.

- P14P (horas extras de la ocup. Ppal. En la semana de ref.): se **elimina**, debido a que 4554 individuos ocupados tienen valor 0 (0,97%).

- P14S (horas extras de la ocup. Sec. En la semana de ref.), P14O (horas extras de la otras ocup. En la semana de ref.)y P14T(horas extras total En la semana de ref.): Se **eliminan** las variables, debido a que no contienen información alguna.

- P15P (Total hs trab + extras sem.de ref/ ocup. ppal.), P15S (Total hs trab + extras sem.de ref/ ocup.sec.) P150 (Total hs trab + extras sem.de ref/ otras ocup.) y P15T (Total hs trab + horas extras sem.de ref). se **eliminan**, brindan información demasiado detallada.

- P16, P16B: se **eliminan**.

- P18(A que se dedica o que produce el establecimiento donde trabaja), P18B (Tipo de establecimiento: público, privado, etc): se **conservan**.

- P19: se **conserva.**

```
P19          Cuántas personas trab. en ese estab.
             1 = 1; 2 = 2 a 5; 3 = 6 a 15; 4 = 16 a 25
             5 = 26 a 50; 6 = 51 a 100; 7 = 101 a 500
             8 = 501 o más; 9 = no sabe
```

- P19B: se **elimina**, se considera información demasiado detallada.

```
Para P19 = 9 se le consulta si sabe si,  en ese establecimiento trabajan
                  1 = hasta 40 personas
                  2 = más de 40 personas
                  9 = no sabe
```

- P20 (Cuál es el nombre de su ocupación y que tarea realiza en ella): esta variable consta de 3 dígitos donde cada uno de ellos tiene un significado diferente, se **separa** la variable en 20_A, 20_B y 20_C, con la información de cada uno de ellos.

- P21, P21D: se **conservan**.

- P22 y P22M: se **unifican**. Ambas indican el tiempo que hace que una persona se encuentra en su ocupación, en años y mese respectivamente. Se pasan todos los valores a meses y se deja únicamente la variable P22M. Además la variable P22M es discretizada, en el siguiente cuadro se muestra la *variable*, su cantidad y la *variable discretizada.*

P22M	Cantidad	P22M_CAT
0	7263	**0**
0-12	1440	**1-12**
13-24	537	**13-24**
25-36	370	**23-36**
37-60	529	**37-60**
61-84	347	**61-84**
85-120	412	**85-120**
121-240	612	**121-240**
241-360	262	**241-360**
361-1500	139	**361-1500**

- P23 y P24: se **conserva**.

- P24_2_M y P24_2_D: se **eliminan**, Debido a que el 85% tiene trabajo permanente (P24=1) y el 10% no indica duración (P24=4)

- P29, P30, P31: se **eliminan**, porque brindan información demasiado detallada.

Bloque de Desocupado

- P32 y P32D: se **unifican**. Indican el tiempo que hace que está buscando trabajo en meses y días respectivamente. Además la variable P32D es discretizada como se muestra en el siguiente cuadro.

P32D	Cantidad	P32D_CAT
0	11183	**0**
1-7	29	**1-7**
8-15	68	**8-15**
16-30	138	**16-30**
31-60	102	**31-60**
61-180	170	**61-180**
181-360	148	**181-360**
361-3070	73	**361-3070**

- P41: se **conserva**. Indica la tarea que realizaba en la última ocupación.

- El resto de las variables se **eliminan** porque la mayoría de los datos no contienen información.

Bloque de Educación

- P54, P55, P56: se **conservan**.

- P57: se **elimina**, debido a que no se tuvo un criterio unificado al ingresar los datos.

```
P57:        Cual es la carrera o especial. que cursa o curso (variable sin codificar)
```

- P58 y P58B: se **conservan**.

Bloque de Ingresos

- P47_X y P48_X (X = 1,2,3.....,9): se **eliminan**, debido a que dan información muy detallada sobre el ingreso.

- P47T y CODINGRE: se **conservan**. La variable P47T es discretizada como se puede observar en el siguiente cuadro.

```
P47T        MONTO INGRESOS TOTALES (sumatoria de montos de P47 y P48)

            Recodificación del P47 y P48
CODINGRE    1=tiene y declara monto
            2=no tiene ingreso en ambas
            9=tiene ingreso y no declara monto en alguna
```

P47T	Cantidad	P47TCAT
0	6292	**0**
(0-100]	299	**0-100**
(100-300]	1521	**100-300**
(300-600]	1883	**300-600**
(600-1000]	1049	**600-1000**
(1000-2000]	632	**1000-2000**
(2000-4000]	191	**2000-4000**
(4000-15000]	44	**4000-15000**

Boque de Migraciones

- P59 y P59COD: se **conservan**.

```
P59         Dónde nació
            1=en esta ciúdad
            2=en otro lugar de esta Pcia.
            3=en otra Pcia.
            4=en otro país
            9=No responde

P59COD      Código de provincia o país
```

- El resto se **eliminan** debido a que los datos no fueron completados para la mayoría de los registros.

Capítulo VII

Evaluación de los algoritmos

7.1 Construcción de los conjuntos de datos

Una vez realizada la limpieza de los datos (Ver Capítulo VI), éstos fueron procesados con el *See5* y el *SODAS*. La idea fue encontrar las características de cada uno de los aglomerados, se tomó entonces como atributo *clase* el *aglomerado*.

Para procesar los datos se dividió al conjunto total de datos (11.911 registros) en 5 partes aproximadamente iguales, esto se hizo para tener una mejor estimación del error[1]. Para realizar la división se tuvo en cuenta que cada uno de los conjuntos de datos sería usado tanto por el algoritmo *"See5"* como por el *"Tree"*, recordemos que este último agrupará los datos a nivel familia, es por eso que en la división de los datos se mantienen los registros correspondientes a una misma familia en el mismo bloque. Además se mantuvo la proporción de cada una de las clases para cada división, el siguiente cuadro muestra los resultados de la división.

Nro. Bloque	Cantidad de registros	Cantidad de Familias	Cantidad de Aglom. 32	%	Cantidad de Aglom. 33	%
1	2.397	716	192	27%	524	73%
2	2.379	714	196	27%	518	73%
3	2.398	714	179	25%	535	75%
4	2.391	699	215	30%	484	70%
5	2.346	711	169	24%	542	76%
Total	**11.911**	**3.554**	**951**	**27%**	**2.603**	**73%**

Tabla 17: Bloques a procesar

Para cada uno de los bloques se separó el conjunto de datos en un porcentaje[2] fijo de 2/3 de los datos para el conjunto de entrenamiento y 1/3 de los datos para el conjunto de testeo.

Para el algoritmo *"Tree"* el conjunto de testeo es seleccionado automáticamente, sólo es necesario indicarle como parámetro qué porcentaje es el que se desea utilizar para testear el árbol que construye. En el *"See5"* es necesario, armar por separado el conjunto de datos de testeo, con lo cual para cada uno de los bloques se creó su conjunto de testeo con 1/3 de los datos, manteniendo el porcentaje de las clases (aglomerados 32 y 33) para cada uno de ellos.

[1] Ver 2.2.4 Construcción de los conjuntos de datos para estimar el error
[2] Ver 2.2.4 el porcentaje sugerido para el *Holdout*

7.2 Procesamiento con el See5

7.2.1 Resultados del See5.

A continuación se analizarán los resultados obtenidos con el *See5*. El *See5* posee varias opciones parametrizables en el momento de ejecutar el programa, dos opciones importantes son las correspondientes al pospruning y prepruning, ambas fueron utilizadas por nosotros, para el pospruning utilizamos *30%* este valor indica el umbral para calcular el índice de error en el podado. Para el prepruning fijamos en 15 la cantidad mínima de casos que debe haber para al menos en dos ramas de las que se separa el árbol.

Al procesar los cinco grupos se obtuvieron los *árboles de decisión* correspondientes, en este punto, mostraremos la salida correspondiente al primer grupo. Las salidas completas de cada una de las ejecuciones se encuentran en el Apéndice III.

Árbol de decisión del 1er. Grupo:

```
HP06A-HP06B-HP06C-HP06D in {1_2_2_2,1_1_9_1,1_2_1_2}: 33
HP06A-HP06B-HP06C-HP06D = 1_1_2_2: 33
HP06A-HP06B-HP06C-HP06D = 1_1_1_2: 32
HP06A-HP06B-HP06C-HP06D = 1_1_2_1: 33
HP06A-HP06B-HP06C-HP06D = 1_2_2_1: 33
HP06A-HP06B-HP06C-HP06D = 1_1_3_1: 33
HP06A-HP06B-HP06C-HP06D = 1_3_0_0: 33
HP06A-HP06B-HP06C-HP06D = 1_2_1_1: 33
HP06A-HP06B-HP06C-HP06D = 1_2_3_1: 33
HP06A-HP06B-HP06C-HP06D = 1_2_3_2: 33
HP06A-HP06B-HP06C-HP06D = 2_0_0_0: 33
HP06A-HP06B-HP06C-HP06D = 1_1_3_2: 33
HP06A-HP06B-HP06C-HP06D = 1_1_1_1:
|----H_P01-H_P02 in {2_8,4_8,2_7,
|                    2_9,4_1,4_2,
|                    5_1,1_8,7_1,
|                    1_7,6_1,1_9,
|                    7_3,3_1,7_2,
|                    3_2,4_3,2_8}: 33
|
|----H_P01-H_P02 = 1_3: 33
|----H_P01-H_P02 = 2_2: 32
|----H_P01-H_P02 = 1_4: 33
|----H_P01-H_P02 = 2_5: 32
|----H_P01-H_P02 = 1_6: 33
|----H_P01-H_P02 = 1_2: 33
|----H_P01-H_P02 = 2_1: 32
|----H_P01-H_P02 = 1_1: 33
|----H_P01-H_P02 = 2_6: 33
|----H_P01-H_P02 = 4_4: 32
|----H_P01-H_P02 = 2_3:
|           |----H_ITFCAT = 0: 32
|           |----H_ITFCAT = 0_200: 32
|           |----H_ITFCAT = 200_500: 32
|           |----H_ITFCAT = 500_800: 33
|           |----H_ITFCAT = 800_1000: 32
|           |----H_ITFCAT = 1000_1500: 32
```

```
|           |----H_ITFCAT = 1500_2000: 32
|           |----H_ITFCAT = 2000_4000: 32
|           |----H_ITFCAT = 4000_17000: 32
|
|----H_P01-H_P02 = 2_4:
|           |----H_R01CAT in {7_8_,9_10_,11_12_13_14_,14_}: 32
|           |----H_R01CAT = 2_: 33
|           |----H_R01CAT = 1_: 32
|           |----H_R01CAT = 5_: 32
|           |----H_R01CAT = 4_: 32
|           |----H_R01CAT = 3_: 32
|           |----H_R01CAT = 6_: 33
|
|----H_P01-H_P02 = 1_5:
|           |----H_R01CAT in {2_,4_,7_8_,9_10_,11_12_13_14_,14_}: 33
|           |----H_R01CAT = 1_: 33
|           |----H_R01CAT = 5_: 32
|           |----H_R01CAT = 3_: 33
|           |----H_R01CAT = 6_: 33
```

Como se puede observar, el primer nodo evalúa las condiciones del *baño* de la familia, casi todas las personas que no cumplen la condición de tener *"baño con inodoro, con botón o cadena con arrastre de agua a red pública y ser de uso exclusivo del hogar"*(HP06A-HP06B-HP06C-HP06D = 1-1-1-1) pertenecen al aglomerado 33.

Si las personas *"tienen baño con inodoro, con botón o cadena con arrastre de agua a red pública y ser de uso exclusivo del hogar"*, se evalúa entonces el tipo de *vivienda* (HP01-HP02), recordemos que el primero, HP01 indica si es departamento, casa, hotel, villa mientras que el segundo indica la cantidad de habitaciones.

Para ciertos valores de HP01-HP02 es necesario considerar el *Ingreso Total Familiar* (H_ITFCAT) o la *cantidad de personas en el hogar* (H_R01CAT), para poder determinar el aglomerado.

Robustez del algoritmo

A continuación mostramos como interpretamos los resultados de la evaluación del árbol de decisión del primer grupo sobre el conjunto de entrenamiento (1569 registros). Los valores de las evaluaciones para cada uno de los grupos se encuentran en el Apéndice III:

```
Decision Tree
----------------
Size      Errors

  44  158(10.1%)   <<


 (a)   (b)    <-classified as
----  ----
 274    88    (a): class 32
  70  1137    (b): class 33
```

El *"Size"* es el número de hojas y *"Errors"* muestra el número y porcentaje de casos mal clasificados. Es decir, que el árbol de decisión sobre el conjunto de entrenamiento tiene un error de 10.1%.

Como el número de clases es menor o igual que 20, se muestra una matriz de confusión. La matriz de confusión para los *datos de entrenamiento* indica que 274 personas que pertenecen al aglomerado 32 clasifican correctamente mientras que 88 personas de ese aglomerado clasifican incorrectamente, mientras que para el aglomerado 33, 1.137 personas clasifican correctamente y 70 lo hacen incorrectamente.

Observando la matriz de confusión se puede ver que:

En el conjunto de entrenamiento:

- El *total* de casos es **1.569**.
- **362** casos corresponden al aglomerado **32** (23%).
- **1.207** casos corresponden al aglomerado **33** (77%).

Análisis del error mayoritario

El análisis del *error mayoritario* nos dice cual es el máximo porcentaje de error que se debería admitir para aceptar el *árbol de decisión.* Y este porcentaje coincide con el porcentaje de la clase que es minoría en nuestro caso entonces, para el *conjunto de entrenamiento* el error mayoritario es de 23%. Un ejemplo de un árbol de decisión que tenga un error igual al *error mayoritario* es aquel que simplemente clasifique todos los casos como pertenecientes al aglomerado 33, este *árbol de decisión* clasificaría incorrectamente 362 casos es decir que tendría un error de 23%. Por tal motivo, es necesario analizar que el error cometido al construir el árbol de decisión no sea cercano al *error mayoritario.*

A continuación analizamos la robustez del algoritmo, para que un algoritmo sea robusto, los errores producidos sobre los diferentes conjuntos de testeo no deben ser muy diferentes entre sí, es decir el algoritmo debe mantenerse estable a diferentes conjuntos de entrenamiento sobre el mismo conjunto de datos.

Según los resultados de las evaluaciones[3] los errores producidos son los siguientes:

Nro. Grupo	Error	Error Mayoritario
1	13.1%	27%
2	10.1%	27%
3	14.1%	25%
4	14.6%	30%
5	17.5%	24%

Tabla 18: Errores del conjunto de testeo de cada uno de los grupos

Según calculo de la media y desvío estándar, visto en el punto 2.2.5 para éste algoritmo (A_{see5}) tenemos los siguientes valores:

$$media(A_{see5}) = \frac{13.1+10.1+14.1+14.6+17.5}{5} = 13.88$$

$$ds(A_{see5}) = \sqrt{\frac{1}{5(4)}\left((13.1-13.8)^2+(10.1-13.8)^2+(14.1-13.8)^2+(14.6-13.8)^2+(17.1-13.8)^2\right)} = 0.87$$

[3] Las evaluaciones se encuentran al final de cada corrida en el Apéndice III

Por lo tanto tenemos el siguiente error 13.88±0.87. Podemos decir entonces que el algoritmo es robusto y no tiene variaciones en el índice de error para los diferentes conjuntos de entrenamiento.

7.3 Procesamiento con el Tree

7.3.1 Resultados del Tree.

Analizamos ahora el resultado del proceso con el *Tree*. El *Tree* también posee parámetros, este algoritmo posee solamente la opción de *prepruning*, las opciones son establecer la cantidad mínima de casos para separar el nodo, la cantidad mínima de casos de la clase no mayoritaria y la cantidad mínima de casos en los nodos descendiente, tanto derecho como izquierdo, nosotros establecimos para todas esas opciones el valor 20.

A continuación mostramos el árbol de decisión construido para el primer grupo, el resto de los árboles de decisión se encuentran en el Apéndice IV

Árbol de decisión del 1er. Grupo:

```
                +----  [ 8 ]32  (     2.00      0.00 )
                !
          !----4[ H_P07 = 000100 ]
          !     !
          !     !          +----  [ 36 ]33  (    15.00     29.00 )
          !     !          !
          !     !     !---18[ H_R01CAT = 010110000 ]
          !     !     !    !
          !     !     !    +----  [ 37 ]33  (    12.00     49.00 )
          !     !     !
          !     !----9[ 20B = 11110011000 ]
          !          !
          !          +----  [ 19 ]32  (     3.00      0.00 )
          !
     !----2[ HP01 = 010000 ]
     !    !
     !    !          +----  [ 20 ]33  (     4.00     13.00 )
     !    !          !
     !    !     !---10[ HP02 = 100110000 ]
     !    !     !    !
     !    !     !    +----  [ 21 ]32  (    13.00      8.00 )
     !    !     !
     !    !----5[ H_IPCFCAT = 01000100 ]
     !         !
     !         !    +----  [ 22 ]33  (     1.00      3.00 )
     !         !    !
     !         !---11[ H_P07 = 001100 ]
     !              !
     !              +----  [ 23 ]32  (    91.00     13.00 )
     !
!----1[ HP06C = 1000 ]
     !
     +----  [ 3 ]33  (     1.00    225.00 )
```

Como podemos observar en el árbol, es elegido el atributo del *"tipo de desagüe del baño"* como primer nodo HP06 = 1 (*"a red pública"*, para la rama superior) y HP06 = 0,2,3 (para la rama inferior), los valores que toman las dos ramas se encuentran en cada punto de separación de la salida de la corrida (Apéndice IV).

El siguiente atributo por el que se separa la rama superior del árbol es el *"tipo de vivienda"*, quedando separadas las familias que tienen casa (HP01=1) de las que no. Luego en la rama superior se evalúa el *"tipo de tenencia"* (HP07), quedando por un lado los que son propietarios de la vivienda de los que no, los primeros se encuentran en el aglomerado 32, los segundos según el atributo P20B se encuentran en el aglomerado 33 o en el 32.

Continuando por la rama de los que poseen otro tipo de vivienda que no sea casa, se produce una división según el *"ingreso per cápita familiar"* (H_IPCFCAT)entre los que ganan menos de $300 y los que ganan más de $300 para los primeros se produce una división según la *cantidad de habitaciones* (H_P01) si tienen entre 3, 5 o 6 habitaciones se encuentran en el aglomerado 33 sino en el aglomerado 32, para los segundos se fija en el *tipo de tenencia* (H_P07), si son *ocupantes o propietarios de la vivienda solamente*, se encuentran en el aglomerado 33 de lo contrario en el aglomerado 32.

Robustez del algoritmo

La evaluación del algoritmo en el *Tree* es semejante a la del *See5*, muestra una matriz de confusión para cada uno de los conjuntos, de entrenamiento y de testeo.

La siguiente es la matriz de confusión para el conjunto de entrenamiento del primer grupo

```
CONFUSION MATRIX FOR TRAINNING SET

      ========================================
      |      | 32        | 33        |  Total    |
      ========================================
      | 32   |       109 |        33 |       142 |
      | 33   |        21 |       319 |       340 |
      ========================================
      | Total|       130 |       352 |       482 |
      ========================================

   MISCLASSIFICATION RATE BY CLASS

      TRUE CLASS              (  ERROR /SIZE     )   FREQUENCY
      32                      (     33 /    142 )      23.24
      33                      (     21 /    340 )       6.18

      TOTAL                   (     54 /    482 )      11.20
```

A continuación analizamos la robustez del algoritmo, para poder medir la estabilidad del algoritmo sobre los diferentes conjuntos de entrenamiento.

Según los resultados de las evaluaciones[4] los errores producidos son los siguientes

[4] Las evaluaciones se encuentran al final de cada corrida en el Apéndice IV

Nro. Grupo	Error	Error Mayoritario[5]
1	21.37%	27%
2	16.81%	27%
3	17.67%	25%
4	18.94%	30%
5	19.48%	24%

Tabla 19: Errores del conjunto de testeo de cada uno de los grupos

Según el calculo de la media y desvío estándar, visto en el punto 2.2.5 para éste algoritmo (A_{Tree}) tenemos los siguientes valores:

$$media(A_{Tee}) = \frac{21.37 + 16.81 + 17.67 + 18.94 + 19.48}{5} = 18.85$$

$$ds(A_{Tree}) = \sqrt{\frac{1}{5(4)}\left((21.37-18.85)^2 + (16.81-18.85)^2 + (17.67-18.85)^2 + (18.94-18.85)^2 + (19.48-18.85)^2\right)} = 0.78$$

Por lo tanto tenemos el siguiente error 18.85±0.78. Podemos decir entonces que el algoritmo es robusto y no tiene variaciones en el índice de error para los diferentes conjuntos de entrenamiento.

7.4 Comparación de los dos algoritmos

Performance

Si bien podríamos realizar una evaluación de los algoritmos según lo planteado en el punto 2.2.5, deberíamos tener en cuenta que, para cada uno de los bloques en que se dividió el conjunto de datos la cantidad de registros que procesa el *Tree* es mucho menor que la del *See5*.

Nro. Bloque	See5	Tree	Diferencia
1	2.397	716	70%
2	2.379	714	70%
3	2.398	714	70%
4	2.391	699	69%
5	2.346	711	70%
Total	**11.911**	**3.554**	**70%**

Tabla 20: Cantidad de registros para cada uno de los grupos

Si Llamamos A_{see5} al algoritmo generado por el *See5* y A_{Tree} al algoritmo generado por el *Tree*, en el punto anterior habíamos calculado su media y desvío estándar. El error para el A_{see5} es 13.88±0.87, mientras que para el A_{ITee} es 18.85±0.78. Hay una diferencia en el porcentaje de error que favorece al

[5] El Error Mayoritario se calcula de la misma forma en que se explicó en el punto 6.2.1

A_{see5} . Sin embargo esa diferencia es sólo de 4.97 %, recordemos que la diferencia en la cantidad de registros es del 70%, según las medidas el A_{Tree} es más estable que el A_{See5} (0.87 y 0.78).

Representación de los datos

Para el *See5*, cada caso o registro representa a un individuo, con lo cual cada atributo contiene el valor que caracteriza al individuo. Por ejemplo, en nuestro caso cada registro representa a una *Persona*, y los atributos que caracterizan a esa persona pueden ser, *Vivienda, Cantidad de habitantes, Sexo, Edad, Ingreso Familiar, Ingreso Personal, Aglomerado* etc. La siguiente tabla muestra un ejemplo de nuestra base.

Persona	Familia	Vivienda	Habitantes	Sexo	Edad	Ing.Fam.	Ing.Pers	Aglomerado
1	1	Casa	3	M	28	$2.000	$2.000	32
2	1	Casa	3	F	27	$2.000	$0	32
3	1	Casa	3	M	3	$2.000	$0	32
4	2	Depto	2	M	35	$1.500	$800	33
5	2	Depto	2	F	30	$1.500	$700	33
6	3	Depto	3	M	54	$1.000	$500	33
7	3	Depto	3	F	55	$1.000	$500	33
8	4	Casa	2	M	68	$600	$300	32

Tabla 21: Ejemplo de la base de datos a procesar por el *See5*

Por otro lado, para el proceso *"Tree" del SODAS*, cada caso o registro puede representar a un grupo de individuos, con lo cual cada atributo contiene una distribución de valores que caracterizan a ese grupo de individuos. Por ejemplo, en nuestro caso cada registro representaría una *Familia* y los atributos que representan a una *Familia* son los mismos *Vivienda, Cantidad de habitantes, Sexo, Edad, Ingreso Familiar, Ingreso Personal, Aglomerado* etc. Pero sus valores ya no son simples sino que pueden ser distribuciones, rangos etc. La siguiente tabla muestra un ejemplo de nuestra base para el *"Tree"*.

Familia	Vivienda	Hab.	Sexo	Edad	Ing.Fam.	Ing.Pers	Aglomerado
1	Casa	3	M(0.66),F(0.33)	[3:28]	$2.000	$2.000(0.33),$0(0.66)	32
2	Depto	2	M(0.5),F(0.5)	[30:35]	$1.500	$800(0.5),$700(0.5)	33
3	Depto	3	M(0.5),F(0.5)	[54:55]	$1.000	$500(1)	33
4	Casa	2	M(0.5),F(0.5)	[62:68]	$600	$300(1)	32

Tabla 22: Ejemplo de la base de datos a procesar por el Sodas

Como se observa, se redujo considerablemente la cantidad de registros, para la segunda tabla, la reducción de los datos es una de las ventajas que nos da el Análisis de Datos Simbólicos. La Tabla 23 muestra la cantidad de registros que se procesaron para cada uno de los grupos[6]. Sin embargo, esto lleva a procesar los datos con atributos más complejos, esta complejidad hace que el criterio de selección de atributo para la separación de los nodos sea más complejo.

[6] Recordemos que la totalidad de la base fue dividida en 5 partes aproximadamente semejantes.

Nro. Grupo	Cantidad de registros *See5*	Cantidad de registros *Tree*
1	2.397	716
2	2.379	714
3	2.398	714
4	2.391	699
5	2.346	711
Total	**11.911**	**3.554**

Tabla 23: Cantidades de registros a procesar para cada uno de los algoritmos

Otra ventaja importante de la representación de los registros en el Sodas, es que es posible encontrar características en la distribución de ciertos atributos de la familia, por ejemplo las familias podrían tener características diferentes según la distribución del *Ingreso Personal.*

Al realizar un Análisis de datos simbólicos, una elección bastante difícil de hacer es el criterio de agrupación de los datos, por lo general ese criterio debe ser propuesto por la persona experta e interesada en los resultados del análisis.

Tipos de variables

Como pudimos ver en el punto anterior las variables del See5 solamente utilizan valores simples, estos pueden ser nominales o continuos. Mientras que el *Sodas* no solamente utiliza variables simples sino estructuradas. (Ver 4.2).

Por ejemplo, la variable *sexo* en la Tabla 21 tomaba valores *M* (masculino) o *F* (femenino) y en la Tabla 22 la variable sexo es representada por una distribución de valores *M* (0.66) (66% de masculino) por ejemplo.

Criterio de Selección

El criterio que utiliza para seleccionar las ramas por las cuales se va abriendo el árbol el *See5* es el de *gain ratio* el cuál está basado en la *Teoría de Información.* Este fue analizado con suficiente detalle en el punto 3.2.2.1.3. Vimos que en cada paso, trata de seleccionar aquel atributo que mayor información de ganancia tenga y de esa forma va construyendo el árbol de decisión.

Una característica importante de este algoritmo es la cantidad de ramas que genera en cada paso. Dependiendo del tipo de variable, la división se produce en dos, si es continua o en *n* si es nominal, donde *n* es la cantidad de valores diferentes que toma esa variable. Es por esa razón que a veces es conveniente agrupar los atributos para reducir la cantidad de ramas.

Por el otro lado, el criterio que utiliza para seleccionar las ramas el *Tree* es el de *log-likelihood.* Este también fue analizado en detalle en el punto 5.4.3.

A diferencia del *See5* el árbol que construye el *Tree* es un árbol binario. Tanto para las variables continuas como nominales, la división siempre se produce en dos subárboles. Si la variable es continua $[Y_j < c]$, se tiene *n* o *2n-1* posibilidades, donde *n* es la cantidad de valores de Y_j. (Ver 5.3). Si la variable es nominal $[Y_j \in V]$, se tiene 2^{n-1}-1 posibilidades, donde *n* es la cantidad de valores de Y_j. (Ver 5.3). Por este motivo una de las restricciones del *Tree* es que las variables nominales no pueden tomar más de 12 valores diferentes.

7.5 Conclusiones

En este trabajo mostramos que, con los datos que releva un censo, a través de los árboles de decisión es posible clasificar los habitantes de las distintas regiones (Capital, Gran Buenos Aires, Tucumán, La Rioja, etc.). En nuestro caso nos fueron proporcionados únicamente los datos correspondientes al Gran Buenos Aires divididos en dos aglomerados, 32 y 33.

Realizamos el análisis utilizando dos métodos de árboles de decisión, uno construye el árbol a partir de los datos simples y el otro construye el árbol a partir de los datos representados en forma simbólica.

En el *análisis de datos simbólico*, el concepto de *datos simbólicos,* permite tomar como unidad de dato la familia, con lo cuál es la familia a quien clasifica en alguna región, en el *aprendizaje de máquina* simplemente se clasifica a las personas.

Por ejemplo, podríamos tener dos personas con las mismas características (Edad, Sexo, Ingreso per Cápita Familiar, Ocupación, etc.), para el *aprendizaje de máquina*, seguramente esas personas van a clasificar para la misma región, pudiendo pertenecer a diferentes regiones debido al tipo de familia en la que viven. No sucedería lo mismo con el *análisis de datos simbólico*, ya que éste clasificaría a la familia y no a la persona.

Si observamos los resultados obtenidos por el *SODAS* y el *See5*, se puede apreciar que muchos de los atributos que caracterizan a las regiones son los mismos para ambos métodos, podemos decir entonces que, el *tipo de instalación del baño, tipo de vivienda y régimen de tenencia* son atributos muy importante para determinar la región en la que viven, ya que en ambos análisis (*See5 y SODAS*), aparecen en los nodos más alto del árbol.

Una de las desventajas que posee el *SODAS* es quizá, no poder procesar simultáneamente datos continuos y nominales. Es por ese motivo que los atributos continuos fueron discretizados, de tal forma de poder procesar en ambos programas la misma cantidad de atributos.

Podemos resumir las diferencias entre el See5 y el SODAS en el siguiente cuadro:

	See5	SODAS
Representación de los casos	Cada caso o registro representa a un individuo, con lo cual cada atributo contiene el valor que caracteriza al individuo. *Por ejemplo:* Si cada caso es una *Persona* el atributo *Sexo* contiene el sexo de cada una de las personas.	Cada caso o registro puede representar a un grupo de individuos, con lo cual cada atributo contiene una distribución de valores que caracterizan a ese grupo de individuos. *Por ejemplo:* Si cada caso es una *Familia* el atributo *Sexo* contiene la frecuencia de cada uno de los sexos dentro de la familia.
Tipo de variables	Los valores que toman las variables son simples. *Por ejemplo: continuo (edad, ingreso, etc.). categórico (sexo, color, etc.)*	Los valores que toman las variables además de simples pueden ser estructurados. *Por ejemplo: intervalo ([ingreso mín:ingreso máx], frecuencias (rojo(0.3), amarillo(0.5), azul(0.2))*
Criterio de selección	El criterio que utiliza para seleccionar las ramas por las cuales se va abriendo el árbol es el de *gain ratio* el cuál está basado en la *Teoría de Información.* (Ver 2.3.2.3)	El criterio que utiliza para seleccionar las ramas por las cuales se va abriendo el árbol es el de *logaritmo de máxima verosimilitud (log-likelihood).* (Ver 5.4.3)

Tabla 24: Cuadro resumen de las diferencias

Ambos métodos fueron útiles para extraer información de la *Encuesta Permanente de Hogares (EPH)*, en nuestro caso, como quisimos analizar los aglomerados o regiones de dicha encuesta considerando como unidad de información de los aglomerados a la familia, fue interesante analizar los datos mediante el concepto de *datos simbólicos* pudiendo entonces caracterizar a las familias que viven en cada uno de los aglomerados, el concepto de *análisis de datos simbólicos* permite reducir considerablemente la base de datos, en nuestro caso, teníamos una base de 11.911 registros, cada uno representaba los datos de una persona, esta cantidad se redujo a 3.567 registros, cada uno representando a una familia.

Referencias

[ADR/97] ***Adriaans, P. - Zantinge, D.***
"DATA MINING"
Addison – Wesley; U.S.A.;1997.

[BAR/00] ***Baranauskas J. A. – Monard M. C.***
"REVIEWING SOME MACHINE LEARNING CONCEPTS AND METHODS"
Tec. Reports Nro. 102; University of Sao Paulo; Brasil; 2000.

[BOC/00] ***Bock H. – Diday E.***
"ANALYSIS OF SYMBOLIC DATA"
Springer; Francia; 2000.

[BRE/84] ***Breiman L., Friedman J. H., Olshen R. A., and Stone C. J.***
"CLASSIFICATION AND REGRESSION TREES"
Belmont; 1984.

[CLA/96] ***Clark P. – Niblett T.***
"THE CN2 INDUCTION ALGORITHM. MACHINE LEARNING"
261-283; U.S.A.;1996.

[FAY/96] ***Fayyad U. M. , Piatetsky –Shapiro G., Smyth P and Uthurusamy R.***
"ADVANCES IN KNOWLEDGE DISCOVERY AND DATA MINING"
IEEE Transactions on Computers 404-408; U.S.A.; 1977.

[FRI/77] ***Friedman J. H.***
"A RECURSIVE PARTITIONING DECISION RULE FOR NON PARAMETRIC CLASSIFICATION"
AAAI Press/MIT Press; U.S.A.; 1996.

[GRO/98] ***Groth R.***
"DATA MINING"
Prentice Hall; U.S.A.; 1998.

[HUN/66] ***Hunt E. B., Marin J., and Stone P.J.***
"EXPERIMENT IN INDUCCION"
Academic Press; U.S.A.; 1966.

[HYA/76] ***Hyafil L., and Rivest, R. L.***
"CONSTRUCTING OPTIMAL BINARY DECISION TREES IS NP-COMPLETE"
Information Processing Letters 5,1, 15-17; U.S.A.; 1976.

[JOH/98] ***Johnson R.***
"APPLIED MULTIVARIATE STATISTICAL ANALYSIS"
Prentice Hall; 4ta. Edición; U.S.A.; 1998.

[PIA/91] ***Piatetsky – Shapiro G. and Frawley W.***
"KNOWLEDGE DISCOVERY IN DATABASES"
Cambridge MA: AAAI Press/MIT Press; U.S.A.; 1991

[QUI/79] ***Quinlan J. R.***
"DISCOVERING RULES BY INDUCTION FROM LARGE COLLECTIONS OF EXAMPLES"
Edinburgh University Press; U.S.A.; 1979.

[QUI/83] ***Quinlan J. R.***
"A CAUTION APPROACH TO UNCERTAIN INFERENCE"
Computer Journal 26, 3, 255-269; Addison Wesley; U.S.A.; 1986.

[QUI/86] ***Quinlan J. R.***
"INDUCTION OF DECISION TREES"
Morgan Kaufmann; Addison Wesley; U.S.A.; 1986.

[QUI/93] ***Quinlan J. R.***
"C4.5: PROGRAMS FOR MACHINE LEARNING"
Morgan Kaufmann Publishers; U.S.A.; 1992.

[RON/96] ***Kohavi R.– Sommerfield D.***
"MLC++: MACHINE LEARNING IN C++"
http://www.sgi.com/Technology/mlc; U.S.A.; 1996.

[ULL/82] ***Ullman J. D.***
"PRINCIPLES OF DATABASE SYSTEMS"
CA: Computer Science Press; U.S.A.; 1982..

Apéndice I

Datos Personales

Diccionario o Diseño de registro

Campo		Tipo		Descripción
1	CODUSU	Ch	11	Código de hogar para matching
2	COMPONENTE	N	2	Número de componente del hogar
3	ONDA	N	1	Mes de relevamiento 1=abril/mayo 3=octubre/noviembre 4=agosto
5	AGLOMERADO	N	2	Número del Area de relevamiento
7	H08	N	2	Relación de parentesco Ol=jefe; 02=cónyuge; 03=hijo; 04=yerno/nuera; 05=hermano; 06=nieto; 07=cuñado; 08=padre o suegro 09=otros familiares; 10-servicio doméstico 11=otros componentes
8	H11	D	8	Fecha de nacimiento
9	H12	N	2	Años cumplidos -1=menos de 1 año; 98=noventa y ocho Y más; 99=ns/nr edad
10	H13	N	1	Sexo 1=varon; 2=mujer
11	H14	N	1	Estado Civil 1=soltero; 2=unido; 3=casado; 4=separado o divorciado; 5=viudo 9=no responde
12	H15	N	1	Entrevista individual realizada 1=si y 2=no
13	H16			Respondente
14	H17			Razón de no entrevista individual 1=Ausencia 2=Rechazo
15	P01	N	1	Ha trabajado en la semana? 1=si; 2=no
16	P02	N	1	Recibe algún pago por su trabajo? 0=no responde 1=si 2=no
17	P03	N	1	Ha trabajado... ? 1=... menos de 15 horas 2=... 15 o más horas
18	P04	N	1	Aunque no haya trabajado, tenia Ud. alguna ocupación

				1=si; 2=no
19	PO5	N	1	No trabajó en su ocup. durante la semana de referencia por ... 1=suspensión (asalariados) 2=falta de trabajo (cta.propia) 3=enfermedad 4=huelga 5=vacaciones o licencia 8=otros
20	P06	N	1	Es la suspensión 1=menor de 1 mes 2=de 1 a 3 meses 3=mayor de 3 meses
21	P07	N	1	Ha buscado trabajo en la semana del....al.... 1=si; 2=no
22	P08	N	1	No buscó.... 1=porque no quiere trabajar 2=por estar enfermo 3=por tener trabajo asegurado 4=porque cree no poder encontrarlo en esa semana 5=porque espera contestación de un trabajo futuro 7=por causas momentáneas 8=pot otras razones
23	P09	N	1	Desearía Ud. trabajar 1=si; 2=no
24	P10	N	1	Desearía trabajar ... 1=menos de 15 horas 2=15 o más horas
25	P11	N	1	Es Ud. ... 1=jubilado o pensionado 2=rentista 3=estudiante 4=ama de casa 5=menor de 6 años 6=incapacitado 8=otros
26	ESTADO	N	1	1=ocupado 2=desocupado 3=inactivo 0=desconocido

BLOQUE DE OCUPADOS

27	P12	N	1	Cuantas ocupaciones tiene usted 9=no responde
28	P13aus	CH	1	1=Solo si no trabajó algún día de la semana por licencia, suspendido, falta de trabajo, sin dato, otras causas, huelga, no responde.
29	P14P	N	5 1	Horas extras/ocup. principal/sem. de ref.
30	P14S	N	5 1	Horas extras/ocup.secund./sem. de ref
31	P140	N	5 11	Horas extras/otras ocup./sem. de ref
32	P14T	N	5 1	Total horas extras /la sem de ref
33	P15P	N	5 1	Total hs trab + extras sem.de ref/ ocup. ppal. 999=no responde
34	P15S	N	5 1	Total hs trab + extras sem.de ref/ ocup.sec. 999=no responde
35	P150	N	5 1	Total hs trab + extras sem.de ref/ otras ocup.

				999=no responde
36	P15T	N	5 1	Total hs trab + horas extras sem.de ref 999=no responde
37	P16	N	1	Desearia trabajar más horas 1=si; 2=no
38	P16B	N	1	Buscó trabajar más hs. en la ocupa. Que tiene o en otra ocupación 1=si; 2=no
39	P17	N	1	Es usted... 1=patrón o empleador 2=trabajador por su cuenta 3=obrero o empleado 4=trabajador sin salario 9=no responde
40	P18	N	3	A que se dedica o que produce el establecimiento donde trabaja 999=no responde
41	P18B	N	1	Tipo de establecimiento 1=público; 2=privado; 3=otros 9=no responde
42	P19	N	1	Cuantas personas trab. en ese estab. 1= 1; 2=2 a 5; 3=6 a 15; 4=16 a 25 5= 26 a 50; 6=51 a 100; 7=101 a 500 8= 501 o más; 9=no sabe
43	P19B	N	1	Para pl9=9 se le consulta si sabe si en ese establecimiento trabajan 1=hasta 40 personas 2=más de 40 personas 9=no sabe
44	P20	N	3	Cual es el nombre de su ocupación y que tarea realiza en ella (ver Clasif. Nac. Ocupac.INDEC) 999=no responde
45	P21	N	10	Cuanto gana en esa ocupación 9=no responde
	P 21D	N		Cantidad de días por el que recibe paga. 99=no responde
46	P22	N	2	Cuanto tiempo hace que esta en esa ocupación (años)... 99=no responde
47	P22M	N	2	Cuanto tiempo hace que esta en esa ocupación (meses) ... 99=no responde
49	P23	N	2	En esa ocupación goza ud. De 32=indemnización por despido 08=vacaciones 04=aguinaldo 02=jubilación 16=seguro de trabajo 01=otras incluye obra social 63=todos los beneficios 64=sin beneficios 99=no responde
50	P24	N	1	Esa ocupación es 1=permanente 2=un trab. temporario(por plazo fijo, tarea u obra) 3=una changa 4=de duración desconocida(inestable) 9=no responde
51	P24_2_M	N	2	Para p24=2 por cuantos meses 99=no responde

52	P24_D	N	2	Para p24=2 y p24=3 por cuantos días 99=no responde
53	P29	N	1	Busca otra ocupaci6n 1=si; 2=no; 9=no responde
54	P30	N	1	Busca trabajo 1=porque gana poco 2=porque esta insatisfecho con su tarea 3=porque la relación con el empleador es mala 4=porque cree que lo van a despedir (asal.) 5=porque el trabajo que tiene se va a acabar 6=porque tiene poco trabajo (no asal.) 7=por otras causas laborales 8=por motivos personales
55	P31	N	1	Busca para 1=cambiar su ocupación principal 2=cambiar sus otras ocupaciones 3=tener otra ocupación además de la que tiene

BLOQUE DE DESOCUPADOS

56	P32	N	2	Cuánto tiempo hace que esta buscando empleo(meses)
57	P32D	N	2	Cuánto tiempo hace que esta buscando empleo (días)
58	233	N	1	Busca trabajar para 1=cubrir el presupuesto básico del hogar 2=complement.el presupuesto básico del hogar 3=aportar a otros gastos del hogar 4=solventar sus gastos personales
59	P34_1	N	1	No encuentra trabajo por edad 1=si 2=no
60	P34_2	N	1	No encuentra por nivel educativo requerido 1=si 2=no
61	P34_3	N	1	No encuentra por experien. laboral requerida 1=si 2=no
62	P34_4	N	1	No encuentra no hay trabaj. en su especial. 1=si 2=no
63	P34_5	N	1	No encuentra no hay trabaj. en general 1=si 2=no
64	P34_6	N	1	No encuentra por falta de vinculaciones 1=si 2=no
65	P34_7	N	1	No encuentra porq. los trabaj. que hay son mal pagos 1=si 2=no
66	P34_8	N	1	No encuentra se presenta en poco lugares por falta de dinero p/viajar 1=si 2=no
67	P34_9	N	1	No encuentra por otras razones 1=si 2=no
68	P37	N	1	Ha tenido ocupación anteriormente 1=si 2=no
69	P37B_A	N	2	Cuánto hace que la dejó (años)
70	P37B_M	N	2	Cuánto hace que la dejó (meses)
71	P37B_D	N	2	Cuánto hace que la dejó (días)
72	P38	N	1	En su ocupa. anterior era 1=patrón o empleador

				2=trabajador por su cuenta 3=obrero o empleado 4=trabajador sin salario
73	P38B-3	N	1	para p38=3 le hacían descuentos jubilatorios 1=SI 2=no
74	P38BIS		1	su última ocupación, era 1=permanente 2=un trabaj. temporario .... 3=una changa 4=de duración desconocida(inestable)
75	P38B-A	N	2	Cuanto tiempo trabajó,(años)
76	P38B-M	N	2	Cuanto tiempo trabajó (meses)
77	P38B-D	N	2	Cuanto tiempo trabajó (días)
78	P39	N	3	A que se dedica o produce el establecimiento
79	P39B	N	1	Tipo de establecimiento 1=público 2=privado 3=otros
80	P40	N	1	Cuantas personas trab.en ese estab.(ver apert.P19)
81	P40B	N	1	Para P40=9 ver apertura p19b de ocupados
82	P41	N	3	Cual era el nombre de su ocup. y que tarea realizaba en ella (Ver Clasif.Nac.Ocup.Indec)
83	P42	N	2	Causa fundamental por la que se quedó sin ocupaci6n 1=retiro voluntario sector público 2=le pagaban poco 3=tarea por debajo de su capacitación 4=lo despidieron (incluye cierre) 5=falta de trabajo (cuenta propia) 6=finalización trabaj. temporario 7=jubilación 8=otras causas laborales 9=otros motivos personales
84	P43	N	1	Le enviaron telegrams de despido 1=si 2=no
85	P44	N	1	Ese establece. cerró 1=si 2=no 9=no sabe
86	P45	N		En ese establecimiento 1=fue la única pers. que se quedó sin ocup. 2=o fueron despedidos otros trabajadores 9=no sabe

BLOQUE DE INGRESOS (Para toda la población encuestada)

87	P47	N	1	Recodificación de Ingresos de Fuente Laboral 1=tiene ingreso y declara monto 2=no tiene ingreso 9=tiene ingreso y no declara monto o declara monto parcial Podría indicarnos sus ingresos e/efectivo en el mes (mes anterior al relevamiento)
88	P47_1	N	10	... como asalariado -9=no responde
89	P47_2	N	10	... por bonificaci6n o gratificaciones no habituales (asalariados) -9=no responde
90	P47_3	N	10	... como trabajador cuenta propia -9=no responde
91	P47_4	N	10	...como ganancia de patrón (incluye sueldo asignado)

				-9=no responde
92	P47E_1	N	1	retira mercad. o prod. para consumo propio 1=si 2=no
93	P47E_2	N	1	recibe vales, tickets o similares, para comida o compra de mercadería 1=si 2=no
94	P48	N	1	Decodificaci6n de Ingresos relacionados con el trabajo 1=tiene ingreso y declara monto 2=no tiene ingreso 9=tiene ingreso y no declara monto o declara monto parcial
				Recibió Ingresos en el mes de .... por ... (mes anterior al relevamiento)
95	P48_1	N	10	o pensión
96	P48_2	N	10	alquileres, rentas o intereses
97	P48_3	N	10	utilid.,beneficios o dividendos
98	P48_4	N	10	seguro de desempleo
99	P48_5	N	10	indemnización por despido
100	P48_6	N	10	beca de estudio
101	P48_7	N	10	cuota de alimentos
102	P48_8	N	10	aportes de personas que no vivenen el hogar
103	P48_9	N	10	otros
104	P47T	N	10	MONTO INGRESOS TOTALES (sumatoria de montos de P47 y P48)
				Recodificación del P47 y P48
105	CODINGRE	N	1	1=tiene y declara monto 2=no tiene ingreso en ambas 9=tiene ingreso y no declara monto en alguna
106	P48E-3	N	1	Tuvo otros ingresos en especie (encomiendas fliares., copa de leche, medicamentos, etc.) 1=si 2=no

BLOQUE DE EDUCACION (Para toda la población)

107	P54	N	1	Sabe leer y escribir 1=si 2=no
108	P55	N	1	Asiste o asistió a la escuela 0=no responde 1=asiste 2=asistió 3=nunca asistió
109	P56	Ch	1	Que, estudio cursa o cursó indique solo el nivel más alto alcanzado blanco blanco = pre-escolar "00"=no responde "01"=Primario "02"=nacional "03"=comercial "04"=normal "05"=té6cnica "06"=otra enseñnza media "07"=superior "08"=universitaria
110	P57	Ch	60	Cual es la carrera o especial. que cursa o curso (variable sin codificar)

111	P58	N	1	Finalizo ese estudio 1=si 2=no 0=no responde
112	P58B	Ch	2	Cual es el último grado o año aprobado en ese estudio - preescolar se ingresa como blanco. - 8vo. del EGB se ingresa como 1ro. Nacional. - 9no. del EGB se ingresa como 2do. Nacional. - 99=Finalizó ese estudio. - 0=no responde.

BLOQUE DE MIGRACIONES (Para toda la población encuestada)

113	P59	N	1	Dónde nació 1=en esta ciudad 2=en otro lugar de esta Pcia. 3=en otra Pcia. 4=en otro país 9=No responde
114	P59COD	N	3	Código de provincia o país
115	P59_4	N	4	Año de llegada al país para p59=4
116	P60	N	1	Ha vivido fuera de esta ciudad área de relevamiento más de 6 meses 1=si 2=no
117	P61	N	1	Dónde (p6O=1) (anotar ultimo lugar) 2=en otro lugar de la Pcia. 3=en otra Pcia. 4=en otro país
118	P61COD	N	3	C6digo de Pcia. o país
119	P62	N	4	Desde cuando esta viviendo en forma continua en esta ciudad (años)(últimos 5 años) no sabe= 9999
120	P62M	N	2	Desde cuando esta viviendo en forma continua en esta ciudad (meses)(últimos 5 años)
121	P63_1	N	1	Trabaja en la Cdad. Bs. As. 1=si 2=no 9=no sabe
122	P63_2	N	1	Trabaja en Partidos 1=si 2=no 9=no sabe
123	P63_3	N	1	Trabaja en otro lugar... 1=si 2=no 9=no sabe
124	P63_P_1	N	1	ocup. principal en Cdad. Bs. As. 1=si 2=no 9=no sabe
125	P63_P_2	N	1	ocup. principal en Partidos 1=si 2=no 9=no sabe
126	P63_P_3	N	1	ocup. principal en otro lugar ... 2=no 9=no sabe
127	P63_S_1	N	1	ocup. secundaria en Cdad. Bs.As. 2=no 9=no sabe
128	P63_S_2	N	1	ocup. secundaria en Partidos 1=si 2=no 9=no sabe
129	P63_S_3	N	1	ocup. secundaria en otro lugar ... 1=si 2=no 9=no sabe
130	P64	N	1	trabaja fuera de esta ciudad área de relevamiento 1=si 2=no
				Este trabajo es ...
131	P65A	N	1	a)su ocupación principal 1=si 2=no

132	P65b	N	1	b) su ocupaci6n secundaria y otras 1=si 2=no
133	P66	N	1	Dónde trabaja 2=en otro lugar de esta provincia 3=en otra provincia 4=en otro país
134	P66cod	N	3	código de lugar donde trabaja pcia. o país
135	PONDERA	N	4	Ponderación
136	ITF	N	12 2	Monto del ingreso Total Familiar
137	DECIF	CH	2	Decil de ingreso Total Familiar
138	DECIF-DOM	CH	2	Decll de ingreso Total familiar por dominio (solo para GBA)
139	IPCF		N	12 2 monto del ingreso Per Capita Familiar
140	DECCF	Ch	2	Decil de ingreso Per Capita Familiar.
141	DECCF-DOM	Ch	2	Decil de ingreso Per Capita Familiar por dominio (solo para GBA)
142	DECIND	Ch	2	Decil de ingreso Individual
143	DECIND-DOM	Ch	2	Decil de ingreso Individual por dominio(solo para GBA)
144	DECOCU	Ch	2	Decil de ingreso de la ocupacion principal
145	DECOCU-DOM	Ch	2	Decil de ingreso de la ocupacion principal por dominio (solo para GBA)
146	INGHORA	N	10 2	Ingreso horario de la ocupaci6n principal
147	BENEF2	N	2	11=solo jubilación 12=combinaciones con jubilación 13=combinaciones sin jubilación 14=todos los beneficios 15=sin beneficios (*) 16=ocupados no asalariados(no corresp.benef) 17=no es ocupado (ESTADO=1) (*) Total sin jubilación códigos: (13+15)
148	RAMA	N	2	1=actividades primarias 2=Ind.alimentos, bebidas y tabaco 3=Ind.Textiles, confecciones y calzado 4=Ind.Prod.quimicos y de la refinación petróleo y combustible nuclear 5=Ind.Prod.metálicos, maquinarias y equipos 6=Otras industries manufactureras 7=Suministro de electricidad, gas y agua 8=Construcción 9=Comercio al por Mayor 10=Comercio al por Menor 11=Restaurantes y Hoteles 12=Transporte Servicios Conexos de Transporte y comunic. 14=Intermediación financiera 15=Actividades inmobiliarias, empresariales y de alquiler 16=Administración Pública y Defensa 17=Enseñanza 18=Servicios Sociales y de Salud 19=Otras Actividades de Servicios Comunitarios y sociales 20=Servicios de Reparación 21=Hogares privados con serv. doméstico. 22=Otros Servicios personales 89=Nuevos Trabajadores 99=Sin especificar
149	FUENTE	N	2	1=solo de trabajo asalariado 2=solo de trabajo por cuenta propia 3=solo de utilidades y beneficios 4=solo de alq. rentas, int. y dividendos

5=solo de jubilación o pensión
6=solo de otros ingresos
7=de trabajo asalariado y de trabajo por cta. propia.
8=de trabajo asalariado y de trabajo por cta.propia.
9=de trabajo asalariado y alquileres rentas, intereses y dividendos.
10=de trabajo asalariado y jubilac. o pensión
11=de trabajo asal y otros ingresos
12=de trabajo por cta. propia y utilidades y beneficios
13=de trabajo por cta propia y alq.,rentas, intereses y dividendos o pensión
15=de trabajo por cta propia y otros ingresos
16=de utilidades y beneficios y alquileres, rentas, intereses y dividendos
17=de utilidades y beneficios y jubilación o pensión
18=de utilidades y benef.y otros Ingresos
19=de alquileres, rentas, inter.y beneficios y jubilación o pensión
20=de alquil.rentas, int.y beneficios y otros Ingresos
21=de jubilación, pensión y otros ingresos
22=cualquier combinación de tres o más fuentes
23=no perciben ingresos

150	IMPUTA	CH	1	Campo usado únicamente en el aglomerado GBA que indica que este registro tiene imputado estado, categoría ocupacional. S=si
151	AREASNUE	CH	1	Variable que permite identificar en GBA el conjunto de nuevas áreas incorporadas a partir de Mayo de 1998(ver aclarac.novedades de la onda) S=si

Apéndice II

Datos del Hogar

Diccionario o Diseño de registro

	Campo	Tipo		Descripción
1	CODUSU	*N*	11	código de hogar para matching
2	ONDA	*N*	1	mes de relevamiento 1=abril/mayo; 3=octubre/noviembre; 4=agosto
3	ANO	*N*	2	al de relevamiento
4	AGLOMERADO	*N*	2	número de Area de relevamiento
5	REALIZADA	*N*	1	si se realizó la entrevista al hogar 0=no; 1=si
6	RAZONUSU	*N*	1	causa de no respuesta de vivienda (preg.22 del cuestionario) 1=Vivienda encuestable 2=Vivienda NO encuestable
7	P01	*N*	1	tipo de vivienda 1=Casa 2=Departamento 3=Vivienda en el lugar de trabajo 4=Inquilinato 5=Hotel o pensión 6=Vivienda no destinada a fines habitacionales 7=Vivienda en villa 8=Otro
8	P02	*N*	2	cantidad de habitaciones que tiene la vivienda (excluido baño y cocina)
9	P03	*N*	2	habitaciones de uso exclusivo del hogar 99=No responde
10	P04	*N*	1	instalación de agua 1=si; 2=no
11	P05	*N*	1	instalación de electricidad 1=si; 2=no
12	P06a	*N*	1	instalación de baño 1= tiene 2= no tiene
13	P06b	*N*	1	el baño tiene 1=inodoro con botón o cadena con arrastre de agua 2=inodoro sin botón o cadena con arrastre de agua 3=letrina
14	P06c	*N*	1	el desagüe es 1:a red pública o cloaca 2:a cámara séptica u pozo ciego 3:solo a pozo ciego

15	P06d	*N*	1	el baño es de 1=de uso exclusive del hogar 2=compartido con otro hogar
16	P07	*N*	1	régimen de tenencia de la vivienda 1=propietario de la vivienda y el terreno 2=propietario de la vivienda solamente 3=inquilino o arrendatario de la vivienda 4=ocupante con relación de dependencia 5=ocupante gratuito 8=otros
17	P08	*N*	1	tipo de materiales de la vivienda (predominantes de paredes externas) 1=mamposteria(ladrillo, bloques, paneles, etc.) 2=madera 3=metal o fibrocem.(chapas lisas o fibroc) 4=adobe 5=chorizo, cartón o desechos 8=otros

CUADRO RESUMEN

15	R01	*N*	2	cant. de personas en el hogar
19	R02	*N*	2	cant. de personas encuestadas
20	R03	*N*	2	cant. de personas no encuestadas
21	R04	*N*	2	cant. de personas ocupadas
22	R05	*N*	2	cant. de personas desocupadas
23	R06	*N*	2	cant. de personas inactivas
24	PONDERA	*N*	4	ponderación
25	ITF	*N*	12 2	monto de ingreso total familiar
26	DECIF	Ch	2	No. de decil de ingreso total familiar
27	DECIFDOM	Ch	2	No. de decil de ingreso total familiar por dominio (solo para GBA)
28	IPCF	*N*	12 2	monto de ingreso per cápita familiar
29	DECCF	Ch	2	No. de decil de ingreso per cápita familiar
30	DECCFDOM	Ch	2	No. de decil de ingreso per cápita familiar por dominio (solo para GBA)
31	MEN14	*N*	2	No. de menores de 14 años en el hogar
32	CAT60	*N*	2	No. de personas de 60 y más en el hogar
34	PERCEPT	*N*	2	No de perceptores de ingreso en el hogar (se consideran para el cálculo los hogares en los cuales se conoce el monto de ingreso de todos sus miembros)
35	HOGAR1	Ch	3	variable construida que se usa combinando tres dígitos e identifica características del hogar. PRIMER DIGITO (Condición de actividad del jefe. 1=jefe ocupado 2=jefe desocupado 3=jefe inactivo 0=jefe con entrevista no realizada SEGUNDO DIGITO -(Tipo de hogar) 1=unipersonal 2=jefe y cónyuge solamente 3=jefe, cónyuge e hijos 4=jefe e hijos 5=cualquier alternativa siguiente con hijos políticos y/o nietos (1-2-3-4) 6=cualquier alternativa siguiente con padres 0 suegros (1-2-3-4)

				7=cualquier alternativa siguiente con otros familiares (1-2-3-4) 8=cualquier alternativa siguiente con otros componentes.(1-2-3-4) 9=otras combinaciones. TERCER DIGITO (Presencia de servicio doméstico en el hogar) 1=hogar con servicio doméstico 2=hogar sin servicio doméstico
36	AREASNUE	CH	1	Variable que permite identificar en GBA el conjunto de nuevas áreas incorporadas a partir de Mayo 1998 (ver aclarac.novedades de la onda). S=si

Apéndice III

Resultados See5

Todas las hojas del árbol indican la clase o aglomerado *'33'/'32'*, seguida por *(n)* o *(n/m)*. El valor de *n* es el número de casos en el archivo *eph.data* que son mapeados a esa hoja, y *m* es el número de ellos que son clasificados incorrectamente por la hoja (un número no entero de casos puede ser porque el valor de algún atributo en el árbol es desconocido, el *See5* separa el caso y divide una fracción bajo cada rama).

Resultados del 1er. Grupo

```
See5 [Release 1.14]   Sat Apr 14 12:03:29 2001
-------------------

    Options:
       Pruning confidence level 30%
       Test requires two branches with >= 15 items

Read 1595 cases (37 attributes) from SEE5_G1.data

Decision tree:

HP06A-HP06B-HP06C-HP06D in {1_2_2_2,1_1_9_1,1_2_1_2}: 33 (0)
HP06A-HP06B-HP06C-HP06D = 1_1_2_2: 33 (2)
HP06A-HP06B-HP06C-HP06D = 1_1_1_2: 32 (22/10)
HP06A-HP06B-HP06C-HP06D = 1_1_2_1: 33 (437)
HP06A-HP06B-HP06C-HP06D = 1_2_2_1: 33 (72)
HP06A-HP06B-HP06C-HP06D = 1_1_3_1: 33 (106)
HP06A-HP06B-HP06C-HP06D = 1_3_0_0: 33 (70/3)
HP06A-HP06B-HP06C-HP06D = 1_2_1_1: 33 (10)
HP06A-HP06B-HP06C-HP06D = 1_2_3_1: 33 (100/5)
HP06A-HP06B-HP06C-HP06D = 1_2_3_2: 33 (8/2)
HP06A-HP06B-HP06C-HP06D = 2_0_0_0: 33 (65)
HP06A-HP06B-HP06C-HP06D = 1_1_3_2: 33 (4)
HP06A-HP06B-HP06C-HP06D = 1_1_1_1:
:...H_P01-H_P02 in {2_8,4_8,2_7,2_9,4_1,4_2,5_1,1_8,7_1,1_7,6_1,1_9,7_3,3_1,
    :               7_2,3_2,4_3,2_8}: 33 (0)
    H_P01-H_P02 = 1_3: 33 (149/47)
    H_P01-H_P02 = 2_2: 32 (101/24)
    H_P01-H_P02 = 1_4: 33 (91/20)
    H_P01-H_P02 = 2_5: 32 (27/6)
    H_P01-H_P02 = 1_6: 33 (16/5)
    H_P01-H_P02 = 1_2: 33 (43/9)
    H_P01-H_P02 = 2_1: 32 (11/3)
    H_P01-H_P02 = 1_1: 33 (5)
    H_P01-H_P02 = 2_6: 33 (9/3)
    H_P01-H_P02 = 4_4: 32 (3)
    H_P01-H_P02 = 2_3:
    :...H_ITFCAT = 0: 32 (10/1)
    :   H_ITFCAT = 0_200: 32 (1)
    :   H_ITFCAT = 200_500: 32 (15/7)
    :   H_ITFCAT = 500_800: 33 (27/1)
    :   H_ITFCAT = 800_1000: 32 (7)
    :   H_ITFCAT = 1000_1500: 32 (16/6)
    :   H_ITFCAT = 1500_2000: 32 (8)
    :   H_ITFCAT = 2000_4000: 32 (47/10)
    :   H_ITFCAT = 4000_17000: 32 (4)
    H_P01-H_P02 = 2_4:
    :...H_R01CAT in {7_8_,9_10_,11_12_13_14_,7-8_,14_}: 32 (0)
    :   H_R01CAT = 2_: 33 (10/4)
    :   H_R01CAT = 1_: 32 (4/2)
```

```
:   H_R01CAT = 5_: 32 (5)
:   H_R01CAT = 4_: 32 (20)
:   H_R01CAT = 3_: 32 (15)
:   H_R01CAT = 6_: 33 (6)
H_P01-H_P02 = 1_5:
:...H_R01CAT in {2_,4_,7_8_,9_10_,11_12_13_14_,7-8_,14_}: 33 (0)
    H_R01CAT = 1_: 33 (2)
    H_R01CAT = 5_: 32 (20)
    H_R01CAT = 3_: 33 (9/3)
    H_R01CAT = 6_: 33 (18)


Evaluation on training data (1595 cases):

      Decision Tree
    ----------------
    Size      Errors

      40  171(10.7%)   <<


     (a)   (b)    <-classified as
    ----  ----
     267   102    (a): class 32
      69  1157    (b): class 33


Evaluation on test data (802 cases):

      Decision Tree
    ----------------
    Size      Errors

      40  128(16.0%)   <<


     (a)   (b)    <-classified as
    ----  ----
     104    87    (a): class 32
      41   570    (b): class 33


Time: 0.6 secs
```

Resultados del 2do. Grupo

```
See5 [Release 1.14]   Sat Apr 14 12:03:13 2001
-------------------

    Options:
       Pruning confidence level 30%
       Test requires two branches with >= 15 items

Read 1569 cases (37 attributes) from SEE5_G2.data

Decision tree:

HP06A-HP06B-HP06C-HP06D in {1_2_2_2,1_2_1_2}: 33 (0)
HP06A-HP06B-HP06C-HP06D = 1_1_2_2: 33 (6)
HP06A-HP06B-HP06C-HP06D = 1_1_1_2: 32 (17)
HP06A-HP06B-HP06C-HP06D = 1_1_2_1: 33 (365)
HP06A-HP06B-HP06C-HP06D = 1_2_2_1: 33 (84)
HP06A-HP06B-HP06C-HP06D = 1_1_3_1: 33 (160/3)
HP06A-HP06B-HP06C-HP06D = 1_3_0_0: 33 (38)
HP06A-HP06B-HP06C-HP06D = 1_2_1_1: 33 (12/5)
HP06A-HP06B-HP06C-HP06D = 1_2_3_1: 33 (100/4)
HP06A-HP06B-HP06C-HP06D = 1_2_3_2: 33 (5/2)
HP06A-HP06B-HP06C-HP06D = 2_0_0_0: 33 (66/1)
HP06A-HP06B-HP06C-HP06D = 1_1_3_2: 33 (2)
HP06A-HP06B-HP06C-HP06D = 1_1_9_1: 33 (3)
HP06A-HP06B-HP06C-HP06D = 1_1_1_1:
:...H_P01-H_P02 in {2_8,4_8,2_9,4_1,4_2,5_1,1_8,7_1,6_1,1_9,7_3,3_1,7_2,4_3,
    :               2_8,4_4}: 33 (0)
    H_P01-H_P02 = 1_3: 33 (145/27)
```

```
    H_P01-H_P02 = 1_4: 33 (121/28)
    H_P01-H_P02 = 1_5: 32 (22/8)
    H_P01-H_P02 = 2_5: 32 (9)
    H_P01-H_P02 = 1_6: 33 (19/3)
    H_P01-H_P02 = 1_2: 33 (42/11)
    H_P01-H_P02 = 2_1: 32 (21/9)
    H_P01-H_P02 = 2_7: 32 (4)
    H_P01-H_P02 = 1_1: 33 (4)
    H_P01-H_P02 = 2_6: 32 (3)
    H_P01-H_P02 = 1_7: 32 (5/2)
    H_P01-H_P02 = 3_2: 32 (1)
    H_P01-H_P02 = 2_4:
    :...H_MEN14CAT in {3_,4_,5_,7_8_9_}: 32 (0)
    :   H_MEN14CAT = 0_: 32 (42/2)
    :   H_MEN14CAT = 1_: 32 (9/3)
    :   H_MEN14CAT = 2_: 32 (16/5)
    :   H_MEN14CAT = 6_: 33 (11)
    H_P01-H_P02 = 2_3:
    :...H_R01CAT in {9_10_,11_12_13_14_,7-8_,14_}: 32 (0)
    :   H_R01CAT = 2_: 32 (40/10)
    :   H_R01CAT = 1_: 32 (10/1)
    :   H_R01CAT = 5_: 33 (10)
    :   H_R01CAT = 3_: 32 (15/6)
    :   H_R01CAT = 7_8_: 32 (8)
    :   H_R01CAT = 6_: 32 (12)
    :   H_R01CAT = 4_:
    :   :...H_MEN14CAT in {3_,4_,5_,6_,7_8_9_}: 32 (0)
    :       H_MEN14CAT = 0_: 33 (16/4)
    :       H_MEN14CAT = 1_: 32 (16/4)
    :       H_MEN14CAT = 2_: 32 (8)
    H_P01-H_P02 = 2_2:
    :...H_MEN14CAT in {3_,4_,6_,7_8_9_}: 32 (0)
        H_MEN14CAT = 1_: 32 (29/5)
        H_MEN14CAT = 2_: 33 (4)
        H_MEN14CAT = 5_: 33 (6)
        H_MEN14CAT = 0_:
        :...H_R01CAT in {5_,7_8_,6_,9_10_,11_12_13_14_,7-8_,14_}: 32 (0)
            H_R01CAT = 2_: 32 (34/10)
            H_R01CAT = 1_: 32 (19/5)
            H_R01CAT = 4_: 32 (4)
            H_R01CAT = 3_: 33 (6)

Evaluation on training data (1569 cases):

          Decision Tree
        ----------------
        Size      Errors

          44  158(10.1%)   <<


         (a)   (b)    <-classified as
        ----  ----
         274    88    (a): class 32
          70  1137    (b): class 33


Evaluation on test data (810 cases):

          Decision Tree
        ----------------
        Size      Errors

          44  106(13.1%)   <<


         (a)   (b)    <-classified as
        ----  ----
         128    44    (a): class 32
          62   576    (b): class 33


Time: 0.5 secs
```

Resultados del 3er. Grupo

```
See5 [Release 1.14]   Sat Apr 14 12:02:57 2001
-------------------

    Options:
       Pruning confidence level 30%
       Test requires two branches with >= 15 items

Read 1597 cases (37 attributes) from SEE5_G3.data

Decision tree:

HP06A-HP06B-HP06C-HP06D in {1_1_9_1,1_2_1_2}: 33 (0)
HP06A-HP06B-HP06C-HP06D = 1_1_2_2: 33 (12)
HP06A-HP06B-HP06C-HP06D = 1_1_1_2: 32 (11)
HP06A-HP06B-HP06C-HP06D = 1_1_2_1: 33 (319)
HP06A-HP06B-HP06C-HP06D = 1_2_2_1: 33 (87)
HP06A-HP06B-HP06C-HP06D = 1_1_3_1: 33 (143)
HP06A-HP06B-HP06C-HP06D = 1_3_0_0: 33 (13)
HP06A-HP06B-HP06C-HP06D = 1_2_1_1: 33 (5)
HP06A-HP06B-HP06C-HP06D = 1_2_3_1: 33 (130/7)
HP06A-HP06B-HP06C-HP06D = 1_2_3_2: 33 (2)
HP06A-HP06B-HP06C-HP06D = 2_0_0_0: 33 (64)
HP06A-HP06B-HP06C-HP06D = 1_1_3_2: 33 (10)
HP06A-HP06B-HP06C-HP06D = 1_2_2_2: 33 (5)
HP06A-HP06B-HP06C-HP06D = 1_1_1_1:
:...H_P01-H_P02 in {4_8,2_9,4_1,4_2,5_1,7_1,6_1,1_9,7_3,3_1,7_2,3_2,4_3,
    :               2_8}: 33 (0)
    H_P01-H_P02 = 1_3: 33 (156/24)
    H_P01-H_P02 = 2_8: 32 (10/4)
    H_P01-H_P02 = 1_4: 33 (101/22)
    H_P01-H_P02 = 1_5: 33 (34/13)
    H_P01-H_P02 = 2_5: 32 (46/11)
    H_P01-H_P02 = 1_6: 33 (20)
    H_P01-H_P02 = 1_2: 33 (41/2)
    H_P01-H_P02 = 2_1: 32 (5/2)
    H_P01-H_P02 = 2_7: 32 (5)
    H_P01-H_P02 = 1_1: 33 (9)
    H_P01-H_P02 = 2_6: 32 (3)
    H_P01-H_P02 = 1_8: 33 (6)
    H_P01-H_P02 = 1_7: 32 (11)
    H_P01-H_P02 = 2_2:
    :...H_R01CAT in {7_8_,6_,9_10_,11_12_13_14_,7-8_,14_}: 32 (0)
    :   H_R01CAT = 2_: 32 (40/6)
    :   H_R01CAT = 1_: 32 (14/2)
    :   H_R01CAT = 5_: 33 (5)
    :   H_R01CAT = 4_: 33 (20/8)
    :   H_R01CAT = 3_: 32 (9/3)
    H_P01-H_P02 = 2_4:
    :...H_IPCFCAT in {100_300,4000_9000}: 32 (0)
    :   H_IPCFCAT = 0: 32 (7/2)
    :   H_IPCFCAT = 0_100: 33 (9)
    :   H_IPCFCAT = 300_600: 32 (13)
    :   H_IPCFCAT = 600_1000: 32 (32/3)
    :   H_IPCFCAT = 1000_2000: 32 (19/4)
    :   H_IPCFCAT = 2000_4000: 32 (2)
    H_P01-H_P02 = 2_3:
    :...H_MEN14CAT in {5_,6_,7_8_9_}: 32 (0)
        H_MEN14CAT = 2_: 32 (20/4)
        H_MEN14CAT = 3_: 33 (12)
        H_MEN14CAT = 4_: 33 (6)
        H_MEN14CAT = 0_:
        :...H_R01CAT in {7_8_,6_,9_10_,11_12_13_14_,7-8_,14_}: 32 (0)
        :   H_R01CAT = 2_: 32 (32/8)
        :   H_R01CAT = 1_: 32 (8/2)
        :   H_R01CAT = 5_: 32 (5)
        :   H_R01CAT = 4_: 32 (12)
        :   H_R01CAT = 3_: 33 (24/9)
        H_MEN14CAT = 1_:
        :...H14 = 9: 32 (0)
            H14 = 1: 32 (22/7)
            H14 = 2: 33 (6)
            H14 = 3: 32 (29/7)
            H14 = 4: 32 (1)
            H14 = 5: 33 (2)
```

```
Evaluation on training data (1597 cases):

          Decision Tree
        ----------------
        Size      Errors

          49  150( 9.4%)   <<


         (a)   (b)    <-classified as
        ----  ----
         291    85    (a): class 32
          65  1156    (b): class 33


Evaluation on test data (801 cases):

          Decision Tree
        ----------------
        Size      Errors

          49  113(14.1%)   <<


         (a)   (b)    <-classified as
        ----  ----
         110    75    (a): class 32
          38   578    (b): class 33


Time: 0.6 secs
```

Resultados del 4to. Grupo

```
See5 [Release 1.14]   Sat Apr 14 12:02:40 2001
-------------------

    Options:
       Pruning confidence level 30%
       Test requires two branches with >= 15 items

Read 1591 cases (37 attributes) from SEE5_G4.data

Decision tree:

HP06A-HP06B-HP06C-HP06D in {1_1_9_1,1_2_1_2}: 33 (0)
HP06A-HP06B-HP06C-HP06D = 1_1_2_2: 33 (13)
HP06A-HP06B-HP06C-HP06D = 1_1_1_2: 32 (7/2)
HP06A-HP06B-HP06C-HP06D = 1_1_2_1: 33 (424)
HP06A-HP06B-HP06C-HP06D = 1_2_2_1: 33 (66)
HP06A-HP06B-HP06C-HP06D = 1_1_3_1: 33 (179/3)
HP06A-HP06B-HP06C-HP06D = 1_3_0_0: 33 (50)
HP06A-HP06B-HP06C-HP06D = 1_2_1_1: 33 (4)
HP06A-HP06B-HP06C-HP06D = 1_2_3_1: 33 (85)
HP06A-HP06B-HP06C-HP06D = 1_2_3_2: 33 (7)
HP06A-HP06B-HP06C-HP06D = 2_0_0_0: 33 (81)
HP06A-HP06B-HP06C-HP06D = 1_1_3_2: 33 (6)
HP06A-HP06B-HP06C-HP06D = 1_2_2_2: 33 (17)
HP06A-HP06B-HP06C-HP06D = 1_1_1_1:
:...H_P01-H_P02 in {2_8,4_8,2_9,4_1,4_2,5_1,7_1,6_1,1_9,7_2,3_2,
    :                2_8}: 32 (0)
    H_P01-H_P02 = 2_3: 32 (169/28)
    H_P01-H_P02 = 1_3: 33 (118/12)
    H_P01-H_P02 = 2_4: 32 (51/9)
    H_P01-H_P02 = 1_4: 33 (68/9)
    H_P01-H_P02 = 1_5: 33 (57/18)
    H_P01-H_P02 = 2_5: 32 (11/2)
    H_P01-H_P02 = 1_6: 33 (15)
    H_P01-H_P02 = 1_2: 33 (29/4)
    H_P01-H_P02 = 2_1: 32 (15/2)
    H_P01-H_P02 = 2_7: 32 (3)
    H_P01-H_P02 = 1_1: 33 (6/2)
```

```
    H_P01-H_P02 = 2_6: 33 (6)
    H_P01-H_P02 = 1_8: 33 (8/2)
    H_P01-H_P02 = 1_7: 33 (1)
    H_P01-H_P02 = 7_3: 33 (2)
    H_P01-H_P02 = 3_1: 32 (2)
    H_P01-H_P02 = 4_3: 32 (3)
    H_P01-H_P02 = 2_2:
    :...H_ITFCAT = 4000_17000: 32 (0)
        H_ITFCAT = 0: 32 (5)
        H_ITFCAT = 0_200: 32 (1)
        H_ITFCAT = 200_500: 32 (16/2)
        H_ITFCAT = 500_800: 33 (16/3)
        H_ITFCAT = 800_1000: 32 (13/2)
        H_ITFCAT = 1000_1500: 32 (17)
        H_ITFCAT = 1500_2000: 32 (4/2)
        H_ITFCAT = 2000_4000: 32 (16)


Evaluation on training data (1591 cases):

          Decision Tree
        ----------------
        Size      Errors

          37  102( 6.4%)   <<


         (a)   (b)    <-classified as
        ----  ----
         284    53    (a): class 32
          49  1205    (b): class 33


Evaluation on test data (800 cases):

          Decision Tree
        ----------------
        Size      Errors

          37  117(14.6%)   <<


         (a)   (b)    <-classified as
        ----  ----
         109    65    (a): class 32
          52   574    (b): class 33


Time: 0.6 secs
```

Resultados del 5to. Grupo

```
See5 [Release 1.14]   Sat Apr 14 12:02:25 2001
-------------------

    Options:
       Pruning confidence level 30%
       Test requires two branches with >= 15 items

Read 1538 cases (37 attributes) from SEE5_G5.data

Decision tree:

HP06A-HP06B-HP06C-HP06D in {1_2_2_2,1_1_9_1}: 33 (0)
HP06A-HP06B-HP06C-HP06D = 1_1_2_2: 33 (6)
HP06A-HP06B-HP06C-HP06D = 1_1_1_2: 32 (12/2)
HP06A-HP06B-HP06C-HP06D = 1_2_2_1: 33 (82/6)
HP06A-HP06B-HP06C-HP06D = 1_1_3_1: 33 (104)
HP06A-HP06B-HP06C-HP06D = 1_3_0_0: 33 (70)
HP06A-HP06B-HP06C-HP06D = 1_2_1_1: 33 (15)
HP06A-HP06B-HP06C-HP06D = 1_2_3_1: 33 (90)
HP06A-HP06B-HP06C-HP06D = 1_2_3_2: 33 (14)
HP06A-HP06B-HP06C-HP06D = 2_0_0_0: 33 (60/3)
HP06A-HP06B-HP06C-HP06D = 1_1_3_2: 33 (3)
```

```
HP06A-HP06B-HP06C-HP06D = 1_2_1_2: 33 (5)
HP06A-HP06B-HP06C-HP06D = 1_1_2_1:
:...H_P07-H_P08 in {2_2,5_2,1_3,2_3,1_2,5_3,5_8,3_3}: 33 (0)
:   H_P07-H_P08 = 1_1: 33 (256)
:   H_P07-H_P08 = 5_1: 33 (20)
:   H_P07-H_P08 = 4_1: 33 (1)
:   H_P07-H_P08 = 3_1: 33 (26)
:   H_P07-H_P08 = 2_1: 33 (8)
:   H_P07-H_P08 = 8_1: 33 (2)
:   H_P07-H_P08 = 3_2: 32 (6)
HP06A-HP06B-HP06C-HP06D = 1_1_1_1:
:...H_P01-H_P02 in {2_8,4_8,2_7,2_9,4_1,5_1,1_8,7_1,6_1,1_9,7_3,3_1,7_2,3_2,
    :              2_8}: 33 (0)
    H_P01-H_P02 = 1_3: 33 (149/26)
    H_P01-H_P02 = 2_5: 32 (34/7)
    H_P01-H_P02 = 2_1: 32 (23/9)
    H_P01-H_P02 = 1_1: 33 (5)
    H_P01-H_P02 = 4_2: 32 (4)
    H_P01-H_P02 = 2_6: 32 (10)
    H_P01-H_P02 = 1_7: 32 (9/4)
    H_P01-H_P02 = 4_3: 32 (4)
    H_P01-H_P02 = 2_4:
    :...H_MEN14CAT in {4_,5_,6_,7_8_9_}: 32 (0)
    :   H_MEN14CAT = 0_: 32 (26)
    :   H_MEN14CAT = 1_: 32 (18)
    :   H_MEN14CAT = 2_: 33 (13/4)
    :   H_MEN14CAT = 3_: 32 (7)
    H_P01-H_P02 = 1_4:
    :...H_IPCFCAT in {0_100,1000_2000,4000_9000}: 33 (0)
    :   H_IPCFCAT = 0: 32 (10)
    :   H_IPCFCAT = 100_300: 33 (37/12)
    :   H_IPCFCAT = 300_600: 33 (28/3)
    :   H_IPCFCAT = 600_1000: 32 (3/1)
    :   H_IPCFCAT = 2000_4000: 33 (4)
    H_P01-H_P02 = 1_5:
    :...H_R01CAT in {1_,7_8_,9_10_,11_12_13_14_,7-8_,14_}: 32 (0)
    :   H_R01CAT = 2_: 32 (8/4)
    :   H_R01CAT = 5_: 32 (20/10)
    :   H_R01CAT = 4_: 32 (16/4)
    :   H_R01CAT = 3_: 32 (6)
    :   H_R01CAT = 6_: 33 (6)
    H_P01-H_P02 = 1_6:
    :...CODINGRE = 9: 32 (0)
    :   CODINGRE = 1: 33 (15/6)
    :   CODINGRE = 2: 32 (15/5)
    H_P01-H_P02 = 1_2:
    :...H_CAT60 = 1_: 32 (9/3)
    :   H_CAT60 = 2_: 33 (15/4)
    :   H_CAT60 = 3_: 33 (37)
    H_P01-H_P02 = 2_3:
    :...H_MEN14CAT in {5_,6_,7_8_9_}: 32 (0)
    :   H_MEN14CAT = 2_: 32 (12)   ,
    :   H_MEN14CAT = 3_: 32 (5)
    :   H_MEN14CAT = 4_: 33 (6)
    :   H_MEN14CAT = 0_:
    :   :...P24 in {3,9}: 32 (0)
    :   :   P24 = 0: 33 (36/16)
    :   :   P24 = 1: 32 (30/9)
    :   :   P24 = 2: 33 (3)
    :   :   P24 = 4: 33 (1)
    :   H_MEN14CAT = 1_:
    :   :...H_P07-H_P08 in {5_1,2_1,2_2,8_1,5_2,1_3,2_3,1_2,3_2,5_3,5_8,
    :       :              3_3}: 32 (0)
    :       H_P07-H_P08 = 1_1: 33 (23/10)
    :       H_P07-H_P08 = 4_1: 33 (3)
    :       H_P07-H_P08 = 3_1: 32 (16/5)
    H_P01-H_P02 = 2_2:
    :...H_IPCFCAT in {0_100,2000_4000,4000_9000}: 32 (0)
        H_IPCFCAT = 0: 32 (7)
        H_IPCFCAT = 300_600: 32 (25/1)
        H_IPCFCAT = 600_1000: 32 (11)
        H_IPCFCAT = 1000_2000: 32 (3)
        H_IPCFCAT = 100_300:
        :...H_MEN14CAT in {3_,4_,6_,7_8_9_}: 33 (0)
            H_MEN14CAT = 0_: 32 (21/9)
            H_MEN14CAT = 1_: 33 (15/6)
            H_MEN14CAT = 2_: 32 (3)
```

```
            H_MEN14CAT = 5_: 33 (7)

Evaluation on training data (1538 cases):

          Decision Tree
        ----------------
        Size      Errors

          63  169(11.0%)   <<


         (a)   (b)    <-classified as
        ----  ----
         300    96    (a): class 32
          73  1069    (b): class 33


Evaluation on test data (808 cases):

          Decision Tree
        ----------------
        Size      Errors

          63  141(17.5%)   <<


         (a)   (b)    <-classified as
        ----  ----
         102    86    (a): class 32
          55   565    (b): class 33


Time: 0.6 secs
```

Apéndice IV

Resultados SODAS

Resultados del 1er. Grupo

```
-------------------------------------------------------
 BASE= C:\MIMI\TESIS\SODAS_G1.SDS
 Number of OS = 716
 Number of variables = 41
 METHOD=SODAS_TREE Version 1.2     INRIA 1998
-------------------------------------------------------

 Learning Set          :   482
 Number of variables :    30
 Max. number of nodes:    59
 Soft Assign           : ( 0 ) PURE
 Criterion coding      : ( 1 ) GINI
 Min. number of object by node    :       15
 Min. size of no-majority classes :       15
 Min. size of descendant nodes    :    15.00
 Frequency of test set            :    33.00

 GROUP OF PREDICATE VARIABLES :
 (   1 ) HP01                                    6 MODALITIES
 (   2 ) HP02                                    9 MODALITIES
 (   3 ) HP06A                                   2 MODALITIES
 (   4 ) HP06B                                   4 MODALITIES
 (   5 ) HP06C                                   4 MODALITIES
 (   6 ) HP06D                                   3 MODALITIES
 (   7 ) H_P07                                   6 MODALITIES
 (   8 ) H_P08                                   3 MODALITIES
 (   9 ) H_R01CAT                                9 MODALITIES
 (  11 ) H_IPCFCAT                               8 MODALITIES
 (  12 ) H_CAT60                                 3 MODALITIES
 (  14 ) H08                                    11 MODALITIES
 (  15 ) H12CAT                                  8 MODALITIES
 (  16 ) H13                                     2 MODALITIES
 (  17 ) H14                                     5 MODALITIES
 (  18 ) P11                                     8 MODALITIES
 (  19 ) ESTADO                                  3 MODALITIES
 (  22 ) P18B                                    5 MODALITIES
 (  24 ) 20A                                    11 MODALITIES
 (  25 ) 20B                                    11 MODALITIES
 (  26 ) 20C                                     6 MODALITIES
 (  27 ) P22MCAT                                10 MODALITIES
 (  29 ) P24                                     6 MODALITIES
 (  30 ) P32DCAT                                 8 MODALITIES
 (  32 ) CODINGRE                                3 MODALITIES
 (  34 ) P54                                     2 MODALITIES
 (  35 ) P55                                     3 MODALITIES
 (  36 ) P56                                     9 MODALITIES
 (  37 ) P58                                     3 MODALITIES
 (  39 ) P59                                     5 MODALITIES

 CLASSIFICATION VARIABLE :
 (  41 ) H_AGLOMERADO

 NUMBER OF A PRIORI CLASSES :   2
      ID_CLASS      NAME_CLASS
             1              32
             2              33


   CLASS  SIZE   LEARNING     TEST

      1    212      142         70
      2    504      340        164

 TOTAL    716      482        234

 ================================
 | SPLIT OF A NODE        :     1 |
 ================================

 LEARNING SET

    ====================================================
    |      | N(k/t)  |   N(k)    |  P(k/t)  |  P(t/k)  |
    ====================================================
    | 32   |   142.00 |   142.00 |    29.46 |   100.00 |
    | 33   |   340.00 |   340.00 |    70.54 |   100.00 |
    ====================================================


 TEST SET

    ====================================================
    |      | N(k/t)  |   N(k)    |  P(k/t)  |  P(t/k)  |
    ====================================================
    | 32   |    70.00 |    70.00 |    29.91 |   100.00 |
    | 33   |   164.00 |   164.00 |    70.09 |   100.00 |
    ====================================================

TREE CRITERION  0.415626

    ==========================================================
```

Ord	variable	value	criterion
1	(5) HP06C	1000	0.2670
2	(1) HP01	011000	0.3033
3	(11) H_IPCPCAT	01000100	0.3614
4	(36) P56	000010110	0.3758
5	(26) 20C	011000	0.3840

SPLITTING NODE: 1

VARIABLE : (5) HP06C
SPLIT : 1000 (1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
(1) 1
MODALITIES BELONG RIGHT NODE :
(2) 3
(3) 0
(4) 2
CRITERION : 0.266952

LEARNING SET

nod	left node 2	right node 3	Row totals 1
32	141.00	1.00	142.00
33	115.00	225.00	340.00
Tot	256.00	226.00	482.00

TEST SET

nod	left node 2	right node 3	Row totals 1
32	68.00	2.00	70.00
33	65.00	99.00	164.00
Tot	133.00	101.00	234.00

SPLIT OF A NODE : 2

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	141.00	142.00	55.08	99.30
33	115.00	340.00	44.92	33.82

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	68.00	70.00	51.13	97.14
33	65.00	164.00	48.87	39.63

TREE CRITERION 0.494843

Ord	variable	value	criterion
1	(1) HP01	010000	0.3931
2	(11) H_IPCPCAT	01000100	0.4603
3	(36) P56	000011110	0.4712
4	(2) HP02	101010000	0.4799
5	(26) 20C	100110	0.4805

SPLITTING NODE: 2

VARIABLE : (1) HP01
SPLIT : 010000 (1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
(2) 1
MODALITIES BELONG RIGHT NODE :
(1) 2
(3) 7
(4) 5
(5) 4
(6) 3
CRITERION : 0.393080

LEARNING SET

nod	left node 4	right node 5	Row totals 2
32	32.00	109.00	141.00
33	78.00	37.00	115.00
Tot	110.00	146.00	256.00

TEST SET

nod	left node 4	right node 5	Row totals 2
32	19.00	49.00	68.00
33	44.00	21.00	65.00
Tot	63.00	70.00	133.00

```
 ================================
 | SPLIT OF A NODE       :     3 |
 ================================

 LEARNING SET

    ================================================
    |      | N(k/t)  |   N(k)   | P(k/t)  | P(t/k)  |
    ================================================
    | 32   |    1.00 |   142.00 |    0.44 |    0.70 |
    | 33   |  225.00 |   340.00 |   99.56 |   66.18 |
    ================================================

THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
     SIZE OF THE NO-MAJORITY CLASSES 1.000000  VALUE OF STOP-SPLITTING RULE 15.000000

 TEST SET

    ================================================
    |      | N(k/t)  |   N(k)   | P(k/t)  | P(t/k)  |
    ================================================
    | 32   |    2.00 |    70.00 |    1.98 |    2.86 |
    | 33   |   99.00 |   164.00 |   98.02 |   60.37 |
    ================================================


 THIS NODE IS A TERMINAL NODE


 ================================
 | SPLIT OF A NODE       :     4 |
 ================================

 LEARNING SET

    ================================================
    |      | N(k/t)  |   N(k)   | P(k/t)  | P(t/k)  |
    ================================================
    | 32   |   32.00 |   142.00 |   29.09 |   22.54 |
    | 33   |   78.00 |   340.00 |   70.91 |   22.94 |
    ================================================


 TEST SET

    ================================================
    |      | N(k/t)  |   N(k)   | P(k/t)  | P(t/k)  |
    ================================================
    | 32   |   19.00 |    70.00 |   30.16 |   27.14 |
    | 33   |   44.00 |   164.00 |   69.84 |   26.83 |
    ================================================

TREE CRITERION  0.412562

    ========================================================
    | Ord |   variable       |    value      |   criterion |
    ========================================================
    |   1 |(   7) H_P07      | 000100        |      0.3939 |
    |   2 |(   6) HP06D      | 100           |      0.3967 |
    |   3 |(  25) 20B        | 11110011000   |      0.3979 |
    |   4 |(  24) 20A        | 00000100000   |      0.4027 |
    |   5 |(   9) H_R01CAT   | 010100000     |      0.4035 |
    ========================================================


SPLITTING NODE:    4

VARIABLE      : (   7) H_P07
SPLIT         : 000100  ( 1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
         (  4) 2
MODALITIES BELONG RIGHT NODE :
         (  1) 1
         (  2) 3
         (  3) 5
         (  5) 4
         (  6) 8
CRITERION     : 0.393939

     LEARNING SET

    ================================================
    |      | left node  | right node | Row totals |
    |  nod |         8  |        9   |        4   |
    ================================================
    | 32   |      2.00  |     30.00  |     32.00  |
    | 33   |      0.00  |     78.00  |     78.00  |
    ================================================
    |  Tot |      2.00  |    108.00  |    110.00  |
    ================================================

     TEST SET

    ================================================
    |      | left node  | right node | Row totals |
    |  nod |         8  |        9   |        4   |
    ================================================
    | 32   |      0.00  |     19.00  |     19.00  |
    | 33   |      2.00  |     42.00  |     44.00  |
    ================================================
    |  Tot |      2.00  |     61.00  |     63.00  |
    ================================================


 ================================
 | SPLIT OF A NODE       :     5 |
 ================================

 LEARNING SET

    ================================================
    |      | N(k/t)  |   N(k)   | P(k/t)  | P(t/k)  |
    ================================================
    | 32   |  109.00 |   142.00 |   74.66 |   76.76 |
    | 33   |   37.00 |   340.00 |   25.34 |   10.88 |
    ================================================
```

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	49.00	70.00	70.00	70.00
33	21.00	164.00	30.00	12.80

TREE CRITERION 0.378401

Ord	variable	value	criterion
1	(11) H_IPCFCAT	01000100	0.3154
2	(36) P56	001100000	0.3459
3	(7) H_P07	001100	0.3491
4	(9) H_R01CAT	000100100	0.3550
5	(25) 20B	01111000000	0.3613

SPLITTING NODE: 5

VARIABLE : (11) H_IPCFCAT
SPLIT : 01000100 (1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
(2) 100-300
(6) 0-100
MODALITIES BELONG RIGHT NODE :
(1) 300-600
(3) 600-1000
(4) 0
(5) 1000-2000
(7) 2000-4000
(8) 4000-9000
CRITERION : 0.315402

LEARNING SET

nod	left node 10	right node 11	Row totals 5
32	17.00	92.00	109.00
33	21.00	16.00	37.00
Tot	38.00	108.00	146.00

TEST SET

nod	left node 10	right node 11	Row totals 5
32	9.00	40.00	49.00
33	6.00	15.00	21.00
Tot	15.00	55.00	70.00

SPLIT OF A NODE : 8

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	2.00	142.00	100.00	1.41
33	0.00	340.00	0.00	0.00

THIS STOP-SPLITTING RULE IS TRUE : The size of the node is too small
SIZE OF THE NODE 2.000000 VALUE OF STOP-SPLITTING RULE 15.000000
THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
SIZE OF THE NO-MAJORITY CLASSES 0.000000 VALUE OF STOP-SPLITTING RULE 15.000000

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	0.00	70.00	0.00	0.00
33	2.00	164.00	100.00	1.22

THIS NODE IS A TERMINAL NODE

SPLIT OF A NODE : 9

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	30.00	142.00	27.78	21.13
33	78.00	340.00	72.22	22.94

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	19.00	70.00	31.15	27.14
33	42.00	164.00	68.85	25.61

TREE CRITERION 0.401235

Ord	variable	value	criterion

Ord	variable	value	criterion
1	(25) 20B	11110011000	0.3866
2	(24) 20A	00000100000	0.3899
3	(9) H_R01CAT	010110000	0.3926
4	(6) HP06D	100	0.3926
5	(36) P56	000010000	0.3946

SPLITTING NODE: 9

VARIABLE : (25) 20B
SPLIT : 11110011000 (1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
(1) S/D
(2) 2
(3) 3
(4) 6
(7) 8
(8) 4
MODALITIES BELONG RIGHT NODE :
(5) 1
(6) 7
(9) 5
(10) 9
(11) 0
CRITERION : 0.386556

LEARNING SET

nod	left node 18	right node 19	Row totals 9
32	27.00	3.00	30.00
33	78.00	-0.00	78.00
Tot	105.00	3.00	108.00

TEST SET

nod	left node 18	right node 19	Row totals 9
32	19.00	0.00	19.00
33	42.00	0.00	42.00
Tot	61.00	0.00	61.00

SPLIT OF A NODE : 10

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	17.00	142.00	44.74	11.97
33	21.00	340.00	55.26	6.18

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	9.00	70.00	60.00	12.86
33	6.00	164.00	40.00	3.66

TREE CRITERION 0.494460

Ord	variable	value	criterion
1	(2) HP02	100110000	0.4216
2	(36) P56	011000100	0.4528
3	(9) H_R01CAT	111011000	0.4602
4	(7) H_P07	111000	0.4605
5	(1) HP01	100000	0.4605

SPLITTING NODE: 10

VARIABLE : (2) HP02
SPLIT : 100110000 (1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
(1) 3
(4) 5
(5) 6
MODALITIES BELONG RIGHT NODE :
(2) 2
(3) 4
(6) 1
(7) 7
(8) 8
(9) 9
CRITERION : 0.421642

LEARNING SET

nod	left node 20	right node 21	Row totals 10
32	4.00	13.00	17.00
33	13.00	8.00	21.00
Tot	17.00	21.00	38.00

TEST SET

nod	left node 20	right node 21	Row totals 10
32	3.00	6.00	9.00
33	2.00	4.00	6.00
Tot	5.00	10.00	15.00

SPLIT OF A NODE : 11

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	92.00	142.00	85.19	64.79
33	16.00	340.00	14.81	4.71

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	40.00	70.00	72.73	57.14
33	15.00	164.00	27.27	9.15

TREE CRITERION 0.252401

Ord	variable	value	criterion
1	(7) H_P07	001100	0.2245
2	(12) H_CAT60	110	0.2412
3	(17) H14	11000	0.2422
4	(2) HP02	001001000	0.2431
5	(36) P56	110111000	0.2435

SPLITTING NODE: 11

VARIABLE : (7) H_P07
SPLIT : 001100 (1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
(3) 5
(4) 2
MODALITIES BELONG RIGHT NODE :
(1) 1
(2) 3
(5) 4
(6) 8
CRITERION : 0.224537

LEARNING SET

nod	left node 22	right node 23	Row totals 11
32	1.00	91.00	92.00
33	3.00	13.00	16.00
Tot	4.00	104.00	108.00

TEST SET

nod	left node 22	right node 23	Row totals 11
32	5.00	35.00	40.00
33	0.00	15.00	15.00
Tot	5.00	50.00	55.00

SPLIT OF A NODE : 18

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	27.00	142.00	25.71	19.01
33	78.00	340.00	74.29	22.94

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	19.00	70.00	31.15	27.14
33	42.00	164.00	68.85	25.61

TREE CRITERION 0.382041

Ord	variable	value	criterion
1	(9) H_R01CAT	010110000	0.3719
2	(27) P22MCAT	1111100100	0.3732
3	(24) 20A	00100010000	0.3746
4	(35) P55	110	0.3764
5	(25) 20B	00000010000	0.3764

```
 SPLITTING NODE:   18

 VARIABLE      : (   9) H_R01CAT
 SPLIT         : 010110000  ( 1=left node, 0=right node)
 MODALITIES BELONG LEFT NODE :
         (  2) 4
         (  4) 7-8
         (  5) 5
 MODALITIES BELONG RIGHT NODE :
         (  1) 2
         (  3) 3
         (  6) 1
         (  7) 6
         (  8) 9-10
         (  9) 11-12-13-14
 CRITERION     : 0.371918

      LEARNING SET

    ================================================
    |       | left node  | right node | Row totals |
    |  nod  |       36   |      37    |      18    |
    ================================================
    | 32    |     15.00  |     12.00  |     27.00  |
    | 33    |     29.00  |     49.00  |     78.00  |
    ================================================
    |  Tot  |     44.00  |     61.00  |    105.00  |
    ================================================

      TEST SET

    ================================================
    |       | left node  | right node | Row totals |
    |  nod  |       36   |      37    |      18    |
    ================================================
    | 32    |      8.00  |     11.00  |     19.00  |
    | 33    |     14.00  |     28.00  |     42.00  |
    ================================================
    |  Tot  |     22.00  |     39.00  |     61.00  |
    ================================================


  ==================================
  | SPLIT OF A NODE       :    19 |
  ==================================

  LEARNING SET

     ========================================================
     |       |  N(k/t)   |   N(k)    |  P(k/t)   |  P(t/k)   |
     ========================================================
     | 32    |      3.00 |    142.00 |    100.00 |      2.11 |
     | 33    |      0.00 |    340.00 |      0.00 |      0.00 |
     ========================================================

 THIS STOP-SPLITTING RULE IS TRUE : The size of the node is too small
      SIZE OF THE NODE 3.000000  VALUE OF STOP-SPLITTING RULE 15.000000
 THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
      SIZE OF THE NO-MAJORITY CLASSES 0.000000  VALUE OF STOP-SPLITTING RULE 15.000000

  THIS NODE IS A TERMINAL NODE

 List of objects :

 ( 1)"2-97-1-1232"      ( 1)"2-97-1-1294"      ( 1)"3-98-1-0228"


  ==================================
  | SPLIT OF A NODE       :    20 |
  ==================================

  LEARNING SET

     ========================================================
     |       |  N(k/t)   |   N(k)    |  P(k/t)   |  P(t/k)   |
     ========================================================
     | 32    |      4.00 |    142.00 |     23.53 |      2.82 |
     | 33    |     13.00 |    340.00 |     76.47 |      3.82 |
     ========================================================

 THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
      SIZE OF THE NO-MAJORITY CLASSES 4.000000  VALUE OF STOP-SPLITTING RULE 15.000000

  TEST SET

     ========================================================
     |       |  N(k/t)   |   N(k)    |  P(k/t)   |  P(t/k)   |
     ========================================================
     | 32    |      3.00 |     70.00 |     60.00 |      4.29 |
     | 33    |      2.00 |    164.00 |     40.00 |      1.22 |
     ========================================================


  THIS NODE IS A TERMINAL NODE


  ==================================
  | SPLIT OF A NODE       :    21 |
  ==================================

  LEARNING SET

     ========================================================
     |       |  N(k/t)   |   N(k)    |  P(k/t)   |  P(t/k)   |
     ========================================================
     | 32    |     13.00 |    142.00 |     61.90 |      9.15 |
     | 33    |      8.00 |    340.00 |     38.10 |      2.35 |
     ========================================================

 THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
      SIZE OF THE NO-MAJORITY CLASSES 8.000000  VALUE OF STOP-SPLITTING RULE 15.000000

  TEST SET

     ========================================================
     |       |  N(k/t)   |   N(k)    |  P(k/t)   |  P(t/k)   |
     ========================================================
     | 32    |      6.00 |     70.00 |     60.00 |      8.57 |
```

```
| 33   |     4.00 |   164.00 |    40.00 |     2.44 |
=================================================

 THIS NODE IS A TERMINAL NODE

 ================================
 | SPLIT OF A NODE       :    22 |
 ================================

 LEARNING SET

    ======================================================
    |      | N(k/t)   |   N(k)   | P(k/t)   | P(t/k)   |
    ======================================================
    | 32   |     1.00 |   142.00 |    25.00 |     0.70 |
    | 33   |     3.00 |   340.00 |    75.00 |     0.88 |
    ======================================================

THIS STOP-SPLITTING RULE IS TRUE : The size of the node is too small
     SIZE OF THE NODE 4.000000  VALUE OF STOP-SPLITTING RULE 15.000000
THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
     SIZE OF THE NO-MAJORITY CLASSES 1.000000  VALUE OF STOP-SPLITTING RULE 15.000000

 TEST SET

    ======================================================
    |      | N(k/t)   |   N(k)   | P(k/t)   | P(t/k)   |
    ======================================================
    | 32   |     5.00 |    70.00 |   100.00 |     7.14 |
    | 33   |     0.00 |   164.00 |     0.00 |     0.00 |
    ======================================================

 THIS NODE IS A TERMINAL NODE

 ================================
 | SPLIT OF A NODE       :    23 |
 ================================

 LEARNING SET

    ======================================================
    |      | N(k/t)   |   N(k)   | P(k/t)   | P(t/k)   |
    ======================================================
    | 32   |    91.00 |   142.00 |    87.50 |    64.08 |
    | 33   |    13.00 |   340.00 |    12.50 |     3.82 |
    ======================================================

THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
     SIZE OF THE NO-MAJORITY CLASSES 13.000000  VALUE OF STOP-SPLITTING RULE 15.000000

 TEST SET

    ======================================================
    |      | N(k/t)   |   N(k)   | P(k/t)   | P(t/k)   |
    ======================================================
    | 32   |    35.00 |    70.00 |    70.00 |    50.00 |
    | 33   |    15.00 |   164.00 |    30.00 |     9.15 |
    ======================================================

 THIS NODE IS A TERMINAL NODE

 ================================
 | SPLIT OF A NODE       :    36 |
 ================================

 LEARNING SET

    ======================================================
    |      | N(k/t)   |   N(k)   | P(k/t)   | P(t/k)   |
    ======================================================
    | 32   |    15.00 |   142.00 |    34.09 |    10.56 |
    | 33   |    29.00 |   340.00 |    65.91 |     8.53 |
    ======================================================

THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
     SIZE OF THE NO-MAJORITY CLASSES 15.000000  VALUE OF STOP-SPLITTING RULE 15.000000

 TEST SET

    ======================================================
    |      | N(k/t)   |   N(k)   | P(k/t)   | P(t/k)   |
    ======================================================
    | 32   |     8.00 |    70.00 |    36.36 |    11.43 |
    | 33   |    14.00 |   164.00 |    63.64 |     8.54 |
    ======================================================

 THIS NODE IS A TERMINAL NODE

 ================================
 | SPLIT OF A NODE       :    37 |
 ================================

 LEARNING SET

    ======================================================
    |      | N(k/t)   |   N(k)   | P(k/t)   | P(t/k)   |
    ======================================================
    | 32   |    12.00 |   142.00 |    19.67 |     8.45 |
    | 33   |    49.00 |   340.00 |    80.33 |    14.41 |
    ======================================================

THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
     SIZE OF THE NO-MAJORITY CLASSES 12.000000  VALUE OF STOP-SPLITTING RULE 15.000000

 TEST SET

    ======================================================
    |      | N(k/t)   |   N(k)   | P(k/t)   | P(t/k)   |
    ======================================================
    | 32   |    11.00 |    70.00 |    28.21 |    15.71 |
    | 33   |    28.00 |   164.00 |    71.79 |    17.07 |
    ======================================================

 THIS NODE IS A TERMINAL NODE

    CONFUSION MATRIX FOR TRAINNING SET
```

	32	33	Total
32	109	33	142
33	21	319	340
Total	130	352	482

MISCLASSIFICATION RATE BY CLASS

TRUE CLASS	(ERROR /SIZE)	FREQUENCY
32	(33 / 142)	23.24
33	(21 / 340)	6.18
TOTAL	(54 / 482)	11.20

CONFUSION MATRIX FOR TEST SET

	32	33	Total
32	41	29	70
33	21	143	164
Total	62	172	234

MISCLASSIFICATION RATE BY CLASS

TRUE CLASS	(ERROR /SIZE)	FREQUENCY
32	(29 / 70)	41.43
33	(21 / 164)	12.80
TOTAL	(50 / 234)	21.37

NAME OF INTERNAL TREE FILE : C:\SODAS\FILIERES\ZHC8IF01.TREE

```
   ================================
   | EDITION OF DECISION TREE     |
   ================================


 PARAMETERS :
 Learning Set          :   482
 Number of variables :    30
 Max. number of nodes:    17
 Soft Assign           : ( 0 ) PURE
 Criterion coding      : ( 1 ) GINI
 Min. number of object by node    :       15
 Min. size of no-majority classes :       15
 Min. size of descendant nodes    :    15.00
 Frequency of test set            :    33.00

    + --- IF ASSERTION IS TRUE  (up)
    !
--- x [ ASSERTION ]
    !
    + --- IF ASSERTION IS FALSE (down)


               +----  [ 8 ]32  (     2.00      0.00 )
               !
          !----4[ H_P07 = 000100 ]
          !    !
          !    !          +----  [ 36 ]33  (    15.00     29.00 )
          !    !          !
          !    !    !---18[ H_R01CAT = 010110000 ]
          !    !    !     !
          !    !    !     +----  [ 37 ]33  (    12.00     49.00 )
          !    !    !
          !    !----9[ 20B = 11110011000 ]
          !         !
          !         +----  [ 19 ]32  (     3.00      0.00 )
          !
     !----2[ HP01 = 010000 ]
     !    !
     !    !          +----  [ 20 ]33  (     4.00     13.00 )
     !    !          !
     !    !    !---10[ HP02 = 100110000 ]
     !    !    !     !
     !    !    !     +----  [ 21 ]32  (    13.00      8.00 )
     !    !    !
     !    !----5[ H_IPCFCAT = 01000100 ]
     !         !
     !         !     +----  [ 22 ]33  (     1.00      3.00 )
     !         !     !
     !         !---11[ H_P07 = 001100 ]
     !               !
     !               +----  [ 23 ]32  (    91.00     13.00 )
     !
!----1[ HP06C = 1000 ]
     !
     +----  [ 3 ]33  (     1.00    225.00 )
```

Resultados del 2do. Grupo

```
-------------------------------------------------------
 BASE= C:\MIMI2\ANÁLIS-1\SODAS_G2.SDS
 Number of OS = 714
 Number of variables = 41
 METHOD=SODAS_TREE Version 1.2     INRIA 1998
-------------------------------------------------------

 Learning Set          :   482
 Number of variables :    30
 Max. number of nodes:    59
 Soft Assign           : ( 0 ) PURE
 Criterion coding      : ( 1 ) GINI
 Min. number of object by node    :       15
 Min. size of no-majority classes :       15
 Min. size of descendant nodes    :    15.00
 Frequency of test set            :    33.00

 GROUP OF PREDICATE VARIABLES :
 (   1 ) HP01                                          5 MODALITIES
 (   2 ) HP02                                          8 MODALITIES
 (   3 ) HP06A                                         2 MODALITIES
 (   4 ) HP06B                                         4 MODALITIES
 (   5 ) HP06C                                         5 MODALITIES
 (   6 ) HP06D                                         3 MODALITIES
 (   7 ) H_P07                                         6 MODALITIES
 (   8 ) H_P08                                         4 MODALITIES
 (   9 ) H_R01CAT                                     10 MODALITIES
 (  11 ) H_IPCFCAT                                     8 MODALITIES
 (  12 ) H_CAT60                                       3 MODALITIES
 (  14 ) H08                                          11 MODALITIES
 (  15 ) H12CAT                                        8 MODALITIES
 (  16 ) H13                                           2 MODALITIES
 (  17 ) H14                                           5 MODALITIES
 (  18 ) P11                                           8 MODALITIES
 (  19 ) ESTADO                                        3 MODALITIES
 (  22 ) P18B                                          5 MODALITIES
 (  24 ) 20A                                          11 MODALITIES
 (  25 ) 20B                                          11 MODALITIES
 (  26 ) 20C                                           6 MODALITIES
 (  27 ) P22MCAT                                      10 MODALITIES
 (  29 ) P24                                           6 MODALITIES
 (  30 ) P32DCAT                                       8 MODALITIES
 (  32 ) CODINGRE                                      3 MODALITIES
 (  34 ) P54                                           2 MODALITIES
 (  35 ) P55                                           3 MODALITIES
 (  36 ) P56                                           9 MODALITIES
 (  37 ) P58                                           3 MODALITIES
 (  39 ) P59                                           5 MODALITIES

 CLASSIFICATION VARIABLE :
 (  41 ) H_AGLOMERADO

 NUMBER OF A PRIORI CLASSES :   2
      ID_CLASS      NAME_CLASS
```

```
        1           32
        2           33

  CLASS  SIZE   LEARNING     TEST

    1    210      141         69
    2    504      341        163

 TOTAL   714      482        232
```

```
 ================================
 | SPLIT OF A NODE       :     1 |
 ================================
```

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	141.00	141.00	29.25	100.00
33	341.00	341.00	70.75	100.00

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	69.00	69.00	29.74	100.00
33	163.00	163.00	70.26	100.00

TREE CRITERION 0.413913

Ord	variable	value	criterion
1	(5) HP06C	10000	0.2905
2	(1) HP01	01000	0.3032
3	(11) H_IPCPCAT	11100100	0.3606
4	(36) P56	0100000010	0.3892
5	(9) H_R01CAT	1100000000	0.3894

```
SPLITTING NODE:    1

VARIABLE      : (   5) HP06C
SPLIT         : 10000  ( 1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
        (  1) 1
MODALITIES BELONG RIGHT NODE :
        (  2) 3
        (  3) 0
        (  4) 2
        (  5) 9
CRITERION     : 0.290543
```

LEARNING SET

nod	left node 2	right node 3	Row totals 1
32	136.00	5.00	141.00
33	125.00	216.00	341.00
Tot	261.00	221.00	482.00

TEST SET

nod	left node 2	right node 3	Row totals 1
32	68.00	1.00	69.00
33	64.00	99.00	163.00
Tot	132.00	100.00	232.00

```
 ================================
 | SPLIT OF A NODE       :     2 |
 ================================
```

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	136.00	141.00	52.11	96.45
33	125.00	341.00	47.89	36.66

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	68.00	69.00	51.52	98.55
33	64.00	163.00	48.48	39.26

TREE CRITERION 0.499112

Ord	variable	value	criterion
1	(1) HP01	01000	0.3812
2	(11) H_IPCPCAT	11100100	0.4696
3	(9) H_R01CAT	1100011000	0.4789
4	(2) HP02	11101000	0.4792
5	(24) 20A	01111100000	0.4793

SPLITTING NODE: 2

VARIABLE : (1) HP01
SPLIT : 01000 (1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
(2) 1
MODALITIES BELONG RIGHT NODE :
(1) 2
(3) 4
(4) 7
(5) 3
CRITERION : 0.381248

LEARNING SET

nod	left node 4	right node 5	Row totals 2
32	27.00	109.00	136.00
33	85.00	40.00	125.00
Tot	112.00	149.00	261.00

TEST SET

nod	left node 4	right node 5	Row totals 2
32	13.00	55.00	68.00
33	47.00	17.00	64.00
Tot	60.00	72.00	132.00

| SPLIT OF A NODE : 3 |

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	5.00	141.00	2.26	3.55
33	216.00	341.00	97.74	63.34

THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
SIZE OF THE NO-MAJORITY CLASSES 5.000000 VALUE OF STOP-SPLITTING RULE 15.000000

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	1.00	69.00	1.00	1.45
33	99.00	163.00	99.00	60.74

THIS NODE IS A TERMINAL NODE

| SPLIT OF A NODE : 4 |

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	27.00	141.00	24.11	19.15
33	85.00	341.00	75.89	24.93

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	13.00	69.00	21.67	18.84
33	47.00	163.00	78.33	28.83

TREE CRITERION 0.365912

Ord	variable	value	criterion
1	(2) HP02	11101000	0.3261
2	(6) HP06D	100	0.3450
3	(15) H12CAT	10000000	0.3493
4	(11) H_IPCFCAT	01001000	0.3523
5	(36) P56	110010110	0.3524

SPLITTING NODE: 4

VARIABLE : (2) HP02
SPLIT : 11101000 (1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
(1) 3
(2) 4
(3) 2
(5) 6
MODALITIES BELONG RIGHT NODE :
(4) 1
(6) 5
(7) 9
(8) 7
CRITERION : 0.326065

LEARNING SET

nod	left node 8	right node 9	Row totals 4
32	21.00	6.00	27.00
33	83.00	2.00	85.00
Tot	104.00	8.00	112.00

TEST SET

nod	left node 8	right node 9	Row totals 4
32	11.00	2.00	13.00
33	40.00	7.00	47.00
Tot	51.00	9.00	60.00

SPLIT OF A NODE : 5

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	109.00	141.00	73.15	77.30
33	40.00	341.00	26.85	11.73

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	55.00	69.00	76.39	79.71
33	17.00	163.00	23.61	10.43

TREE CRITERION 0.392775

Ord	variable	value	criterion
1	(11) H_IPCFCAT	10100100	0.3569
2	(12) H_CAT60	110	0.3752
3	(26) 20C	010110	0.3775
4	(9) H_R01CAT	0111100000	0.3787
5	(32) CODINGRE	010	0.3791

SPLITTING NODE: 5

VARIABLE : (11) H_IPCFCAT
SPLIT : 10100100 (1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
(1) 0
(3) 100-300
(6) 0-100
MODALITIES BELONG RIGHT NODE :
(2) 300-600
(4) 600-1000
(5) 1000-2000
(7) 2000-4000
(8) 4000-9000
CRITERION : 0.356905

LEARNING SET

nod	left node 10	right node 11	Row totals 5
32	32.00	77.00	109.00
33	25.00	15.00	40.00
Tot	57.00	92.00	149.00

TEST SET

nod	left node 10	right node 11	Row totals 5
32	21.00	34.00	55.00
33	12.00	5.00	17.00
Tot	33.00	39.00	72.00

SPLIT OF A NODE : 8

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	21.00	141.00	20.19	14.89
33	83.00	341.00	79.81	24.34

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)

32	11.00	69.00	21.57	15.94
33	40.00	163.00	78.43	24.54

TREE CRITERION 0.322300

Ord	variable	value	criterion
1	(7) H_P07	001100	0.3063
2	(30) P32DCAT	01100100	0.3088
3	(11) H_IPCFCAT	01001000	0.3136
4	(15) H12CAT	01011100	0.3138
5	(39) P59	01100	0.3144

SPLITTING NODE: 8

VARIABLE : (7) H_P07
SPLIT : 001100 (1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
(3) 3
(4) 5
MODALITIES BELONG RIGHT NODE :
(1) 1
(2) 4
(5) 2
(6) 8
CRITERION : 0.306319

LEARNING SET

nod	left node 16	right node 17	Row totals 8
32	6.00	15.00	21.00
33	8.00	75.00	83.00
Tot	14.00	90.00	104.00

TEST SET

nod	left node 16	right node 17	Row totals 8
32	1.00	10.00	11.00
33	6.00	34.00	40.00
Tot	7.00	44.00	51.00

SPLIT OF A NODE : 9

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	6.00	141.00	75.00	4.26
33	2.00	341.00	25.00	0.59

THIS STOP-SPLITTING RULE IS TRUE : The size of the node is too small
SIZE OF THE NODE 8.000000 VALUE OF STOP-SPLITTING RULE 15.000000
THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
SIZE OF THE NO-MAJORITY CLASSES 2.000000 VALUE OF STOP-SPLITTING RULE 15.000000

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	2.00	69.00	22.22	2.90
33	7.00	163.00	77.78	4.29

THIS NODE IS A TERMINAL NODE

SPLIT OF A NODE : 10

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	32.00	141.00	56.14	22.70
33	25.00	341.00	43.86	7.33

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	21.00	69.00	63.64	30.43
33	12.00	163.00	36.36	7.36

TREE CRITERION 0.492459

Ord	variable	value	criterion
1	(9) H_R01CAT	0110100000	0.4498
2	(18) P11	10000000	0.4659
3	(12) H_CAT60	100	0.4686
4	(15) H12CAT	11010110	0.4720
5	(37) P58	100	0.4759

```
SPLITTING NODE:   10

VARIABLE      : (   9) H_R01CAT
SPLIT         : 0110100000  ( 1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
         (  2) 2
         (  3) 5
         (  5) 3
MODALITIES BELONG RIGHT NODE :
         (  1) 1
         (  4) 6
         (  6) 4
         (  7) 9-10
         (  8) 7-8
         (  9) 11-12-13-14
         ( 10) 14
CRITERION     : 0.449773

     LEARNING SET

    ================================================
    |      | left node  | right node | Row totals |
    |  nod |        20  |       21   |       10   |
    ================================================
    | 32   |     11.00  |     21.00  |     32.00  |
    | 33   |     16.00  |      9.00  |     25.00  |
    ================================================
    |  Tot |     27.00  |     30.00  |     57.00  |
    ================================================

     TEST SET

    ================================================
    |      | left node  | right node | Row totals |
    |  nod |        20  |       21   |       10   |
    ================================================
    | 32   |      9.00  |     12.00  |     21.00  |
    | 33   |      6.00  |      6.00  |     12.00  |
    ================================================
    |  Tot |     15.00  |     18.00  |     33.00  |
    ================================================



 ================================
 | SPLIT OF A NODE       :    11 |
 ================================

 LEARNING SET

    ====================================================
    |      | N(k/t)   |   N(k)    |  P(k/t)   |  P(t/k)  |
    ====================================================
    | 32   |    77.00 |   141.00  |    83.70  |    54.61 |
    | 33   |    15.00 |   341.00  |    16.30  |     4.40 |
    ====================================================

THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
     SIZE OF THE NO-MAJORITY CLASSES 15.000000  VALUE OF STOP-SPLITTING RULE 15.000000

 TEST SET

    ====================================================
    |      | N(k/t)   |   N(k)    |  P(k/t)   |  P(t/k)  |
    ====================================================
    | 32   |    34.00 |    69.00  |    87.18  |    49.28 |
    | 33   |     5.00 |   163.00  |    12.82  |     3.07 |
    ====================================================


 THIS NODE IS A TERMINAL NODE

 ================================
 | SPLIT OF A NODE       :    16 |
 ================================

 LEARNING SET

    ====================================================
    |      | N(k/t)   |   N(k)    |  P(k/t)   |  P(t/k)  |
    ====================================================
    | 32   |     6.00 |   141.00  |    42.86  |     4.26 |
    | 33   |     8.00 |   341.00  |    57.14  |     2.35 |
    ====================================================

THIS STOP-SPLITTING RULE IS TRUE : The size of the node is too small
     SIZE OF THE NODE 14.000000  VALUE OF STOP-SPLITTING RULE 15.000000
THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
     SIZE OF THE NO-MAJORITY CLASSES 6.000000  VALUE OF STOP-SPLITTING RULE 15.000000

 TEST SET

    ====================================================
    |      | N(k/t)   |   N(k)    |  P(k/t)   |  P(t/k)  |
    ====================================================
    | 32   |     1.00 |    69.00  |    14.29  |     1.45 |
    | 33   |     6.00 |   163.00  |    85.71  |     3.68 |
    ====================================================


 THIS NODE IS A TERMINAL NODE

 ================================
 | SPLIT OF A NODE       :    17 |
 ================================

 LEARNING SET

    ====================================================
    |      | N(k/t)   |   N(k)    |  P(k/t)   |  P(t/k)  |
    ====================================================
    | 32   |    15.00 |   141.00  |    16.67  |    10.64 |
    | 33   |    75.00 |   341.00  |    83.33  |    21.99 |
    ====================================================

THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
     SIZE OF THE NO-MAJORITY CLASSES 15.000000  VALUE OF STOP-SPLITTING RULE 15.000000

 TEST SET

    ================================================
```

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	10.00	69.00	22.73	14.49
33	34.00	163.00	77.27	20.86

THIS NODE IS A TERMINAL NODE

SPLIT OF A NODE : 20

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	11.00	141.00	40.74	7.80
33	16.00	341.00	59.26	4.69

THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
SIZE OF THE NO-MAJORITY CLASSES 11.000000 VALUE OF STOP-SPLITTING RULE 15.000000

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	9.00	69.00	60.00	13.04
33	6.00	163.00	40.00	3.68

THIS NODE IS A TERMINAL NODE

SPLIT OF A NODE : 21

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	21.00	141.00	70.00	14.89
33	9.00	341.00	30.00	2.64

THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
SIZE OF THE NO-MAJORITY CLASSES 9.000000 VALUE OF STOP-SPLITTING RULE 15.000000

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	12.00	69.00	66.67	17.39
33	6.00	163.00	33.33	3.68

THIS NODE IS A TERMINAL NODE

CONFUSION MATRIX FOR TRAINNING SET

	32	33	Total
32	104	37	141
33	26	315	341
Total	130	352	482

MISCLASSIFICATION RATE BY CLASS

TRUE CLASS	(ERROR /SIZE)	FREQUENCY
32	(37 / 141)	26.24
33	(26 / 341)	7.62
TOTAL	(63 / 482)	13.07

CONFUSION MATRIX FOR TEST SET

	32	33	Total
32	48	21	69
33	18	145	163
Total	66	166	232

MISCLASSIFICATION RATE BY CLASS

TRUE CLASS	(ERROR /SIZE)	FREQUENCY
32	(21 / 69)	30.43
33	(18 / 163)	11.04
TOTAL	(39 / 232)	16.81

NAME OF INTERNAL TREE FILE : C:\MIMI2\ZHDMKP01.TREE

```
   ================================
   | EDITION OF DECISION TREE     |
   ================================


 PARAMETERS :
 Learning Set          :   482
 Number of variables :    30
 Max. number of nodes:    13
 Soft Assign           : ( 0 ) PURE
 Criterion coding      : ( 1 ) GINI
 Min. number of object by node     :       15
 Min. size of no-majority classes :       15
 Min. size of descendant nodes     :    15.00
 Frequency of test set             :    33.00

    + --- IF ASSERTION IS TRUE  (up)
    !
 --- x [ ASSERTION ]
    !
    + --- IF ASSERTION IS FALSE (down)


                    +-----  [ 16 ]33  (     6.00      8.00 )
                    !
               !----8[ H_P07 = 001100 ]
               !    !
               !    +-----  [ 17 ]33  (    15.00     75.00 )
               !
          !----4[ HP02 = 11101000 ]
          !    !
          !    +-----  [  9 ]32  (     6.00      2.00 )
          !
     !----2[ HP01 = 01000 ]
     !    !
     !    !          +-----  [ 20 ]33  (    11.00     16.00 )
     !    !          !
     !    !    !---10[ H_R01CAT = 0110100000 ]
     !    !    !     !
     !    !    !     +-----  [ 21 ]32  (    21.00      9.00 )
     !    !    !
     !    !----5[ H_IPCFCAT = 10100100 ]
     !         !
     !         +-----  [ 11 ]32  (    77.00     15.00 )
     !
 !----1[ HP06C = 10000 ]
     !
     +-----  [  3 ]33  (     5.00    216.00 )
```

Resultados del 3er. Grupo

```
-------------------------------------------------------
 BASE= C:\MIMI\TESIS\SODAS_G3.SDS
 Number of OS = 714
 Number of variables = 41
 METHOD=SODAS_TREE Version 1.2     INRIA 1998
-------------------------------------------------------

 Learning Set          :   482
 Number of variables :    30
 Max. number of nodes:    59
 Soft Assign           : ( 0 ) PURE
 Criterion coding      : ( 1 ) GINI
 Min. number of object by node     :       15
 Min. size of no-majority classes :       15
 Min. size of descendant nodes     :    15.00
 Frequency of test set             :    33.00

 GROUP OF PREDICATE VARIABLES :
 (   1 ) HP01                                          4 MODALITIES
 (   2 ) HP02                                          8 MODALITIES
 (   3 ) HP06A                                         2 MODALITIES
 (   4 ) HP06B                                         4 MODALITIES
 (   5 ) HP06C                                         4 MODALITIES
 (   6 ) HP06D                                         3 MODALITIES
 (   7 ) H_P07                                         6 MODALITIES
 (   8 ) H_P08                                         3 MODALITIES
 (   9 ) H_R01CAT                                     10 MODALITIES
 (  11 ) H_IPCFCAT                                     8 MODALITIES
 (  12 ) H_CAT60                                       3 MODALITIES
 (  14 ) H08                                          11 MODALITIES
 (  15 ) H12CAT                                        8 MODALITIES
 (  16 ) H13                                           2 MODALITIES
 (  17 ) H14                                           6 MODALITIES
 (  18 ) P11                                           8 MODALITIES
 (  19 ) ESTADO                                        3 MODALITIES
 (  22 ) P18B                                          5 MODALITIES
 (  24 ) 20A                                          11 MODALITIES
 (  25 ) 20B                                          11 MODALITIES
 (  26 ) 20C                                           7 MODALITIES
 (  27 ) P22MCAT                                      10 MODALITIES
 (  29 ) P24                                           6 MODALITIES
 (  30 ) P32DCAT                                       8 MODALITIES
 (  32 ) CODINGRE                                      3 MODALITIES
 (  34 ) P54                                           2 MODALITIES
 (  35 ) P55                                           3 MODALITIES
 (  36 ) P56                                           9 MODALITIES
 (  37 ) P58                                           3 MODALITIES
 (  39 ) P59                                           5 MODALITIES

CLASSIFICATION VARIABLE :
(  41 ) H_AGLOMERADO

NUMBER OF A PRIORI CLASSES :   2
     ID_CLASS     NAME_CLASS
        1            33
        2            32


  CLASS  SIZE   LEARNING     TEST

    1    504       341        163
    2    210       141         69
```

TOTAL 714 482 232

SPLIT OF A NODE : 1

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
33	341.00	341.00	70.75	100.00
32	141.00	141.00	29.25	100.00

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
33	163.00	163.00	70.26	100.00
32	69.00	69.00	29.74	100.00

TREE CRITERION 0.413913

Ord	variable	value	criterion
1	(5) HP06C	1000	0.2833
2	(1) HP01	1010	0.2836
3	(36) P56	111110000	0.3768
4	(11) H_IPCFCAT	10111110	0.3783
5	(4) HP06B	1000	0.3899

SPLITTING NODE: 1

VARIABLE : (5) HP06C
SPLIT : 1000 (1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
(1) 1
MODALITIES BELONG RIGHT NODE :
(2) 3
(3) 2
(4) 0
CRITERION : 0.283287

LEARNING SET

nod	left node 2	right node 3	Row totals 1
33	124.00	217.00	341.00
32	138.00	3.00	141.00
Tot	262.00	220.00	482.00

TEST SET

nod	left node 2	right node 3	Row totals 1
33	69.00	94.00	163.00
32	69.00	0.00	69.00
Tot	138.00	94.00	232.00

SPLIT OF A NODE : 2

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
33	124.00	341.00	47.33	36.36
32	138.00	141.00	52.67	97.87

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
33	69.00	163.00	50.00	42.33
32	69.00	69.00	50.00	100.00

TREE CRITERION 0.498572

Ord	variable	value	criterion
1	(1) HP01	0100	0.3649
2	(36) P56	111110000	0.4749
3	(11) H_IPCFCAT	11111110	0.4797
4	(9) H_R01CAT	0110100000	0.4803
5	(24) 20A	00010010000	0.4805

SPLITTING NODE: 2

VARIABLE : (1) HP01
SPLIT : 0100 (1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
(2) 1
MODALITIES BELONG RIGHT NODE :
(1) 2

```
         ( 3) 4
         ( 4) 7
CRITERION      : 0.364935

     LEARNING SET

    ================================================
    |       | left node  | right node | Row totals |
    |  nod  |          4 |          5 |          2 |
    ================================================
    | 33    |      86.00 |      38.00 |     124.00 |
    | 32    |      25.00 |     113.00 |     138.00 |
    ================================================
    |  Tot  |     111.00 |     151.00 |     262.00 |
    ================================================

     TEST SET

    ================================================
    |       | left node  | right node | Row totals |
    |  nod  |          4 |          5 |          2 |
    ================================================
    | 33    |      50.00 |      19.00 |      69.00 |
    | 32    |      13.00 |      56.00 |      69.00 |
    ================================================
    |  Tot  |      63.00 |      75.00 |     138.00 |
    ================================================


  ================================
  | SPLIT OF A NODE       :     3 |
  ================================

  LEARNING SET

     ======================================================
     |       | N(k/t)   |   N(k)   | P(k/t)   | P(t/k)   |
     ======================================================
     | 33    |   217.00 |   341.00 |    98.64 |    63.64 |
     | 32    |     3.00 |   141.00 |     1.36 |     2.13 |
     ======================================================

 THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
      SIZE OF THE NO-MAJORITY CLASSES 3.000000  VALUE OF STOP-SPLITTING RULE 15.000000

  TEST SET

     ======================================================
     |       | N(k/t)   |   N(k)   | P(k/t)   | P(t/k)   |
     ======================================================
     | 33    |    94.00 |   163.00 |   100.00 |    57.67 |
     | 32    |     0.00 |    69.00 |     0.00 |     0.00 |
     ======================================================


  THIS NODE IS A TERMINAL NODE
                    ;
  ================================
  | SPLIT OF A NODE       :     4 |
  ================================

  LEARNING SET

     ======================================================
     |       | N(k/t)   |   N(k)   | P(k/t)   | P(t/k)   |
     ======================================================
     | 33    |    86.00 |   341.00 |    77.48 |    25.22 |
     | 32    |    25.00 |   141.00 |    22.52 |    17.73 |
     ======================================================


  TEST SET

     ======================================================
     |       | N(k/t)   |   N(k)   | P(k/t)   | P(t/k)   |
     ======================================================
     | 33    |    50.00 |   163.00 |    79.37 |    30.67 |
     | 32    |    13.00 |    69.00 |    20.63 |    18.84 |
     ======================================================

TREE CRITERION  0.348998

     ============================================================
     |  Ord |   variable        |     value      |    criterion |
     ============================================================
     |    1 |(   2) HP02        | 00010100       |       0.3272 |
     |    2 |(  11) H_IPCFCAT   | 10111100       |       0.3282 |
     |    3 |(  24) 20A         | 11000001000    |       0.3370 |
     |    4 |(  25) 20B         | 11001100000    |       0.3384 |
     |    5 |(  36) P56         | 111111000      |       0.3393 |
     ============================================================


SPLITTING NODE:    4

VARIABLE       : (   2) HP02
SPLIT          : 00010100  ( 1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
         (  4) 5
         (  6) 7
MODALITIES BELONG RIGHT NODE :
         (  1) 3
         (  2) 2
         (  3) 4
         (  5) 1
         (  7) 6
         (  8) 8
CRITERION      : 0.327203

     LEARNING SET

    ================================================
    |       | left node  | right node | Row totals |
    |  nod  |          8 |          9 |          4 |
    ================================================
    | 33    |       7.00 |      79.00 |      86.00 |
    | 32    |       7.00 |      18.00 |      25.00 |
    ================================================
    |  Tot  |      14.00 |      97.00 |     111.00 |
    ================================================
```

TEST SET

nod	left node	right node	Row totals
33	4.00	46.00	50.00
32	1.00	12.00	13.00
Tot	5.00	58.00	63.00

SPLIT OF A NODE : 5

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
33	38.00	341.00	25.17	11.14
32	113.00	141.00	74.83	80.14

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
33	19.00	163.00	25.33	11.66
32	56.00	69.00	74.67	81.16

TREE CRITERION 0.376650

Ord	variable	value	criterion
1	(12) H_CAT60	100	0.3510
2	(9) H_R01CAT	0110100000	0.3530
3	(2) HP02	01101010	0.3549
4	(11) H_IPCFCAT	11111110	0.3561
5	(24) 20A	00010010000	0.3624

SPLITTING NODE: 5

VARIABLE : (12) H_CAT60
SPLIT : 100 (1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
(1) 1
MODALITIES BELONG RIGHT NODE :
(2) 2
(3) 3
CRITERION : 0.350993

LEARNING SET

nod	left node	right node	Row totals
33	13.00	25.00	38.00
32	13.00	100.00	113.00
Tot	26.00	125.00	151.00

TEST SET

nod	left node	right node	Row totals
33	2.00	17.00	19.00
32	11.00	45.00	56.00
Tot	13.00	62.00	75.00

SPLIT OF A NODE : 8

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
33	7.00	341.00	50.00	2.05
32	7.00	141.00	50.00	4.96

THIS STOP-SPLITTING RULE IS TRUE : The size of the node is too small
SIZE OF THE NODE 14.000000 VALUE OF STOP-SPLITTING RULE 15.000000
THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
SIZE OF THE NO-MAJORITY CLASSES 7.000000 VALUE OF STOP-SPLITTING RULE 15.000000

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
33	4.00	163.00	80.00	2.45
32	1.00	69.00	20.00	1.45

THIS NODE IS A TERMINAL NODE

SPLIT OF A NODE : 9

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
33	79.00	341.00	81.44	23.17
32	18.00	141.00	18.56	12.77

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
33	46.00	163.00	79.31	28.22
32	12.00	69.00	20.69	17.39

TREE CRITERION 0.302264

Ord	variable	value	criterion
1	(11) H_IPCPCAT	10111000	0.2809
2	(24) 20A	00111110000	0.2892
3	(2) HP02	10100010	0.2907
4	(36) P56	101100000	0.2911
5	(25) 20B	11001100000	0.2924

SPLITTING NODE: 9

VARIABLE : (11) H_IPCPCAT
SPLIT : 10111000 (1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
(1) 300-600
(3) 600-1000
(4) 0
(5) 2000-4000
MODALITIES BELONG RIGHT NODE :
(2) 100-300
(6) 1000-2000
(7) 4000-9000
(8) 0-100
CRITERION : 0.280908

LEARNING SET

nod	left node 18	right node 19	Row totals 9
33	39.00	40.00	79.00
32	15.00	3.00	18.00
Tot	54.00	43.00	97.00

TEST SET

nod	left node 18	right node 19	Row totals 9
33	20.00	26.00	46.00
32	9.00	3.00	12.00
Tot	29.00	29.00	58.00

SPLIT OF A NODE : 10

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
33	13.00	341.00	50.00	3.81
32	13.00	141.00	50.00	9.22

THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
SIZE OF THE NO-MAJORITY CLASSES 13.000000 VALUE OF STOP-SPLITTING RULE 15.000000

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
33	2.00	163.00	15.38	1.23
32	11.00	69.00	84.62	15.94

THIS NODE IS A TERMINAL NODE

SPLIT OF A NODE : 11

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
33	25.00	341.00	20.00	7.33
32	100.00	141.00	80.00	70.92

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)

| 33 | 17.00 | 163.00 | 27.42 | 10.43 |
| 32 | 45.00 | 69.00 | 72.58 | 65.22 |

TREE CRITERION 0.320000

Ord	variable	value	criterion
1	(11) H_IPCFCAT	11111110	0.3000
2	(25) 20B	11111110000	0.3038
3	(9) H_R01CAT	0110100000	0.3040
4	(2) HP02	01111010	0.3047
5	(36) P56	111110010	0.3060

SPLITTING NODE: 11

VARIABLE : (11) H_IPCFCAT
SPLIT : 11111110 (1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
(1) 300-600
(2) 100-300
(3) 600-1000
(4) 0
(5) 2000-4000
(6) 1000-2000
(7) 4000-9000
MODALITIES BELONG RIGHT NODE :
(8) 0-100
CRITERION : 0.300000

LEARNING SET

nod	left node 22	right node 23	Row totals 11
33	22.00	3.00	25.00
32	99.00	1.00	100.00
Tot	121.00	4.00	125.00

TEST SET

nod	left node 22	right node 23	Row totals 11
33	16.00	1.00	17.00
32	44.00	1.00	45.00
Tot	60.00	2.00	62.00

SPLIT OF A NODE : 18

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
33	39.00	341.00	72.22	11.44
32	15.00	141.00	27.78	10.64

THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
SIZE OF THE NO-MAJORITY CLASSES 15.000000 VALUE OF STOP-SPLITTING RULE 15.000000

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
33	20.00	163.00	68.97	12.27
32	9.00	69.00	31.03	13.04

THIS NODE IS A TERMINAL NODE

SPLIT OF A NODE : 19

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
33	40.00	341.00	93.02	11.73
32	3.00	141.00	6.98	2.13

THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
SIZE OF THE NO-MAJORITY CLASSES 3.000000 VALUE OF STOP-SPLITTING RULE 15.000000

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
33	26.00	163.00	89.66	15.95
32	3.00	69.00	10.34	4.35

THIS NODE IS A TERMINAL NODE

SPLIT OF A NODE : 22

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
33	22.00	341.00	18.18	6.45
32	99.00	141.00	81.82	70.21

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
33	16.00	163.00	26.67	9.82
32	44.00	69.00	73.33	63.77

TREE CRITERION 0.297521

Ord	variable	value	criterion
1	(25) 20B	11111110000	0.2794
2	(30) P32DCAT	11111010	0.2835
3	(2) HP02	01111010	0.2839
4	(17) H14	000010	0.2857
5	(7) H_P07	111100	0.2864

SPLITTING NODE: 22

VARIABLE : (25) 20B
SPLIT : 11111110000 (1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
(1) S/D
(2) 4
(3) 3
(4) 7
(5) 8
(6) 2
(7) 9
MODALITIES BELONG RIGHT NODE :
(8) 6
(9) 1
(10) 0
(11) 5
CRITERION : 0.279367

LEARNING SET

nod	left node	right node	Row totals
33	20.00	2.00	22.00
32	99.00	-0.00	99.00
Tot	119.00	2.00	121.00

TEST SET

nod	left node	right node	Row totals
33	16.00	0.00	16.00
32	44.00	0.00	44.00
Tot	60.00	0.00	60.00

SPLIT OF A NODE : 23

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
33	3.00	341.00	75.00	0.88
32	1.00	141.00	25.00	0.71

THIS STOP-SPLITTING RULE IS TRUE : The size of the node is too small
SIZE OF THE NODE 4.000000 VALUE OF STOP-SPLITTING RULE 15.000000
THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
SIZE OF THE NO-MAJORITY CLASSES 1.000000 VALUE OF STOP-SPLITTING RULE 15.000000

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
33	1.00	163.00	50.00	0.61
32	1.00	69.00	50.00	1.45

THIS NODE IS A TERMINAL NODE

SPLIT OF A NODE : 44

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
33	20.00	341.00	16.81	5.87
32	99.00	141.00	83.19	70.21

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)

33	16.00	163.00	26.67	9.82
32	44.00	69.00	73.33	63.77

TREE CRITERION 0.279641

Ord	variable	value	criterion
1	(36) P56	111110000	0.2639
2	(30) P32DCAT	11111010	0.2649
3	(4) HP06B	1000	0.2679
4	(7) H_P07	111100	0.2679
5	(15) H12CAT	10111110	0.2691

SPLITTING NODE: 44

VARIABLE : (36) P56
SPLIT : 111110000 (1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
(1) 2
(2) 1
(3) 5
(4) S/D
(5) 3
MODALITIES BELONG RIGHT NODE :
(6) 8
(7) 4
(8) 7
(9) 6
CRITERION : 0.263911

LEARNING SET

nod	left node 88	right node 89	Row totals 44
33	19.00	1.00	20.00
32	70.00	29.00	99.00
Tot	89.00	30.00	119.00

TEST SET

nod	left node 88	right node 89	Row totals 44
33	12.00	4.00	16.00
32	30.00	14.00	44.00
Tot	42.00	18.00	60.00

SPLIT OF A NODE : 45

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
33	2.00	341.00	100.00	0.59
32	0.00	141.00	0.00	0.00

THIS STOP-SPLITTING RULE IS TRUE : The size of the node is too small
SIZE OF THE NODE 2.000000 VALUE OF STOP-SPLITTING RULE 15.000000
THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
SIZE OF THE NO-MAJORITY CLASSES 0.000000 VALUE OF STOP-SPLITTING RULE 15.000000

THIS NODE IS A TERMINAL NODE

SPLIT OF A NODE : 88

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
33	19.00	341.00	21.35	5.57
32	70.00	141.00	78.65	49.65

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
33	12.00	163.00	28.57	7.36
32	30.00	69.00	71.43	43.48

TREE CRITERION 0.335816

Ord	variable	value	criterion
1	(30) P32DCAT	11111000	0.3182
2	(2) HP02	10010000	0.3211
3	(15) H12CAT	10101010	0.3217
4	(4) HP06B	1000	0.3218
5	(7) H_P07	101100	0.3218

SPLITTING NODE: 88

VARIABLE : (30) P32DCAT
SPLIT : 11111000 (1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :

```
          (  1) 0
          (  2) 181-360
          (  3) 16-30
          (  4) 61-180
          (  5) 361-3070
 MODALITIES BELONG RIGHT NODE :
          (  6) 1-7
          (  7) 31-60
          (  8) 8-15
 CRITERION      : 0.318192

      LEARNING SET

     ================================================
     |       | left node  | right node | Row totals |
     |  nod  |       176  |      177   |       88   |
     ================================================
     | 33    |     19.00  |     0.00   |     19.00  |
     | 32    |     70.00  |    -0.00   |     70.00  |
     ================================================
     |  Tot  |     89.00  |    -0.00   |     89.00  |
     ================================================

      TEST SET

     ================================================
     |       | left node  | right node | Row totals |
     |  nod  |       176  |      177   |       88   |
     ================================================
     | 33    |     12.00  |     0.00   |     12.00  |
     | 32    |     30.00  |     0.00   |     30.00  |
     ================================================
     |  Tot  |     42.00  |     0.00   |     42.00  |
     ================================================


  THIS NODE IS A TERMINAL NODE

  ==================================
  | SPLIT OF A NODE       :    89 |
  ==================================

  LEARNING SET

     ======================================================
     |       |  N(k/t)  |   N(k)    |  P(k/t)  |  P(t/k)  |
     ======================================================
     | 33    |     1.00 |   341.00  |     3.33 |     0.29 |
     | 32    |    29.00 |   141.00  |    96.67 |    20.57 |
     ======================================================

 THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
      SIZE OF THE NO-MAJORITY CLASSES 1.000000  VALUE OF STOP-SPLITTING RULE 15.000000

  TEST SET

     ======================================================
     |       |  N(k/t)  |   N(k)    |  P(k/t)  |  P(t/k)  |
     ======================================================
     | 33    |     4.00 |   163.00  |    22.22 |     2.45 |
     | 32    |    14.00 |    69.00  |    77.78 |    20.29 |
     ======================================================


  THIS NODE IS A TERMINAL NODE

     CONFUSION MATRIX FOR TRAINNING SET

     ==========================================
     |      | 33        | 32        |  Total    |
     ==========================================
     | 33   |      321  |       20  |      341  |
     | 32   |       42  |       99  |      141  |
     ==========================================
     | Total|      363  |      119  |      482  |
     ==========================================

  MISCLASSIFICATION RATE BY CLASS

     TRUE CLASS             (  ERROR /SIZE    )   FREQUENCY
     33                     (     20 /    341 )      5.87
     32                     (     42 /    141 )     29.79

     TOTAL                  (     62 /    482 )     12.86

     CONFUSION MATRIX FOR TEST SET

     ==========================================
     |      | 33        | 32        |  Total    |
     ==========================================
     | 33   |      147  |       16  |      163  |
     | 32   |       25  |       44  |       69  |
     ==========================================
     | Total|      172  |       60  |      232  |
     ==========================================

  MISCLASSIFICATION RATE BY CLASS

     TRUE CLASS             (  ERROR /SIZE    )   FREQUENCY
     33                     (     16 /    163 )      9.82
     32                     (     25 /     69 )     36.23

     TOTAL                  (     41 /    232 )     17.67

 NAME OF INTERNAL TREE FILE : C:\SODAS\Tmp\ZHDM2A01.TREE
```

```
    ==================================
    | EDITION OF DECISION TREE       |
    ==================================

  PARAMETERS :
  Learning Set          :   482
  Number of variables :    30
  Max. number of nodes:    17
  Soft Assign           : ( 0 ) PURE
  Criterion coding      : ( 1 ) GINI
  Min. number of object by node     :       15
  Min. size of no-majority classes :       15
  Min. size of descendant nodes     :    15.00
  Frequency of test set             :    33.00

     + --- IF ASSERTION IS TRUE  (up)
     !
  --- x [ ASSERTION ]
     !
     + --- IF ASSERTION IS FALSE (down)


                 +----  [ 8 ]33  (     7.00      7.00 )
                 !
            !----4[ HP02 = 00010100 ]
            !    !
            !    !    +----  [ 18 ]33  (    39.00     15.00 )
            !    !    !
            !    !----9[ H_IPCFCAT = 10111000 ]
            !         !
            !         +----  [ 19 ]33  (    40.00      3.00 )
            !
       !----2[ HP01 = 0100 ]
       !    !
       !    !    +----  [ 10 ]33  (    13.00     13.00 )
       !    !    !
       !    !----5[ H_CAT60 = 100 ]
       !         !
       !         !               +----  [ 88 ]32  (    19.00     70.00 )
       !         !               !
       !         !         !---44[ P56 = 111110000 ]
       !         !         !     !
       !         !         !     +----  [ 89 ]32  (     1.00     29.00 )
       !         !         !
       !         !    !---22[ 20B = 11111110000 ]
       !         !    !    !
       !         !    !    +----  [ 45 ]33  (     2.00      0.00 )
       !         !    !
       !         !---11[ H_IPCFCAT = 11111110 ]
       !              !
       !              +----  [ 23 ]33  (     3.00      1.00 )
       !
  !----1[ HP06C = 1000 ]
       !
       +----  [ 3 ]33  (   217.00      3.00 )
```

Resultados del 4to. Grupo

```
-------------------------------------------------------
 BASE= C:\MIMI2\SODAS_G4.SDS
 Number of OS = 699
 Number of variables = 41
 METHOD=SODAS_TREE Version 1.2     INRIA 1998
-------------------------------------------------------

 Learning Set          :   472
 Number of variables :    30
 Max. number of nodes:    59
 Soft Assign           : ( 0 ) PURE
 Criterion coding      : ( 1 ) GINI
 Min. number of object by node     :       20
 Min. size of no-majority classes :       20
 Min. size of descendant nodes     :    20.00
 Frequency of test set             :    33.00

 GROUP OF PREDICATE VARIABLES :
 (   1 ) HP01                                          6 MODALITIES
 (   2 ) HP02                                          8 MODALITIES
 (   3 ) HP06A                                         2 MODALITIES
 (   4 ) HP06B                                         4 MODALITIES
 (   5 ) HP06C                                         4 MODALITIES
 (   6 ) HP06D                                         3 MODALITIES
 (   7 ) H_P07                                         6 MODALITIES
 (   8 ) H_P08                                         2 MODALITIES
 (   9 ) H_R01CAT                                     10 MODALITIES
 (  11 ) H_IPCFCAT                                     8 MODALITIES
 (  12 ) H_CAT60                                       3 MODALITIES
 (  14 ) H08                                          11 MODALITIES
 (  15 ) H12CAT                                        8 MODALITIES
 (  16 ) H13                                           2 MODALITIES
 (  17 ) H14                                           5 MODALITIES
 (  18 ) P11                                           8 MODALITIES
 (  19 ) ESTADO                                        3 MODALITIES
 (  22 ) P18B                                          4 MODALITIES
 (  24 ) 20A                                          10 MODALITIES
 (  25 ) 20B                                          11 MODALITIES
 (  26 ) 20C                                           7 MODALITIES
 (  27 ) P22MCAT                                      10 MODALITIES
 (  29 ) P24                                           6 MODALITIES
 (  30 ) P32DCAT                                       8 MODALITIES
 (  32 ) CODINGRE                                      3 MODALITIES
 (  34 ) P54                                           2 MODALITIES
 (  35 ) P55                                           3 MODALITIES
 (  36 ) P56                                           9 MODALITIES
 (  37 ) P58                                           3 MODALITIES
 (  39 ) P59                                           5 MODALITIES

 CLASSIFICATION VARIABLE :
 (  41 ) H_AGLOMERADO

 NUMBER OF A PRIORI CLASSES :   2
      ID_CLASS     NAME_CLASS
```

```
          1           32
          2           33

   CLASS  SIZE   LEARNING     TEST

     1    205      137         68
     2    494      335        159

  TOTAL   699      472        227

 ================================
 | SPLIT OF A NODE        :    1 |
 ================================

 LEARNING SET

    ================================================
    |      | N(k/t)  |   N(k)   | P(k/t)  | P(t/k)  |
    ================================================
    | 32   |  137.00 |  137.00  |   29.03 |  100.00 |
    | 33   |  335.00 |  335.00  |   70.97 |  100.00 |
    ================================================


 TEST SET

    ================================================
    |      | N(k/t)  |   N(k)   | P(k/t)  | P(t/k)  |
    ================================================
    | 32   |   68.00 |   68.00  |   29.96 |  100.00 |
    | 33   |  159.00 |  159.00  |   70.04 |  100.00 |
    ================================================

TREE CRITERION  0.412013

    ============================================================
    |  Ord |  variable       |    value      |    criterion    |
    ============================================================
    |    1 |(   5) HP06C     | 1000          |          0.2730 |
    |    2 |(   1) HP01      | 110110        |          0.3097 |
    |    3 |(  11) H_IPCFCAT | 11010100      |          0.3693 |
    |    4 |(  36) P56       | 111101100     |          0.3787 |
    |    5 |(   9) H_R01CAT  | 1000100000    |          0.3862 |
    ============================================================


SPLITTING NODE:    1

VARIABLE      : (   5) HP06C
SPLIT         : 1000  ( 1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
         (  1) 1
MODALITIES BELONG RIGHT NODE :
         (  2) 0
         (  3) 3
         (  4) 2
CRITERION     : 0.273040

     LEARNING SET

    ================================================
    |      | left node  | right node | Row totals |
    |  nod |          2 |          3 |          1 |
    ================================================
    | 32   |     131.00 |       6.00 |     137.00 |
    | 33   |     106.00 |     229.00 |     335.00 |
    ================================================
    |  Tot |     237.00 |     235.00 |     472.00 |
    ================================================

     TEST SET

    ================================================
    |      | left node  | right node | Row totals |
    |  nod |          2 |          3 |          1 |
    ================================================
    | 32   |      67.00 |       1.00 |      68.00 |
    | 33   |      51.00 |     108.00 |     159.00 |
    ================================================
    |  Tot |     118.00 |     109.00 |     227.00 |
    ================================================


 ================================
 | SPLIT OF A NODE        :    2 |
 ================================

 LEARNING SET

    ================================================
    |      | N(k/t)  |   N(k)   | P(k/t)  | P(t/k)  |
    ================================================
    | 32   |  131.00 |  137.00  |   55.27 |   95.62 |
    | 33   |  106.00 |  335.00  |   44.73 |   31.64 |
    ================================================


TEST SET

    ================================================
    |      | N(k/t)  |   N(k)   | P(k/t)  | P(t/k)  |
    ================================================
    | 32   |   67.00 |   68.00  |   56.78 |   98.53 |
    | 33   |   51.00 |  159.00  |   43.22 |   32.08 |
    ================================================

TREE CRITERION  0.494436

    ============================================================
    |  Ord |  variable       |    value      |    criterion    |
    ============================================================
    |    1 |(   1) HP01      | 110100        |          0.4040 |
    |    2 |(   9) H_R01CAT  | 1001100000    |          0.4498 |
    |    3 |(  12) H_CAT60   | 110           |          0.4680 |
    |    4 |(  11) H_IPCFCAT | 11010100      |          0.4748 |
    |    5 |(   7) H_P07     | 001100        |          0.4776 |
    ============================================================
```

```
SPLITTING NODE:    2

VARIABLE      : (   1) HP01
SPLIT         : 110100   ( 1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
         (  1) 2
         (  2) 3
         (  4) 4
MODALITIES BELONG RIGHT NODE :
         (  3) 1
         (  5) 7
         (  6) 6
CRITERION     : 0.403975

     LEARNING SET

    ================================================
    |      | left node  | right node | Row totals |
    |  nod |          4 |          5 |          2 |
    ================================================
    | 32   |      98.00 |      33.00 |     131.00 |
    | 33   |      34.00 |      72.00 |     106.00 |
    ================================================
    |  Tot |     132.00 |     105.00 |     237.00 |
    ================================================

     TEST SET

    ================================================
    |      | left node  | right node | Row totals |
    |  nod |          4 |          5 |          2 |
    ================================================
    | 32   |      45.00 |      22.00 |      67.00 |
    | 33   |      14.00 |      37.00 |      51.00 |
    ================================================
    |  Tot |      59.00 |      59.00 |     118.00 |
    ================================================


  ==================================
  | SPLIT OF A NODE       :     3 |
  ==================================

  LEARNING SET

    ====================================================
    |      | N(k/t)  |  N(k)    | P(k/t)  | P(t/k)  |
    ====================================================
    | 32   |    6.00 |  137.00  |    2.55 |    4.38 |
    | 33   |  229.00 |  335.00  |   97.45 |   68.36 |
    ====================================================

 THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
      SIZE OF THE NO-MAJORITY CLASSES 6.000000  VALUE OF STOP-SPLITTING RULE 20.000000

  TEST SET

    ====================================================
    |      | N(k/t)  |  N(k)    | P(k/t)  | P(t/k)  |
    ====================================================
    | 32   |    1.00 |   68.00  |    0.92 |    1.47 |
    | 33   |  108.00 |  159.00  |   99.08 |   67.92 |
    ====================================================


  THIS NODE IS A TERMINAL NODE

  ==================================
  | SPLIT OF A NODE       :     4 |
  ==================================

  LEARNING SET

    ====================================================
    |      | N(k/t)  |  N(k)    | P(k/t)  | P(t/k)  |
    ====================================================
    | 32   |   98.00 |  137.00  |   74.24 |   71.53 |
    | 33   |   34.00 |  335.00  |   25.76 |   10.15 |
    ====================================================


  TEST SET

    ====================================================
    |      | N(k/t)  |  N(k)    | P(k/t)  | P(t/k)  |
    ====================================================
    | 32   |   45.00 |   68.00  |   76.27 |   66.18 |
    | 33   |   14.00 |  159.00  |   23.73 |    8.81 |
    ====================================================

TREE CRITERION  0.382461

    ============================================================
    |  Ord |   variable       |    value      |    criterion |
    ============================================================
    |    1 |(  11) H_IPCFCAT  | 11010100      |       0.3448 |
    |    2 |(   2) HP02       | 00111000      |       0.3609 |
    |    3 |(  36) P56        | 111100110     |       0.3626 |
    |    4 |(   7) H_P07      | 111100        |       0.3655 |
    |    5 |(   9) H_R01CAT   | 1001100000    |       0.3664 |
    ============================================================


SPLITTING NODE:    4

VARIABLE      : (  11) H_IPCFCAT
SPLIT         : 11010100   ( 1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
         (  1) 300_600
         (  2) 100_300
         (  4) 0
         (  6) 0_100
MODALITIES BELONG RIGHT NODE :
         (  3) 600_1000
         (  5) 1000_2000
         (  7) 2000_4000
         (  8) 4000_9000
CRITERION     : 0.344758

     LEARNING SET
```

nod	left node 8	right node 9	Row totals 4
32	56.00	42.00	98.00
33	31.00	3.00	34.00
Tot	87.00	45.00	132.00

TEST SET

nod	left node 8	right node 9	Row totals 4
32	28.00	17.00	45.00
33	11.00	3.00	14.00
Tot	39.00	20.00	59.00

SPLIT OF A NODE : 5

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	33.00	137.00	31.43	24.09
33	72.00	335.00	68.57	21.49

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	22.00	68.00	37.29	32.35
33	37.00	159.00	62.71	23.27

TREE CRITERION 0.431020

Ord	variable	value	criterion
1	(7) H_P07	110000	0.3814
2	(9) H_R01CAT	1101000000	0.3992
3	(6) HP06D	100	0.4034
4	(12) H_CAT60	110	0.4157
5	(30) P32DCAT	01100100	0.4204

SPLITTING NODE: 5

VARIABLE : (7) H_P07
SPLIT : 110000 (1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
(1) 1
(2) 3
MODALITIES BELONG RIGHT NODE :
(3) 4
(4) 5
(5) 2
(6) 8
CRITERION : 0.381353

LEARNING SET

nod	left node 10	right node 11	Row totals 5
32	25.00	8.00	33.00
33	70.00	2.00	72.00
Tot	95.00	10.00	105.00

TEST SET

nod	left node 10	right node 11	Row totals 5
32	19.00	3.00	22.00
33	36.00	1.00	37.00
Tot	55.00	4.00	59.00

SPLIT OF A NODE : 8

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	56.00	137.00	64.37	40.88
33	31.00	335.00	35.63	9.25

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	28.00	68.00	71.79	41.18
33	11.00	159.00	28.21	6.92

```
==================================================

TREE CRITERION  0.458713

=============================================================
|  Ord |   variable        |    value      |    criterion |
=============================================================
|    1 |(  12) H_CAT60     | 110           |        0.4326 |
|    2 |(   7) H_P07       | 001100        |        0.4365 |
|    3 |(   9) H_R01CAT    | 0000100000    |        0.4390 |
|    4 |(  18) P11         | 11011100      |        0.4401 |
|    5 |(   2) HP02        | 00111000      |        0.4420 |
=============================================================


SPLITTING NODE:    8

VARIABLE       : (  12) H_CAT60
SPLIT          : 110  ( 1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
          (  1) 2
          (  2) 3
MODALITIES BELONG RIGHT NODE :
          (  3) 1
CRITERION      : 0.432590

     LEARNING SET

    ================================================
    |       | left node  | right node | Row totals |
    |  nod  |        16  |       17   |        8   |
    ================================================
    | 32    |      49.00 |       7.00 |      56.00 |
    | 33    |      21.00 |      10.00 |      31.00 |
    ================================================
    |  Tot  |      70.00 |      17.00 |      87.00 |
    ================================================

     TEST SET

    ================================================
    |       | left node  | right node | Row totals |
    |  nod  |        16  |       17   |        8   |
    ================================================
    | 32    |      21.00 |       7.00 |      28.00 |
    | 33    |      10.00 |       1.00 |      11.00 |
    ================================================
    |  Tot  |      31.00 |       8.00 |      39.00 |
    ================================================



 ==================================
 | SPLIT OF A NODE       :     9 |
 ==================================

 LEARNING SET

    ========================================================
    |       |  N(k/t)  |   N(k)    |  P(k/t)  |  P(t/k)  |
    ========================================================
    | 32    |    42.00 |   137.00  |    93.33 |    30.66 |
    | 33    |     3.00 |   335.00  |     6.67 |     0.90 |
    ========================================================

THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
     SIZE OF THE NO-MAJORITY CLASSES 3.000000  VALUE OF STOP-SPLITTING RULE 20.000000

 TEST SET

    ========================================================
    |       |  N(k/t)  |   N(k)    |  P(k/t)  |  P(t/k)  |
    ========================================================
    | 32    |    17.00 |    68.00  |    85.00 |    25.00 |
    | 33    |     3.00 |   159.00  |    15.00 |     1.89 |
    ========================================================


 THIS NODE IS A TERMINAL NODE


 ==================================
 | SPLIT OF A NODE       :    10 |
 ==================================

 LEARNING SET

    ========================================================
    |       |  N(k/t)  |   N(k)    |  P(k/t)  |  P(t/k)  |
    ========================================================
    | 32    |    25.00 |   137.00  |    26.32 |    18.25 |
    | 33    |    70.00 |   335.00  |    73.68 |    20.90 |
    ========================================================


 TEST SET

    ========================================================
    |       |  N(k/t)  |   N(k)    |  P(k/t)  |  P(t/k)  |
    ========================================================
    | 32    |    19.00 |    68.00  |    34.55 |    27.94 |
    | 33    |    36.00 |   159.00  |    65.45 |    22.64 |
    ========================================================

TREE CRITERION  0.387812

    ============================================================
    |  Ord |   variable        |    value      |    criterion |
    ============================================================
    |    1 |(   9) H_R01CAT    | 1101000000    |        0.3491 |
    |    2 |(  39) P59         | 00010         |        0.3721 |
    |    3 |(  11) H_IPCFCAT   | 10110000      |        0.3728 |
    |    4 |(  12) H_CAT60     | 010           |        0.3748 |
    |    5 |(  24) 20A         | 0101111100    |        0.3756 |
    ============================================================


SPLITTING NODE:   10

VARIABLE       : (   9) H_R01CAT
SPLIT          : 1101000000  ( 1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
```

```
          (  1) 2
          (  2) 5
          (  4) 4
 MODALITIES BELONG RIGHT NODE :
          (  3) 3
          (  5) 1
          (  6) 6
          (  7) 9_10
          (  8) 7_8
          (  9) 14
          ( 10) 11_12_13_14
 CRITERION      : 0.349091

      LEARNING SET

     ================================================
     |       | left node  | right node | Row totals |
     |  nod  |        20  |       21   |       10   |
     ================================================
     | 32    |     21.00  |      4.00  |     25.00  |
     | 33    |     34.00  |     36.00  |     70.00  |
     ================================================
     |  Tot  |     55.00  |     40.00  |     95.00  |
     ================================================

      TEST SET

     ================================================
     |       | left node  | right node | Row totals |
     |  nod  |        20  |       21   |       10   |
     ================================================
     | 32    |     11.00  |      8.00  |     19.00  |
     | 33    |     22.00  |     14.00  |     36.00  |
     ================================================
     |  Tot  |     33.00  |     22.00  |     55.00  |
     ================================================



  ==================================
  | SPLIT OF A NODE        :    11 |
  ==================================

  LEARNING SET

     ======================================================
     |       |  N(k/t)  |   N(k)    |  P(k/t)  |  P(t/k)  |
     ======================================================
     | 32    |     8.00 |   137.00  |    80.00 |     5.84 |
     | 33    |     2.00 |   335.00  |    20.00 |     0.60 |
     ======================================================

 THIS STOP-SPLITTING RULE IS TRUE : The size of the node is too small
      SIZE OF THE NODE 10.000000  VALUE OF STOP-SPLITTING RULE 20.000000
 THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
      SIZE OF THE NO-MAJORITY CLASSES 2.000000  VALUE OF STOP-SPLITTING RULE 20.000000

  TEST SET

     ======================================================
     |       |  N(k/t)  |   N(k)    |  P(k/t)  |  P(t/k)  |
     ======================================================
     | 32    |     3.00 |    68.00  |    75.00 |     4.41 |
     | 33    |     1.00 |   159.00  |    25.00 |     0.63 |
     ======================================================


  THIS NODE IS A TERMINAL NODE


  ==================================
  | SPLIT OF A NODE        :    16 |
  ==================================

  LEARNING SET

     ======================================================
     |       |  N(k/t)  |   N(k)    |  P(k/t)  |  P(t/k)  |
     ======================================================
     | 32    |    49.00 |   137.00  |    70.00 |    35.77 |
     | 33    |    21.00 |   335.00  |    30.00 |     6.27 |
     ======================================================

  TEST SET

     ======================================================
     |       |  N(k/t)  |   N(k)    |  P(k/t)  |  P(t/k)  |
     ======================================================
     | 32    |    21.00 |    68.00  |    67.74 |    30.88 |
     | 33    |    10.00 |   159.00  |    32.26 |     6.29 |
     ======================================================

 TREE CRITERION  0.420000

     ============================================================
     |  Ord |   variable        |    value      |    criterion |
     ============================================================
     |    1 |(   7) H_P07       | 111100        |       0.3912 |
     |    2 |(  24) 20A         | 1000110000    |       0.3968 |
     |    3 |(  18) P11         | 01000100      |       0.3968 |
     |    4 |(  39) P59         | 01100         |       0.4023 |
     |    5 |(  25) 20B         | 00101000000   |       0.4055 |
     ============================================================


 SPLITTING NODE:   16

 VARIABLE       : (   7) H_P07
 SPLIT          : 111100  ( 1=left node, 0=right node)
 MODALITIES BELONG LEFT NODE :
          (  1) 1
          (  2) 3
          (  3) 4
          (  4) 5
 MODALITIES BELONG RIGHT NODE :
          (  5) 2
          (  6) 8
 CRITERION      : 0.391176

      LEARNING SET
```

	left node	right node	Row totals
nod	32	33	16
32	49.00	-0.00	49.00
33	19.00	2.00	21.00
Tot	68.00	2.00	70.00

TEST SET

	left node	right node	Row totals
nod	32	33	16
32	21.00	0.00	21.00
33	10.00	0.00	10.00
Tot	31.00	0.00	31.00

SPLIT OF A NODE : 17

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	7.00	137.00	41.18	5.11
33	10.00	335.00	58.82	2.99

THIS STOP-SPLITTING RULE IS TRUE : The size of the node is too small
SIZE OF THE NODE 17.000000 VALUE OF STOP-SPLITTING RULE 20.000000
THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
SIZE OF THE NO-MAJORITY CLASSES 7.000000 VALUE OF STOP-SPLITTING RULE 20.000000

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	7.00	68.00	87.50	10.29
33	1.00	159.00	12.50	0.63

THIS NODE IS A TERMINAL NODE

SPLIT OF A NODE : 20

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	21.00	137.00	38.18	15.33
33	34.00	335.00	61.82	10.15

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	11.00	68.00	33.33	16.18
33	22.00	159.00	66.67	13.84

TREE CRITERION 0.472066

Ord	variable	value	criterion
1	(11) H_IPCFCAT	10010000	0.4402
2	(39) P59	00010	0.4406
3	(27) P22MCAT	1001001100	0.4417
4	(2) HP02	10001000	0.4500
5	(36) P56	101111100	0.4536

SPLITTING NODE: 20

VARIABLE : (11) H_IPCFCAT
SPLIT : 10010000 (1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
(1) 300_600
(4) 0
MODALITIES BELONG RIGHT NODE :
(2) 100_300
(3) 600_1000
(5) 1000_2000
(6) 0_100
(7) 2000_4000
(8) 4000_9000
CRITERION : 0.440242

LEARNING SET

	left node	right node	Row totals
nod	40	41	20
32	13.00	8.00	21.00
33	12.00	22.00	34.00
Tot	25.00	30.00	55.00

TEST SET

nod	left node 40	right node 41	Row totals 20
32	5.00	6.00	11.00
33	7.00	15.00	22.00
Tot	12.00	21.00	33.00

SPLIT OF A NODE : 21

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	4.00	137.00	10.00	2.92
33	36.00	335.00	90.00	10.75

THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
SIZE OF THE NO-MAJORITY CLASSES 4.000000 VALUE OF STOP-SPLITTING RULE 20.000000

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	8.00	68.00	36.36	11.76
33	14.00	159.00	63.64	8.81

THIS NODE IS A TERMINAL NODE

SPLIT OF A NODE : 32

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	49.00	137.00	72.06	35.77
33	19.00	335.00	27.94	5.67

THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
SIZE OF THE NO-MAJORITY CLASSES 19.000000 VALUE OF STOP-SPLITTING RULE 20.000000

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	21.00	68.00	67.74	30.88
33	10.00	159.00	32.26	6.29

THIS NODE IS A TERMINAL NODE

SPLIT OF A NODE : 33

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	0.00	137.00	0.00	0.00
33	2.00	335.00	100.00	0.60

THIS STOP-SPLITTING RULE IS TRUE : The size of the node is too small
SIZE OF THE NODE 2.000000 VALUE OF STOP-SPLITTING RULE 20.000000
THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
SIZE OF THE NO-MAJORITY CLASSES 0.000000 VALUE OF STOP-SPLITTING RULE 20.000000

THIS NODE IS A TERMINAL NODE

SPLIT OF A NODE : 40

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	13.00	137.00	52.00	9.49
33	12.00	335.00	48.00	3.58

THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
SIZE OF THE NO-MAJORITY CLASSES 12.000000 VALUE OF STOP-SPLITTING RULE 20.000000

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	5.00	68.00	41.67	7.35
33	7.00	159.00	58.33	4.40

THIS NODE IS A TERMINAL NODE

SPLIT OF A NODE : 41

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	8.00	137.00	26.67	5.84
33	22.00	335.00	73.33	6.57

THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
SIZE OF THE NO-MAJORITY CLASSES 8.000000 VALUE OF STOP-SPLITTING RULE 20.000000

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	6.00	68.00	28.57	8.82
33	15.00	159.00	71.43	9.43

THIS NODE IS A TERMINAL NODE

CONFUSION MATRIX FOR TRAINNING SET

	32	33	Total
32	112	25	137
33	36	299	335
Total	148	324	472

MISCLASSIFICATION RATE BY CLASS

TRUE CLASS	(ERROR /SIZE)	FREQUENCY
32	(25 / 137)	18.25
33	(36 / 335)	10.75
TOTAL	(61 / 472)	12.92

CONFUSION MATRIX FOR TEST SET

	32	33	Total
32	46	22	68
33	21	138	159
Total	67	160	227

MISCLASSIFICATION RATE BY CLASS

TRUE CLASS	(ERROR /SIZE)	FREQUENCY
32	(22 / 68)	32.35
33	(21 / 159)	13.21
TOTAL	(43 / 227)	18.94

NAME OF INTERNAL TREE FILE : C:\SODAS\Tmp\ZHGH6G01.TREE

EDITION OF DECISION TREE

PARAMETERS :
Learning Set : 472
Number of variables : 30
Max. number of nodes: 17
Soft Assign : (0) PURE
Criterion coding : (1) GINI
Min. number of object by node : 20
Min. size of no-majority classes : 20
Min. size of descendant nodes : 20.00
Frequency of test set : 33.00

```
    + --- IF ASSERTION IS TRUE  (up)
    !
--- x [ ASSERTION ]
    !
    + --- IF ASSERTION IS FALSE (down)


                        +----  [ 32 ]32  (    49.00     19.00 )
                        !
                    !---16[ H_P07 = 111100 ]
                    !   !
                    !   +----  [ 33 ]33  (     0.00      2.00 )
                    !
                !---8[ H_CAT60 = 110 ]
                !   !
                !   +----  [ 17 ]33  (     7.00     10.00 )
                !
            !---4[ H_IPCFCAT = 11010100 ]
            !   !
            !   +----  [ 9 ]32  (    42.00      3.00 )
            !
        !---2[ HP01 = 110100 ]
        !   !
        !   !               +----  [ 40 ]32  (    13.00     12.00 )
        !   !               !
        !   !           !---20[ H_IPCFCAT = 10010000 ]
        !   !           !   !
        !   !           !   +----  [ 41 ]33  (     8.00     22.00 )
        !   !           !
        !   !       !---10[ H_R01CAT = 1101000000 ]
        !   !       !   !
        !   !       !   +----  [ 21 ]33  (     4.00     36.00 )
        !   !       !
        !   !---5[ H_P07 = 110000 ]
        !       !
        !       +----  [ 11 ]32  (     8.00      2.00 )
        !
    !---1[ HP06C = 1000 ]
        !
        +----  [ 3 ]33  (     6.00    229.00 )
```

Resultados del 5to. Grupo

```
-----------------------------------------------------
 BASE= C:\MIMI2\SODAS_G5.SDS
 Number of OS = 711
 Number of variables = 41
 METHOD=SODAS_TREE Version 1.2     INRIA 1998
-----------------------------------------------------

 Learning Set          :   480
 Number of variables :    30
 Max. number of nodes:    59
 Soft Assign           : ( 0 ) PURE
 Criterion coding      : ( 1 ) GINI
 Min. number of object by node     :       20
 Min. size of no-majority classes :       20
 Min. size of descendant nodes     :    20.00
 Frequency of test set             :    33.00

 GROUP OF PREDICATE VARIABLES :
 (   1 ) HP01                                          6 MODALITIES
 (   2 ) HP02                                          7 MODALITIES
 (   3 ) HP06A                                         2 MODALITIES
 (   4 ) HP06B                                         4 MODALITIES
 (   5 ) HP06C                                         4 MODALITIES
 (   6 ) HP06D                                         3 MODALITIES
 (   7 ) H_P07                                         6 MODALITIES
 (   8 ) H_P08                                         3 MODALITIES
 (   9 ) H_R01CAT                                     10 MODALITIES
 (  11 ) H_IPCFCAT                                     8 MODALITIES
 (  12 ) H_CAT60                                       3 MODALITIES
 (  14 ) H08                                          11 MODALITIES
 (  15 ) H12CAT                                        8 MODALITIES
 (  16 ) H13                                           2 MODALITIES
 (  17 ) H14                                           6 MODALITIES
 (  18 ) P11                                           8 MODALITIES
 (  19 ) ESTADO                                        3 MODALITIES
 (  22 ) P18B                                          5 MODALITIES
 (  24 ) 20A                                          10 MODALITIES
 (  25 ) 20B                                          11 MODALITIES
 (  26 ) 20C                                           7 MODALITIES
 (  27 ) P22MCAT                                      10 MODALITIES
 (  29 ) P24                                           6 MODALITIES
 (  30 ) P32DCAT                                       8 MODALITIES
 (  32 ) CODINGRE                                      3 MODALITIES
 (  34 ) P54                                           2 MODALITIES
 (  35 ) P55                                           3 MODALITIES
 (  36 ) P56                                           9 MODALITIES
 (  37 ) P58                                           3 MODALITIES
 (  39 ) P59                                           5 MODALITIES

 CLASSIFICATION VARIABLE :
 (  41 ) H_AGLOMERADO

 NUMBER OF A PRIORI CLASSES :   2
      ID_CLASS      NAME_CLASS
          1             32
          2             33


   CLASS   SIZE    LEARNING      TEST

      1     207       138          69
      2     504       342         162

 TOTAL     711       480         231


 ================================
 | SPLIT OF A NODE       :     1 |
 ================================

 LEARNING SET

    ======================================================
    |       | N(k/t)   |   N(k)    |  P(k/t)   |  P(t/k)  |
    ======================================================
    | 32    |   138.00 |   138.00  |    28.75  |   100.00 |
    | 33    |   342.00 |   342.00  |    71.25  |   100.00 |
    ======================================================


 TEST SET

    ======================================================
    |       | N(k/t)   |   N(k)    |  P(k/t)   |  P(t/k)  |
    ======================================================
    | 32    |    69.00 |    69.00  |    29.87  |   100.00 |
    | 33    |   162.00 |   162.00  |    70.13  |   100.00 |
    ======================================================

TREE CRITERION  0.409687

    ============================================================
    |  Ord |   variable        |    value      |    criterion |
    ============================================================
    |    1 |(   5) HP06C       | 1000          |        0.2703 |
    |    2 |(   1) HP01        | 100100        |        0.3316 |
    |    3 |(  11) H_IPCFCAT   | 11011010      |        0.3795 |
    |    4 |(   4) HP06B       | 1000          |        0.3816 |
    |    5 |(  36) P56         | 111001000     |        0.3873 |
    ============================================================


SPLITTING NODE:    1

VARIABLE      : (   5) HP06C
SPLIT         : 1000  ( 1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
        (  1) 1
```

```
MODALITIES BELONG RIGHT NODE :
         (  2) 0
         (  3) 3
         (  4) 2
CRITERION     : 0.270315
```

LEARNING SET

nod	left node 2	right node 3	Row totals 1
32	136.00	2.00	138.00
33	117.00	225.00	342.00
Tot	253.00	227.00	480.00

TEST SET

nod	left node 2	right node 3	Row totals 1
32	67.00	2.00	69.00
33	65.00	97.00	162.00
Tot	132.00	99.00	231.00

SPLIT OF A NODE : 2

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	136.00	138.00	53.75	98.55
33	117.00	342.00	46.25	34.21

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	67.00	69.00	50.76	97.10
33	65.00	162.00	49.24	40.12

```
TREE CRITERION  0.497180
```

Ord	variable	value	criterion
1	(1) HP01	100100	0.4160
2	(9) H_R01CAT	0000010000	0.4807
3	(7) H_P07	101100	0.4835
4	(11) H_IPCPCAT	10010000	0.4835
5	(2) HP02	1101100	0.4837

```
SPLITTING NODE:    2

VARIABLE      : (   1) HP01
SPLIT         : 100100  ( 1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
         (  1) 2
         (  4) 4
MODALITIES BELONG RIGHT NODE :
         (  2) 1
         (  3) 5
         (  5) 7
         (  6) 6
CRITERION     : 0.415984
```

LEARNING SET

nod	left node 4	right node 5	Row totals 2
32	98.00	38.00	136.00
33	37.00	80.00	117.00
Tot	135.00	118.00	253.00

TEST SET

nod	left node 4	right node 5	Row totals 2
32	44.00	23.00	67.00
33	20.00	45.00	65.00
Tot	64.00	68.00	132.00

SPLIT OF A NODE : 3

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	2.00	138.00	0.88	1.45
33	225.00	342.00	99.12	65.79

THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
SIZE OF THE NO-MAJORITY CLASSES 2.000000 VALUE OF STOP-SPLITTING RULE 20.000000

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	2.00	69.00	2.02	2.90
33	97.00	162.00	97.98	59.88

THIS NODE IS A TERMINAL NODE

SPLIT OF A NODE : 4

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	98.00	138.00	72.59	71.01
33	37.00	342.00	27.41	10.82

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	44.00	69.00	68.75	63.77
33	20.00	162.00	31.25	12.35

TREE CRITERION 0.397915

Ord	variable	value	criterion
1	(11) H_IPCFCAT	00000100	0.3657
2	(2) HP02	0011110	0.3729
3	(9) H_R01CAT	0000010000	0.3740
4	(35) P55	110	0.3869
5	(24) 20A	0010100000	0.3879

SPLITTING NODE: 4

VARIABLE : (11) H_IPCFCAT
SPLIT : 00000100 (1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
(6) 0-100
MODALITIES BELONG RIGHT NODE :
(1) 0
(2) 300-600
(3) 100-300
(4) 600-1000
(5) 1000-2000
(7) 2000-4000
(8) 4000-9000
CRITERION : 0.365734

LEARNING SET

nod	left node 8	right node 9	Row totals 4
32	0.00	98.00	98.00
33	4.00	33.00	37.00
Tot	4.00	131.00	135.00

TEST SET

nod	left node 8	right node 9	Row totals 4
32	1.00	43.00	44.00
33	0.00	20.00	20.00
Tot	1.00	63.00	64.00

SPLIT OF A NODE : 5

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	38.00	138.00	32.20	27.54
33	80.00	342.00	67.80	23.39

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	23.00	69.00	33.82	33.33
33	45.00	162.00	66.18	27.78

TREE CRITERION 0.436656

Ord	variable	value	criterion

1	(9) H_R01CAT	1000111000		0.4195
2	(11) H_IPCFCAT	10010000		0.4206
3	(30) P32DCAT	00100000		0.4285
4	(24) 20A	1110111000		0.4297
5	(27) P22MCAT	1111110100		0.4302

SPLITTING NODE: 5

VARIABLE : (9) H_R01CAT
SPLIT : 1000111000 (1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
(1) 5
(5) 1
(6) 6
(7) 7-8
MODALITIES BELONG RIGHT NODE :
(2) 2
(3) 4
(4) 3
(8) 9-10
(9) 11-12-13-14
(10) 14
CRITERION : 0.419503

LEARNING SET

nod	left node 10	right node 11	Row totals 5
32	6.00	32.00	38.00
33	28.00	52.00	80.00
Tot	34.00	84.00	118.00

TEST SET

nod	left node 10	right node 11	Row totals 5
32	8.00	15.00	23.00
33	20.00	25.00	45.00
Tot	28.00	40.00	68.00

SPLIT OF A NODE : 8

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	0.00	138.00	0.00	0.00
33	4.00	342.00	100.00	1.17

THIS STOP-SPLITTING RULE IS TRUE : The size of the node is too small
SIZE OF THE NODE 4.000000 VALUE OF STOP-SPLITTING RULE 20.000000
THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
SIZE OF THE NO-MAJORITY CLASSES 0.000000 VALUE OF STOP-SPLITTING RULE 20.000000

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	1.00	69.00	100.00	1.45
33	0.00	162.00	0.00	0.00

THIS NODE IS A TERMINAL NODE

SPLIT OF A NODE : 9

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	98.00	138.00	74.81	71.01
33	33.00	342.00	25.19	9.65

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	43.00	69.00	68.25	62.32
33	20.00	162.00	31.75	12.35

TREE CRITERION 0.376901

Ord	variable	value	criterion
1	(2) HP02	0010110	0.3514
2	(11) H_IPCFCAT	00101000	0.3639
3	(24) 20A	1101011100	0.3658
4	(35) P55	110	0.3661
5	(7) H_P07	111100	0.3683

SPLITTING NODE: 9

VARIABLE : (2) HP02

```
SPLIT          : 0010110  ( 1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
          ( 3) 3
          ( 5) 1
          ( 6) 7
MODALITIES BELONG RIGHT NODE :
          ( 1) 2
          ( 2) 4
          ( 4) 5
          ( 7) 6
CRITERION      : 0.351427

     LEARNING SET

    ================================================
    |       | left node  | right node | Row totals |
    |  nod  |        18  |        19  |          9 |
    ================================================
    | 32    |      39.00 |      59.00 |      98.00 |
    | 33    |      23.00 |      10.00 |      33.00 |
    ================================================
    |  Tot  |      62.00 |      69.00 |     131.00 |
    ================================================

     TEST SET

    ================================================
    |       | left node  | right node | Row totals |
    |  nod  |        18  |        19  |          9 |
    ================================================
    | 32    |      14.00 |      29.00 |      43.00 |
    | 33    |      11.00 |       9.00 |      20.00 |
    ================================================
    |  Tot  |      25.00 |      38.00 |      63.00 |
    ================================================


 ==================================
 | SPLIT OF A NODE       :    10 |
 ==================================

 LEARNING SET

    ========================================================
    |       | N(k/t)  |   N(k)    |  P(k/t)  |  P(t/k)  |
    ========================================================
    | 32    |    6.00 |  138.00 |    17.65 |     4.35 |
    | 33    |   28.00 |  342.00 |    82.35 |     8.19 |
    ========================================================

THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
     SIZE OF THE NO-MAJORITY CLASSES 6.000000  VALUE OF STOP-SPLITTING RULE 20.000000

 TEST SET

    ========================================================
    |       | N(k/t)  |   N(k)    |  P(k/t)  |  P(t/k)  |
    ========================================================
    | 32    |    8.00 |   69.00 |    28.57 |    11.59 |
    | 33    |   20.00 |  162.00 |    71.43 |    12.35 |
    ========================================================


 THIS NODE IS A TERMINAL NODE


 ==================================
 | SPLIT OF A NODE       :    11 |
 ==================================

 LEARNING SET

    ========================================================
    |       | N(k/t)  |   N(k)    |  P(k/t)  |  P(t/k)  |
    ========================================================
    | 32    |   32.00 |  138.00 |    38.10 |    23.19 |
    | 33    |   52.00 |  342.00 |    61.90 |    15.20 |
    ========================================================


 TEST SET

    ========================================================
    |       | N(k/t)  |   N(k)    |  P(k/t)  |  P(t/k)  |
    ========================================================
    | 32    |   15.00 |   69.00 |    37.50 |    21.74 |
    | 33    |   25.00 |  162.00 |    62.50 |    15.43 |
    ========================================================

TREE CRITERION  0.471655

    ============================================================
    |  Ord |   variable       |    value      |    criterion |
    ============================================================
    |    1 |(  11) H_IPCFCAT  | 10110100      |       0.4545 |
    |    2 |(  36) P56        | 100011100     |       0.4579 |
    |    3 |(  27) P22MCAT    | 11111110100   |       0.4580 |
    |    4 |(  17) H14        | 100000        |       0.4583 |
    |    5 |(   2) HP02       | 0110010       |       0.4593 |
    ============================================================


SPLITTING NODE:   11

VARIABLE       : (  11) H_IPCFCAT
SPLIT          : 10110100  ( 1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
          (  1) 0
          (  3) 100-300
          (  4) 600-1000
          (  6) 0-100
MODALITIES BELONG RIGHT NODE :
          (  2) 300-600
          (  5) 1000-2000
          (  7) 2000-4000
          (  8) 4000-9000
CRITERION      : 0.454507

     LEARNING SET

    ================================================
```

nod	left node 22	right node 23	Row totals 11
32	25.00	7.00	32.00
33	31.00	21.00	52.00
Tot	56.00	28.00	84.00

TEST SET

nod	left node 22	right node 23	Row totals 11
32	11.00	4.00	15.00
33	15.00	10.00	25.00
Tot	26.00	14.00	40.00

SPLIT OF A NODE : 18

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	39.00	138.00	62.90	28.26
33	23.00	342.00	37.10	6.73

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	14.00	69.00	56.00	20.29
33	11.00	162.00	44.00	6.79

TREE CRITERION 0.466701

Ord	variable	value	criterion
1	(11) H_IPCFCAT	01101000	0.4182
2	(24) 20A	1101011100	0.4242
3	(26) 20C	1101000	0.4349
4	(27) P22MCAT	0100001100	0.4362
5	(36) P56	111100100	0.4378

SPLITTING NODE: 18

VARIABLE : (11) H_IPCFCAT
SPLIT : 01101000 (1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
(2) 300-600
(3) 100-300
(5) 1000-2000
MODALITIES BELONG RIGHT NODE :
(1) 0
(4) 600-1000
(6) 0-100
(7) 2000-4000
(8) 4000-9000
CRITERION : 0.418216

LEARNING SET

nod	left node 36	right node 37	Row totals 18
32	24.00	15.00	39.00
33	21.00	2.00	23.00
Tot	45.00	17.00	62.00

TEST SET

nod	left node 36	right node 37	Row totals 18
32	10.00	4.00	14.00
33	9.00	2.00	11.00
Tot	19.00	6.00	25.00

SPLIT OF A NODE : 19

LEARNING SET

	N(k/t)	N(k)	P(k/t)	P(t/k)
32	59.00	138.00	85.51	42.75
33	10.00	342.00	14.49	2.92

THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
SIZE OF THE NO-MAJORITY CLASSES 10.000000 VALUE OF STOP-SPLITTING RULE 20.000000

TEST SET

	N(k/t)	N(k)	P(k/t)	P(t/k)

```
    ================================================
    | 32   |    29.00 |    69.00 |    76.32 |    42.03 |
    | 33   |     9.00 |   162.00 |    23.68 |     5.56 |
    ================================================


  THIS NODE IS A TERMINAL NODE

  ================================
  | SPLIT OF A NODE       :    22 |
  ================================

  LEARNING SET

     ================================================
     |      | N(k/t)   |   N(k)   | P(k/t)   | P(t/k)  |
     ================================================
     | 32   |    25.00 |   138.00 |    44.64 |    18.12 |
     | 33   |    31.00 |   342.00 |    55.36 |     9.06 |
     ================================================


  TEST SET

     ================================================
     |      | N(k/t)   |   N(k)   | P(k/t)   | P(t/k)  |
     ================================================
     | 32   |    11.00 |    69.00 |    42.31 |    15.94 |
     | 33   |    15.00 |   162.00 |    57.69 |     9.26 |
     ================================================

 TREE CRITERION  0.494260

     ============================================================
     | Ord |  variable       |    value      |    criterion |
     ============================================================
     |   1 |(   2) HP02      | 1010010       |       0.4672 |
     |   2 |(  15) H12CAT    | 00001010      |       0.4673 |
     |   3 |(  39) P59       | 00100         |       0.4745 |
     |   4 |(  27) P22MCAT   | 1110110100    |       0.4755 |
     |   5 |(  24) 20A       | 0111111000    |       0.4783 |
     ============================================================


 SPLITTING NODE:   22

 VARIABLE      : (   2) HP02
 SPLIT         : 1010010  ( 1=left node, 0=right node)
 MODALITIES BELONG LEFT NODE :
          (  1) 2
          (  3) 3
          (  6) 7
 MODALITIES BELONG RIGHT NODE :
          (  2) 4
          (  4) 5
          (  5) 1
          (  7) 6
 CRITERION     : 0.467246

      LEARNING SET

     ================================================
     |      | left node  | right node | Row totals |
     |  nod |        44  |       45   |       22   |
     ================================================
     | 32   |     12.00  |      13.00 |     25.00  |
     | 33   |     22.00  |       9.00 |     31.00  |
     ================================================
     |  Tot |     34.00  |      22.00 |     56.00  |
     ================================================

      TEST SET

     ================================================
     |      | left node  | right node | Row totals |
     |  nod |        44  |       45   |       22   |
     ================================================
     | 32   |      6.00  |       5.00 |     11.00  |
     | 33   |     13.00  |       2.00 |     15.00  |
     ================================================
     |  Tot |     19.00  |       7.00 |     26.00  |
     ================================================


 ================================
 | SPLIT OF A NODE       :    23 |
 ================================

 LEARNING SET

    ================================================
    |      | N(k/t)   |   N(k)   | P(k/t)   | P(t/k)  |
    ================================================
    | 32   |     7.00 |   138.00 |    25.00 |     5.07 |
    | 33   |    21.00 |   342.00 |    75.00 |     6.14 |
    ================================================

THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
     SIZE OF THE NO-MAJORITY CLASSES 7.000000  VALUE OF STOP-SPLITTING RULE 20.000000

 TEST SET

    ================================================
    |      | N(k/t)   |   N(k)   | P(k/t)   | P(t/k)  |
    ================================================
    | 32   |     4.00 |    69.00 |    28.57 |     5.80 |
    | 33   |    10.00 |   162.00 |    71.43 |     6.17 |
    ================================================


 THIS NODE IS A TERMINAL NODE

 ================================
 | SPLIT OF A NODE       :    36 |
 ================================

 LEARNING SET

    ================================================
    |      | N(k/t)   |   N(k)   | P(k/t)   | P(t/k)  |
    ================================================
```

```
| 32   |    24.00 |   138.00 |    53.33 |    17.39 |
| 33   |    21.00 |   342.00 |    46.67 |     6.14 |
```

TEST SET

```
|      | N(k/t)   |   N(k)   | P(k/t)   | P(t/k)   |
| 32   |    10.00 |    69.00 |    52.63 |    14.49 |
| 33   |     9.00 |   162.00 |    47.37 |     5.56 |
```

TREE CRITERION 0.497778

```
|  Ord |   variable       |    value     |   criterion |
|    1 |(  15) H12CAT     | 11001110     |       0.4334 |
|    2 |(  17) H14        | 011100       |       0.4336 |
|    3 |(  26) 20C        | 1101000      |       0.4392 |
|    4 |(  27) P22MCAT    | 0100000100   |       0.4427 |
|    5 |(  18) P11        | 11010000     |       0.4493 |
```

SPLITTING NODE: 36

VARIABLE : (15) H12CAT
SPLIT : 11001110 (1=left node, 0=right node)
MODALITIES BELONG LEFT NODE :
(1) 31-50
(2) 0-6
(5) 51-65
(6) 13-21
(7) 22-30
MODALITIES BELONG RIGHT NODE :
(3) 7-12
(4) 66-80
(8) 81-150
CRITERION : 0.433357

LEARNING SET

```
|      | left node  | right node | Row totals |
|  nod |         72 |       73   |       36   |
| 32   |      16.00 |       8.00 |      24.00 |
| 33   |      20.00 |       1.00 |      21.00 |
| Tot  |      36.00 |       9.00 |      45.00 |
```

TEST SET

```
|      | left node  | right node | Row totals |
|  nod |         72 |       73   |       36   |
| 32   |      10.00 |       0.00 |      10.00 |
| 33   |       8.00 |       1.00 |       9.00 |
| Tot  |      18.00 |       1.00 |      19.00 |
```

```
| SPLIT OF A NODE       :    37 |
```

LEARNING SET

```
|      | N(k/t)   |   N(k)   | P(k/t)   | P(t/k)   |
| 32   |    15.00 |   138.00 |    88.24 |    10.87 |
| 33   |     2.00 |   342.00 |    11.76 |     0.58 |
```

THIS STOP-SPLITTING RULE IS TRUE : The size of the node is too small
SIZE OF THE NODE 17.000000 VALUE OF STOP-SPLITTING RULE 20.000000
THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
SIZE OF THE NO-MAJORITY CLASSES 2.000000 VALUE OF STOP-SPLITTING RULE 20.000000

TEST SET

```
|      | N(k/t)   |   N(k)   | P(k/t)   | P(t/k)   |
| 32   |     4.00 |    69.00 |    66.67 |     5.80 |
| 33   |     2.00 |   162.00 |    33.33 |     1.23 |
```

THIS NODE IS A TERMINAL NODE

```
| SPLIT OF A NODE       :    44 |
```

LEARNING SET

```
|      | N(k/t)   |   N(k)   | P(k/t)   | P(t/k)   |
| 32   |    12.00 |   138.00 |    35.29 |     8.70 |
| 33   |    22.00 |   342.00 |    64.71 |     6.43 |
```

THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
SIZE OF THE NO-MAJORITY CLASSES 12.000000 VALUE OF STOP-SPLITTING RULE 20.000000

TEST SET

```
|      | N(k/t)   |   N(k)   | P(k/t)   | P(t/k)   |
| 32   |     6.00 |    69.00 |    31.58 |     8.70 |
| 33   |    13.00 |   162.00 |    68.42 |     8.02 |
```

```
    ==================================================

  THIS NODE IS A TERMINAL NODE

  ================================
  | SPLIT OF A NODE        :    45 |
  ================================

  LEARNING SET

     ==================================================
     |      | N(k/t)  |   N(k)   | P(k/t)  | P(t/k)  |
     ==================================================
     | 32   |   13.00 |   138.00 |   59.09 |    9.42 |
     | 33   |    9.00 |   342.00 |   40.91 |    2.63 |
     ==================================================

 THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
      SIZE OF THE NO-MAJORITY CLASSES 9.000000  VALUE OF STOP-SPLITTING RULE 20.000000

  TEST SET

     ==================================================
     |      | N(k/t)  |   N(k)   | P(k/t)  | P(t/k)  |
     ==================================================
     | 32   |    5.00 |    69.00 |   71.43 |    7.25 |
     | 33   |    2.00 |   162.00 |   28.57 |    1.23 |
     ==================================================


  THIS NODE IS A TERMINAL NODE

  ================================
  | SPLIT OF A NODE        :    72 |
  ================================

  LEARNING SET

     ==================================================
     |      | N(k/t)  |   N(k)   | P(k/t)  | P(t/k)  |
     ==================================================
     | 32   |   16.00 |   138.00 |   44.44 |   11.59 |
     | 33   |   20.00 |   342.00 |   55.56 |    5.85 |
     ==================================================

 THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
      SIZE OF THE NO-MAJORITY CLASSES 16.000000  VALUE OF STOP-SPLITTING RULE 20.000000

  TEST SET

     ==================================================
     |      | N(k/t)  |   N(k)   | P(k/t)  | P(t/k)  |
     ==================================================
     | 32   |   10.00 |    69.00 |   55.56 |   14.49 |
     | 33   |    8.00 |   162.00 |   44.44 |    4.94 |
     ==================================================


  THIS NODE IS A TERMINAL NODE

  ================================
  | SPLIT OF A NODE        :    73 |
  ================================

  LEARNING SET

     ==================================================
     |      | N(k/t)  |   N(k)   | P(k/t)  | P(t/k)  |
     ==================================================
     | 32   |    8.00 |   138.00 |   88.89 |    5.80 |
     | 33   |    1.00 |   342.00 |   11.11 |    0.29 |
     ==================================================

 THIS STOP-SPLITTING RULE IS TRUE : The size of the node is too small
      SIZE OF THE NODE 9.000000  VALUE OF STOP-SPLITTING RULE 20.000000
 THIS STOP-SPLITTING RULE IS TRUE : The size of the no-majority classes is too small
      SIZE OF THE NO-MAJORITY CLASSES 1.000000  VALUE OF STOP-SPLITTING RULE 20.000000

  TEST SET

     ==================================================
     |      | N(k/t)  |   N(k)   | P(k/t)  | P(t/k)  |
     ==================================================,
     | 32   |    0.00 |    69.00 |    0.00 |    0.00 |
     | 33   |    1.00 |   162.00 |  100.00 |    0.62 |
     ==================================================


  THIS NODE IS A TERMINAL NODE

     CONFUSION MATRIX FOR TRAINNING SET

     ============================================
     |      | 32       | 33       | Total     |
     ============================================
     | 32   |       95 |       43 |       138 |
     | 33   |       22 |      320 |       342 |
     ============================================
     | Total|      117 |      363 |       480 |
     ============================================

 MISCLASSIFICATION RATE BY CLASS

     TRUE CLASS            (  ERROR /SIZE    )   FREQUENCY
     32                    (     43 /    138 )      31.16
     33                    (     22 /    342 )       6.43

     TOTAL                 (     65 /    480 )      13.54


     CONFUSION MATRIX FOR TEST SET

     ============================================
     |      | 32       | 33       | Total     |
     ============================================
     | 32   |       38 |       31 |        69 |
     | 33   |       14 |      148 |       162 |
     ============================================
     | Total|       52 |      179 |       231 |
     ============================================

 MISCLASSIFICATION RATE BY CLASS
```

```
     TRUE CLASS            (  ERROR /SIZE     )    FREQUENCY
     32                    (     31 /     69 )       44.93
     33                    (     14 /    162 )        8.64

     TOTAL                 (     45 /    231 )       19.48


 NAME OF INTERNAL TREE FILE : C:\SODAS\Tmp\ZHGJUG01.TREE


    ==================================
   | EDITION OF DECISION TREE         |
    ==================================

 PARAMETERS :
 Learning Set          :   480
 Number of variables :    30
 Max. number of nodes:    19
 Soft Assign           : ( 0 ) PURE
 Criterion coding      : ( 1 ) GINI
 Min. number of object by node     :       20
 Min. size of no-majority classes :       20
 Min. size of descendant nodes     :    20.00
 Frequency of test set             :    33.00

    + --- IF ASSERTION IS TRUE  (up)
    !
--- x [ ASSERTION ]
    !
    + --- IF ASSERTION IS FALSE (down)


               +----  [ 8 ]33  (     0.00      4.00 )
               !
          !----4[ H_IPCFCAT = 00000100 ]
          !    !
          !    !               +----  [ 72 ]33  (    16.00     20.00 )
          !    !               !
          !    !          !---36[ H12CAT = 11001110 ]
          !    !          !    !
          !    !          !    +----  [ 73 ]32  (     8.00      1.00 )
          !    !          !
          !    !     !---18[ H_IPCFCAT = 01101000 ]
          !    !     !    !
          !    !     !    +----  [ 37 ]32  (    15.00      2.00 )
          !    !     !
          !    !----9[ HP02 = 0010110 ]
          !         !
          !         +----  [ 19 ]32  (    59.00     10.00 )
          !
     !----2[ HP01 = 100100 ]
     !    !
     !    !    +----  [ 10 ]33  (     6.00     28.00 )
     !    !    !
     !    !----5[ H_R01CAT = 1000111000 ]
     !         !
     !         !          +----  [ 44 ]33  (    12.00     22.00 )
     !         !          !
     !         !     !---22[ HP02 = 1010010 ]
     !         !     !    !
     !         !     !    +----  [ 45 ]32  (    13.00      9.00 )
     !         !     !
     !         !---11[ H_IPCFCAT = 10110100 ]
     !              !
     !              +----  [ 23 ]33  (     7.00     21.00 )
     !
!----1[ HP06C = 1000 ]
     !
     +----  [ 3 ]33  (     2.00    225.00 )
```