UNIDAD 3. Clasificador Basado en Reglas: Chaid, Naive Bayes (Clasificador Bayesiano).

CHAID, (CHi-squared Automated Interaction Detection)

CHAID (Gordon V. Kass, 1980) es un método de clasificación jerárquica, utilizado cuando se dispone de una muestra de elementos cuya clase es conocida, y un conjunto de predictores cualitativos o factores, algunos de los cuales están relacionados con la clasificación y podrían ser utilizados para su predicción. No se requiere ningún supuesto restrictivo (como Normalidad de variables o de residuos), y es muy sencillo en su aplicación e interpretación. Se suele emplear para segmentación de mercados, con el fin de definir las características (grupo de edad, lugar de residencia, sexo, situación familiar, nivel de estudios, etc.) que definen a los compradores o usuarios (en este ejemplo la clasificación es binaria: comprador/no comprador), y orientar así el diseño de productos y las acciones publicitarias.

El algoritmo construye un árbol de clasificación y busca entre los factores o predictores los que están más relacionados con el criterio de clasificación, utilizando para ello la prueba Ji cuadrado de independencia: para cada factor se construye la tabla bidimensional de frecuencias o tabla de contingencia (factor x clase) y se aplica el contraste Ji cuadrado, seleccionando el factor que tiene el valor p más pequeño en esa prueba, es decir el factor más significativo. A continuación la muestra es dividida de acuerdo con los niveles del factor elegido, y a cada grupo resultante se vuelve a aplicar el mismo criterio, dividiendo nuevamente de forma reiterada hasta que no es posible continuar dividiendo o no se encuentra ningún otro factor significativo.

La prueba Ji cuadrado de independencia utiliza el supuesto (instrumental) de que el factor es independiente del criterio de clasificación y calcula el valor del estadístico de contraste mediante la suma de los cuadrados de las diferencias entre cada frecuencia observada O_{ij} y la correspondiente frecuencia esperada E_{ij} (en el supuesto de independencia la frecuencia esperada es el producto de las frecuencias marginales) dividido cada cuadrado por la frecuencia esperada. La distribución del estadístico es, en condiciones bastante generales, Ji cuadrado.

$$X^2 = \sum_i \sum_j (O_{ij} - E_{ij})^2 / E_{ij}$$

El valor del estadístico será más grande cuanto mayor sea la discrepancia entre las frecuencias observadas y esperadas. Con el valor del estadístico se obtiene el valor p, o probabilidad de que siendo independientes ambas variables el valor del estadístico sea mayor o igual que el valor realmente observado (cola derecha en la distribución). Un valor p muy pequeño (típicamente < 0,05, nivel de significación habitual) hace improbable la muestra observada, y por lo tanto permite descartar razonablemente el supuesto de independencia. Se asume por lo tanto que los factores más relacionados con la clasificación son los que tienen los valores p más pequeños, y esos factores permitirán predecir razonablemente la clase de pertenencia para cada caso.

En cada paso del análisis CHAID se divide el árbol en la variable del predictor que tenga el valor de probabilidad o p-valor más bajo, siempre y cuando el valor p sea menor que el nivel de significación ($\alpha = 0,05$).

Las variables explicativas deben ser categóricas u ordinales. Cuando no lo son, es decir cuando son variables cuantitativas, deben ser categorizadas previamente formando intervalos.

CHAID considera todos los cortes posibles en todas las variables, y selecciona el corte que genera el p-valor más pequeño. La búsqueda de la variable y el corte óptimo se llevan a cabo en dos fases: fusión de categorías (merge) y selección de la variable de corte (split).

Merge, fusión de categorías, agrupa los niveles o valores de las variables explicativas. Para cada variable, se agrupan los estados que no sean significativamente diferentes; cada nivel de un factor nominal puede ser agregado a otro, mientras que en las variables ordinales un valor de la variable sólo puede ser agregado a otro si es contiguo en la escala. Se forman todos los pares posibles de categorías, y para cada par se calcula el estadístico JI^2 correspondiente a su cruce con la variable dependiente que indica la clase de pertenencia; el par con valor más bajo del estadístico, siempre que no sea significativo, formará una nueva categoría con los dos valores fusionados. La condición de que no sea significativo asegura que se están agrupando dos niveles similares, dos niveles que no son significativamente distintos. El proceso continúa hasta que no pueden realizarse más fusiones porque los JI^2 muestran resultados significativos.

Split, o selección de la variable de corte: de la fase anterior se toma la agrupación en la variable con contraste más significativo (menor p-valor ajustado). Si es inferior a un mínimo α-split prefijado, se toma dicha agrupación como partición del nodo. Los valores p son previamente corregidos para evitar el problema de las comparaciones múltiples: cuando se hacen muchas comparaciones simultáneas, y utilizando el nivel de significación habitual 0,05, encontraremos entre ellas un 5% de valores p significativos aunque ambas variables (el criterio de clasificación y el factor) sean independientes, es decir falsos positivos, criterios espureos o irrelevantes que no nos ayudarán a predecir la clase correcta, pero que no podremos distinguir de los factores relevantes. Para evitar este efecto debido a las comparaciones múltiples se utilizan distintas correcciones o ajustes, de los cuales el más sencillo es el de Bonferroni, que consiste en dividir el nivel de significación por el número de comparaciones (en lugar de 0,05, si hacemos 20 comparaciones utilizaremos como nivel de significación 0,05/20 = 0,0025)

En esencia, el método CHAID busca entre los factores aquellos que están más relacionados con la clasificación, y los utiliza secuencialmente para dividir la muestra en partes, mediante reglas claras y simples, que permiten establecer las características que definen a los elementos de las distintas clases.

Utilizaremos para la aplicación la función "chaid" del paquete CHAID de R. Ese paquete no está disponible en los repositorios habituales, y debe ser instalado (solamente la primera vez) desde una localización específica, ejecutando la orden siguiente:

```
install.packages("CHAID", repos="http://R-Forge.R-project.org")      # instala CHAID
```

Con la instalación de CHAID deben instalarse automáticamente otros paquetes que CHAID necesita (dependencias). Si no es así R indicará con un mensaje de error el nombre del paquete faltante, que debemos instalar del modo habitual. Suele faltar en la instalación el paquete partykit, que CHAID necesita; en ese caso debemos instalar partykit y a continuación instalar de nuevo CHAID.

La sintaxis de la función chaid es la siguiente, en la que se muestran los principales argumentos:

chaid(formula, data)

El único argumento obligatorio, "formula", expresa la variable dependiente como función de las variables explicativas o predictores; puede indicarse con el argumento "data", opcional, el conjunto al que pertenecen las variables.

Se pueden configurar previamente los distintos elementos de funcionamiento del algoritmo con la función chaid_control, aunque no suele ser necesario, ya que los valores por defecto –que se muestran a continuación- son adecuados para la mayoría de las aplicaciones:

chaid_control(alpha2 = 0.05, alpha3 = -1, alpha4 = 0.05,minsplit = 20, minbucket = 7, minprob = 0.01,stump = FALSE, maxheight = -1)

alpha2: nivel de significación para fundir dos categorías de un predictor.

alpha3: si se utiliza un valor positivo, es el nivel de significación utilizado para la división de categorías con tres o más niveles previamente formadas dentro de un predictor. Por defecto no se realiza (valor negativo).

alpha4: nivel de significación utilizado para la división de un nodo mediante el predictor más significativo.

Minsplit: número de observaciones con el cual no se desea proseguir con la división ulterior.

Minbucket: número mínimo de observaciones en los nodos terminales.

Minprob: mínima frecuencia de observaciones en los nodos terminales.

Stump: si es TRUE solo se divide el nodo raíz.

Maxheight: máxima altura del árbol.

La función chaid solamente acepta predictores categóricos (nominales u ordinales). Si alguno es cuantitativo, debe ser transformado previamente.

Esta función aplica el siguiente algoritmo:

1. Si la variable X solo tiene una categoría, se asigna un valor p = 1
2. Si X tiene 2 niveles, ir al paso 8.

3. En otro caso, encuentra el par de categorías menos significativo, aquel que tiene un valor p más alto con respecto a la variable de clasificación o variable dependiente (es decir busca los dos niveles más similares).

4. Para el par encontrado se comprueba si su valor p es mayor que un nivel prefijado alpha2. Si lo es, ambos niveles se funden en una única categoría compuesta, y se corrige el conjunto de niveles de X, pasando a continuación al paso 7.

5. (Opcional) Si la categoría compuesta recién creada consiste en tres o más niveles originales, se encuentra la mejor división dentro de ella, con el valor p más pequeño, siempre que no sea mayor que un nivel prefijado alpha3.

6. Ir al paso 2.

7. (Opcional) Cualquier categoría con pocas observaciones (comparando con un tamaño de segmento prefijado), es unida a la categoría más similar, aquella con mayor valor p.

8. Se calcula el valor p ajustado para las categorías existentes, aplicando la corrección de Bonferroni.

A continuación se busca la mejor división para todos los predictores, seleccionando aquel con menor valor p ajustado (es decir el más significativo). La división solo se realiza si este valor p es menor o igual que un valor prefijado alpha4; en otro caso no se divide y el nodo se considera terminal.

La construcción del árbol se detiene en los siguientes casos: a) si el nodo es puro (todos los casos pertenecen a la misma clase) b) Si todos los casos en el nodo tienen los mismos valores para cada predictor; c) si se ha alcanzado la profundidad máxima prefijada del árbol; d) si el tamaño del nodo es menor que un valor prefijado; e) si la división del nodo produce un nodo hijo cuyo tamaño es menor que un valor prefijado el nodo hijo será fundido con el más similar (mayor valor p), y si el número de nodos hijo resultantes es 1, el nodo no es dividido.

Utilizaremos para el ejemplo con R el conjunto de datos "compra", que leeremos desde un archivo csv, en formato de texto separado por ";". Se trata de un formato estándar, que se puede obtener desde fuentes muy diversas, como por ejemplo desde un archivo Excel (guardado como csv).

Los datos están en el archivo "compra.csv", y consisten en una muestra de 1000 consumidores elegidos al azar de los cuales algunos han comprado un determinado producto (comprador = si), con variables adicionales como sexo, edad, empleo, formación, o estado civil. El objetivo es establecer las características del comprador, de forma que se pueda orientar más adecuadamente la fabricación, distribución, y promoción del producto.

```
setwd("C:/CURSO DM")
compra  <- read.csv2("compra.csv",header=TRUE,encoding="latin1")
```

Una vez construido el data frame o conjunto de datos "compra" podemos ver cuales son las variables que lo forman y elaborar algunos estadísticos descriptivos elementales que nos permitan conocer mejor los datos:

```
names(compra)
summary(compra)
```

```
> names(compra)
[1] "comprador" "sexo" "edad" "empleo" "formacion"  "estadocivil"

> summary(compra)
comprador     sexo       edad         empleo       formacion     estadocivil
no:524    hombre:457  < 35  :196   inactivo:205   elemen :287   casado   :561
si:476    mujer :543  > 55  :353   ocupado :636   media  :582   no casado:402
                      35 -55:450   parado  :159   univers:128   NA's     : 37
                      NA's  :  1                  NA's   :  3
```

Vemos que todas las variables son cualitativas, y pueden ser utilizadas directamente con esta técnica. Si alguna variable explicativa de interés fuese numérica debería ser transformada –recodificada- convenientemente antes de aplicar el método.

Recordemos que el algoritmo CHAID trata de forma diferente a los factores nominales y ordinales. Dos de las variables, edad y formación, son ordinales, y deben ser declaradas como tales, indicando además el orden correcto de los niveles que deberá sustituir al orden alfabético (reconocible en las tablas del resumen anterior) que R utiliza por defecto para los factores, antes de aplicar el método. Es necesario declarar la variable ordinal como tal aunque el orden alfabético coincida con el correcto:

```
compra$edad <- ordered(compra$edad, levels=c('< 35','35 -55','> 55'))
compra$formacion <- ordered(compra$formacion, levels=c('elemen','media','univers'))
```

Si ejecutamos de nuevo summary(compra) observaremos que los niveles aparecen correctamente ordenados.

Ahora aplicamos el método. Cargamos el paquete CHAID, previamente instalado.

```
library(CHAID)      # carga CHAID
```

Establecemos (es opcional) algunos elementos de configuración:

```
chaid_control(minsplit = 50, minprob = 0.10) # configura algunas opciones de chaid
        # minsplit: número de casos por debajo del cual no se hace la partición
        # minprob: frecuencia mínima de los nodos terminales.

ch <- chaid(comprador ~ sexo+edad+empleo+formacion+estadocivil, data = compra)
        # o bien: ch <- chaid(comprador ~ ., data = compra)

print(ch)     # imprime los resultados de chaid
plot(ch)      # representa gráficamente el árbol de segmentación
```

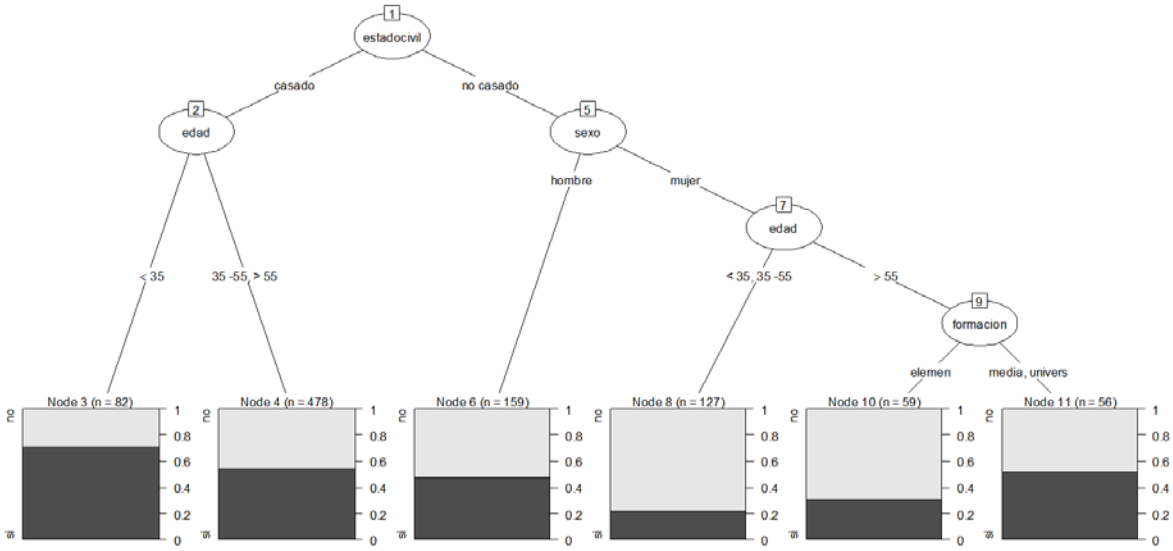
```
> print(ch)      # imprime los resultados de chaid

Model formula:
comprador ~ sexo + edad + empleo + formacion + estadocivil

Fitted party:
[1] root
|   [2] estadocivil in casado
|   |   [3] edad < 35: si (n = 82, err = 29.3%)
|   |   [4] edad in 35 -55, > 55: si (n = 478, err = 46.2%)
|   [5] estadocivil in no casado
|   |   [6] sexo in hombre: no (n = 159, err = 47.8%)
|   |   [7] sexo in mujer
|   |   |   [8] edad < 35, 35 -55: no (n = 127, err = 22.0%)
|   |   |   [9] edad > 55
|   |   |   |   [10] formacion in elemen: no (n = 59, err = 30.5%)
|   |   |   |   [11] formacion in media, univers: si (n = 56, err = 48.2%)

Number of inner nodes:    5
Number of terminal nodes: 6

plot(ch)      # representa gráficamente el árbol de segmentación
```



Observamos en el gráfico los distintos criterios sucesivos de división: la variable más significativa en relación con la característica de comprador es el estado civil, por lo que se forman los dos grandes grupos de casados y no casados. En cada uno de ellos a su vez se utilizan distintos criterios para construir el árbol, utilizando las variables edad, sexo, o formación (la variable empleo no parece importante, no se utiliza).

Si observamos las cajas en las hojas o nodos terminales, vemos que la mayor proporción de compradores corresponde al grupo de casados menores de 35 años, dentro del cual hay aproximadamente un 70% de compradores (los resultados que acompañan al gráfico permiten precisar más: 100 – 29,3 = 70,7%). En el otro extremo vemos que en el grupo de no casado/mujer/ menor de 55 años el porcentaje de compradores desciende casi hasta el 20% (22% si vemos los resultados detallados).

Chaid nos ayuda por lo tanto a definir razonablemente los perfiles de compradores y no compradores.

Naive Bayes (Clasificador bayesiano ingenuo)

Naive Bayes (Maron, 1961) es un método de clasificación basado en el teorema de Bayes (teoría elemental de probabilidad) y algunas hipótesis simplificadoras adicionales, fundamentalmente el supuesto de independencia entre las variables predictoras, es decir que las diferentes características consideradas como variables explicativas no están relacionadas entre sí. Este supuesto es más que discutible en la mayoría de las aplicaciones prácticas, razón por la que recibe el apelativo de ingenuo (naive). Utiliza el criterio de máxima verosimilitud y el supuesto adicional de distribución conocida de los predictores. Sin embargo no se suelen comprobar los supuestos, y el método se considera en general correctamente aplicado si los resultados obtenidos son buenos y se pueden validar con una muestra de prueba.

El clasificador Naive Bayes se puede entrenar de forma eficiente en un entorno de aprendizaje supervisado. En general solo se requiere una pequeña cantidad de datos de entrenamiento para estimar los parámetros necesarios (medias y varianzas de las variables) para la clasificación, y el supuesto de independencia hace innecesario determinar la matriz completa de covarianzas. Pueden utilizarse como variables explicativas indistintamente variables cuantitativas y cualitativas.

La probabilidad de que un caso pertenezca a una clase determinada C se puede expresar como la probabilidad condicionada por los valores de las m variables explicativas:

$P(C / X_1,X_2, \ldots,X_m)$

Utilizando el teorema de Bayes se puede expresar esta probabilidad (a posteriori, conocidos los valores de las variables explicativas) en términos de las probabilidades a priori:

$$P(C / X_1,X_2, \ldots,X_m) = P(C)\, P(X_1,X_2, \ldots,X_m /C) / P(X_1,X_2, \ldots,X_m)$$

Donde P(C) es la probabilidad a priori o incondicionada de pertenecer a la clase C. Solo importa el numerador en la expresión anterior, ya que el denominador es el mismo para todas las clases, y ese numerador puede expresarse, mediante la regla del producto:

$$P(C)\, P(X_1,X_2, \ldots,X_m /C) = P(C)\, P(X_1 /C)\, P(X_2/C, X_1)\, P(X_3/C, X_1,X_2) \ldots$$

y asumiendo que las X_i son independientes:

$$P(C)\, P(X_1,X_2, \ldots,X_m /C) = P(C)\, P(X_1 /C)\, P(X_2/C)\, P(X_3/C) \ldots = P(C) \prod_i P(X_i /C)$$

Las probabilidades se calculan con los estimadores de máxima verosimilitud -las frecuencias relativas- de los datos de entrenamiento. Una vez estimados los parámetros, se calculan las probabilidades a posteriori de pertenencia a las distintas clases, y cada caso es finalmente asignado a la clase más probable.

Si alguna de las probabilidades en el producto definido anteriormente es cero, porque no coinciden en la muestra la clase y algún valor del predictor, todo el producto será cero; para evitar este efecto se corrigen los valores nulos mediante un valor umbral (por ejemplo 0,001).

Utilizaremos para la aplicación de este método la función naiveBayes del paquete e1071 de R. Debe indicarse el conjunto de predictores "x" y la variable de clasificación "y", o de forma alternativa una fórmula:

naiveBayes(x, y)

o bien:

naiveBayes(formula)

Se supone distribución Normal para los predictores cuantitativos.

La función devuelve la distribución de clases para la variable dependiente, así como las probabilidades condicionadas –por las distintas clases- para cada predictor, y para las variables numéricas la media y desviación típica estimadas.

Utilizaremos en el ejemplo de aplicación con R el conjunto de datos "pacientes", en el archivo pacientes.RData. Se trata de una muestra de 881 pacientes de un centro de atención primaria, con 27 variables: número de orden, edad, sexo, consumidor de alcohol, fumador, dieta, peso, talla, tensión arterial,… Algunas variables como el índice de masa corporal o el índice cintura-cadera, se han calculado a partir de otras del mismo conjunto (talla y peso, perímetro de cintura y perímetro de cadera).

Leemos los datos, obteniendo el conjunto o data frame pacientes.

```
setwd("C:/CURSO DM")
load("pacientes.RData")

names(pacientes)
summary(pacientes)
```

```
> names(pacientes)
"num"  "edad"  "sexo"  "alcohol"  "tabaco"  "dieta"  "peso"  "talla"  "tad"  "tas"
"colesterol"  "pericintura"  "peripelvis"  "glucemia"  "hba1c"  "urea"  "trigl"
"creat"  "urico"  "alcoholgrdia"  "imc"  "imc2"  "icc"  "icc2"  "hta2"  "glucemia2"
"glucemia3"
```

```
> summary(pacientes)
      num           edad          sexo     alcohol  tabaco   dieta        peso
 Min.   :  1   Min.   :30.00   mujer:519   no:374   no:750   0:194   Min.   : 35.00
 1st Qu.:221   1st Qu.:51.00   varón:362   si:507   si:131   1:687   1st Qu.: 66.00
 Median :441   Median :58.00                                        Median : 74.00
 Mean   :441   Mean   :58.69                                        Mean   : 74.98
 3rd Qu.:661   3rd Qu.:66.00                                        3rd Qu.: 83.00
 Max.   :881   Max.   :91.00                                        Max.   :134.00


     talla            tad              tas          colesterol   pericintura
 Min.   :1.150   Min.   : 50.00   Min.   : 96.0   Min.   : 28   Min.   : 54.00
 1st Qu.:1.520   1st Qu.: 78.00   1st Qu.:130.0   1st Qu.:197   1st Qu.: 90.00
 Median :1.570   Median : 80.00   Median :142.0   Median :224   Median : 97.00
 Mean   :1.578   Mean   : 81.94   Mean   :145.1   Mean   :227   Mean   : 97.92
 3rd Qu.:1.640   3rd Qu.: 90.00   3rd Qu.:160.0   3rd Qu.:256   3rd Qu.:105.00
 Max.   :1.850   Max.   :120.00   Max.   :210.0   Max.   :479   Max.   :154.00
 NA's   :5       NA's   :10       NA's   :10      NA's   :8     NA's   :71

   peripelvis       glucemia       hba1c            urea            trigl
 Min.   : 50.0   Min.   : 47   Min.   : 3.400   Min.   :  4.00   Min.   :  30.0
 1st Qu.:100.0   1st Qu.:130   1st Qu.: 5.300   1st Qu.: 36.00   1st Qu.:  83.0
 Median :102.0   Median :147   Median : 6.000   Median : 43.00   Median : 114.0
 Mean   :104.3   Mean   :161   Mean   : 6.487   Mean   : 45.74   Mean   : 136.4
 3rd Qu.:109.0   3rd Qu.:180   3rd Qu.: 7.200   3rd Qu.: 50.00   3rd Qu.: 164.5
 Max.   :165.0   Max.   :402   Max.   :15.700   Max.   :390.00   Max.   :1163.0
 NA's   :71      NA's   :4     NA's   :69       NA's   :396      NA's   :14

    creat          urico         alcoholgrdia        imc              imc2
 Min.   :0.4000  Min.   : 1.700  Min.   :  0.00  Min.   :16.65 inferior a normal:  2
 1st Qu.:0.8000  1st Qu.: 3.800  1st Qu.:  0.00  1st Qu.:26.84 normal           :107
 Median :0.9000  Median : 4.800  Median :  0.00  Median :29.75 sobrepeso        :350
 Mean   :0.9176  Mean   : 6.288  Mean   : 12.54  Mean   :30.10 obesidad leve    :282
 3rd Qu.:1.0000  3rd Qu.: 5.800  3rd Qu.: 16.00  3rd Qu.:32.87 obesidad media   :102
 Max.   :4.6000  Max.   :97.000  Max.   :240.00  Max.   :51.42 obesidad mórbida : 33
 NA's   :53      NA's   :378                      NA's   :5     NA's            :  5

     icc            icc2     hta2       glucemia2     glucemia3
 Min.   :0.5000  alto :609  no:487  Min.   : 45.0   Min.   : 44.0
 1st Qu.:0.8788  medio:196  si:394  1st Qu.:128.0   1st Qu.:128.0
 Median :0.9307  bajo :  5          Median :145.0   Median :145.0
 Mean   :0.9405  NA's : 71          Mean   :159.1   Mean   :159.1
 3rd Qu.:0.9906                     3rd Qu.:179.0   3rd Qu.:180.0
 Max.   :1.5000                     Max.   :399.0   Max.   :398.0
 NA's   :71                         NA's   :4       NA's   :4
```

La variable hba1c, hemoglobina glicoxilada, es utilizada habitualmente para el diagnóstico y control de diabetes. Crearemos una variable nueva dicotómica a la que llamamos hba1c7, recodificando la anterior, para identificar los valores mayores que 7.

Para esa recodificación podemos utilizar el menú de R Commander Datos → modificar variables → recodificar variables, escribiendo en el cuadro de diálogo las condiciones de recodificación, o bien construir o copiar la orden correspondiente en la ventana de órdenes ejecutándola a continuación:

```
pacientes$hba1c7 <- recode(pacientes$hba1c, ' lo:7="hasta 7"; 7:hi="mayor que 7" ', as.factor=T)
```

Indicamos en la función recode la variable a recodificar, hba1c del conjunto pacientes, y los intervalos que se deben formar, junto con las etiquetas o niveles del factor resultante; "lo" identifica el valor más bajo de la variable y "hi" el más alto, por lo que indicamos que todos los valores entre el mínimo y 7 (incluido éste) pasarán a denominarse "hasta 7" y los que van de 7 al máximo "mayor que 7".

Construimos una tabla de frecuencias de la nueva variable:

table(pacientes$hba1c7)

```
    hasta 7     mayor que 7
        590             222
```

Ahora aplicaremos el método Naive Bayes para intentar relacionar esta variable que tiene dos niveles con otras variables del conjunto, utilizadas como predictoras.

Naturalmente no utilizaremos como predictor la variable hba1c empleada para construir la variable objetivo hba1c7. Tambien prescindiremos de otras variables asociadas directamente a hba1c (como glucemia, glucemia2, glucemia3, medidas en períodos diferentes), y de variables sin interés como el número de caso. Comenzaremos simplificando el conjunto de datos, manteniendo solamente las variables a utilizar; creamos para eso un nuevo conjunto al que llamaremos "datos".

datos = pacientes[, c('edad', 'sexo', 'alcohol', 'tabaco', 'dieta', 'peso', 'talla', 'tad', 'tas', 'colesterol', 'pericintura', 'peripelvis', 'trigl', 'creat', 'alcoholgrdia','imc', 'icc', 'hta2', 'hba1c7')]

La expresión pacientes[i,j] identifica al valor de la fila o caso i y la columna o variable j; si ponemos pacientes[i,] sin nada a la derecha de la coma, se trata del caso i con todas las variables, y pacientes[,j] , sin nada a la izquierda de la coma, se refiere a todos los casos de la variable j. La expresión anterior que define el conjunto "datos" contiene todos los casos (no hay nada a la izquierda de la coma) y todas las variables relacionadas mediante la función de concatenación "c".

Construimos en primer lugar los conjuntos de entrenamiento y prueba, dividiendo la muestra completa en dos partes, la primera de ellas con los 600 primeros casos y la de prueba con los restantes (en una aplicación real con datos diferentes sería preferible una muestra aleatoria construida como se explica en la unidad anterior).

entrenamiento <- subset(datos[1:600,], select = -hba1c7)
prueba <- subset(datos[601:881,], select = -hba1c7)

En ambos conjuntos hemos suprimido la variable hba1c7 con la función subset.

Ahora construimos el modelo:

library(e1071) # cargamos el paquete e1071 (que debe ser instalado previamente)
modelo <- naiveBayes(x = entrenamiento, y = datos[1:600,"hba1c7"])

El objeto construido "modelo" contiene los resultados: la distribución de probabilidades a priori de las dos clases y –para cada predictor- las probabilidades condicionadas por cada clase (si el predictor es cualitativo) o la media y desviación típica (si es una variable numérica). Si ejecutamos su nombre podremos observar esos resultados (se muestra solamente el inicio, a título de ejemplo):

```
> modelo

A-priori probabilities:
datos[muestra, "hba1c7"]
    hasta 7 mayor que 7
  0.7472727   0.2527273
```

Globalmente, el grupo de hba1c7 "hasta 7" tiene el 74,7% de los casos, y "mayor que 7" el 25,3% restante.

```
Conditional probabilities:
                       edad
datos[1:600, "hba1c7"]     [,1]     [,2]
           hasta 7     59.77616 11.45739
           mayor que 7 55.82014 11.43808

                       sexo
datos[1:600, "hba1c7"]     mujer     varón
           hasta 7     0.5523114 0.4476886
           mayor que 7 0.6115108 0.3884892
……….        ……….        ……….     ……….
```

La edad es una variable numérica. En el grupo "hasta 7" la media de edades es 59,776 años y la desviación típica 11,457 años. Para la variable sexo, cualitativa, se muestra la probabilidad (frecuencia) de hombres y mujeres en cada grupo.

Establecemos la predicción, con los datos del conjunto de validación, mediante la función predict, creando el objeto "resultados", y a continuación cruzamos la variable de predicción con la clase original para conocer el grado de acierto:

```
resultados <- predict(object = modelo, newdata = prueba, type = "class")
t <- table(resultados, pacientes[601:881,"hba1c7"])
t ;  100 * sum(diag(t)) / sum(t)
```

```
      resultados      hasta 7    mayor que 7
        hasta 7         162          49
        mayor que 7      17          34

[1] 74.80916
```

Los resultados son solamente discretos: las variables explicativas permiten predecir la clase correcta solamente en el 74,8% de los casos.

Supongamos que queremos predecir el nivel menor o mayor que 7 de hba1c de un paciente nuevo, del cual tenemos los datos de todas las variables explicativas:

Edad: 62; sexo: mujer; alcohol: si; tabaco: no; dieta: 0; peso: 92; talla: 1.62; tad: 100; tas: 160; colesterol: 263; pericintura: 104; peripelvis: 112; trigl: 116; creat: 0.9; alcoholgrdia: 8; imc: 35.05563; icc: 0.9285714; hta2: si

Añadimos un caso nuevo al conjunto "datos" (caso 882) incorporando esos valores (en el orden exacto en que están en el conjunto; los valores de texto entre comillas; la variable hba1c7 desconocida como NA):

```
datos[882,] <- c(62, "mujer", "si", "no", 0, 92, 1.62, 100, 160, 263, 104, 112, 116, 0.9, 8,
                 35.05563, 0.9285714, "si", NA)
```

y a continuación utilizamos la función predict, con el modelo construido, aplicado a ese caso:

```
predict(modelo, datos[882,], type = "class")
```

```
> predict(modelo, datos[882,], type = "class")
[1] hasta 7
Levels: hasta 7 mayor que 7
```

La predicción es: "hasta 7".

Con el argumento type = "raw" podemos observar la probabilidad de asignación a cada una de las dos clases, y medir de este modo la fiabilidad de la predicción:

```
predict(modelo, datos[882,], type = "raw")
```

```
      hasta 7   mayor que 7
[1,] 0.7080796   0.2919204
```

La probabilidad de que pertenezca a la clase "hasta 7" es 0,708