UADY
UNIVERSIDAD AUTÓNOMA DE YUCATÁN
"Luz, Ciencia y Verdad"

FACULTAD DE MATEMÁTICAS

Minería de Datos

Unidad III: Métodos de Clasificación

ADA 10: *Clasificación basada en reglas.*

Licenciatura en Actuaria.

Integrantes:
- Álvarez Herrera Samantha
- Ciau Puga Abigail
- Colonia Espinosa Cindy
- Fernández Caro Frida
- Padilla Jiménez Meybor
- Sobrino Bermejo Samantha

M.C. Ernesto Guerrero Lara

Fecha de entrega: Miercoles 13 de mayo de 2020

Los datos se encuentran en la base ClientesSegurosADA.

La base cuenta con 718 observaciones y 10 variables, pero solo trabajaremos con algunas de ellas.

De las 10 variables solo usaremos 2 como predictoras: "marital.stat" que es status marital y "sex" que es el sexo de la persona, la variable que deseamos clasificar es "Health.ins" que indica si tienen o no un seguro de salud.

Se escogió una muestra de entrenamiento de 574 observaciones las cuales representan un 80% del total de la base de datos siendo así que, la muestra de prueba o "test" se tiene 144 observaciones.

Lo primero que haremos es crear la tabla de medidas para poder seleccionar la mejor regla, de esta manera con el siguiente comando en R se realizan los cálculos para poder obtener la entriopia, cobertura y precisión.

```
medidas=function(x,y,z,grupo){
          medida<-matrix(nrow = length(x), ncol=3)
          for(i in 1:length(x)){
             medida[i,1]<-(-x[i]/z[i]*log2(x[i]/z[i]))-(y[i]/z[i]*log2(y[i]/z[i]))
             medida[i,2]<-z[i]/length(grupo$health.ins)
             medida[i,3]<-x[i]/z[i]
          }
          colnames(medida)<-c("entriopia","cobertura","precision")
          return(medida)
}
```

Para facilitar el manejo de la base se realizó el siguiente cambio:

```
cierto<-subset(train,health.ins=="TRUE")
falso<-subset(train,health.ins=="FALSE")
```

Los comandos anteriores crean de la base de entrenamiento dos sub-bases. El primer comando "cierto" corresponde a las personas que SÍ cuentan con un seguro de salud y el de "falso" corresponde a aquellas que NO cuentan con seguro de salud.

El siguiente paso es contar cuantas personas cuentan con seguro de salud para cada valor posible de cada atributo, cuántas no cuentan con seguro para cada valor posible de cada atributo y cuántas personas hay en general para cada valor posible de cada atributo, el código correspondiente a lo anterior es el siguiente:

```
x<-unlist(sapply(apply(cierto[,c(1,2)],2,FUN = count ), "[",,2),
use.names = FALSE)

y<-unlist(sapply(apply(falso[,c(1,2)],2,FUN = count ), "[",,2),
use.names = FALSE)

z<-unlist(sapply(apply(train[,c(1,2)],2,FUN = count ), "[",,2),
use.names = FALSE)

med<-medidas(x,y,z,train)

med
```

Med es la tabla que nos da la tabla de las mediciones, la cual fue la siguiente:

```
     entriopia  cobertura precision
[1,] 0.5097345 0.44599303 0.8867188
[2,] 0.6784232 0.55400697 0.8207547
[3,] 0.5770043 0.17770035 0.8627451
[4,] 0.4860132 0.51219512 0.8945578
[5,] 0.8767163 0.23519164 0.7037037
[6,] 0.1593501 0.07491289 0.9767442
```

Escogemos la regla 6, que es la mejor opción en entriopía y presición, pero no en cobertura, por tanto el candidato a la regla 1 es: **Si marital status=Widowed entonces health.ins=True.**

```
regla1_1<-med[6,]
```

Como siguiente paso agregamos la condición marital.stat "Widowed" a la regla (Regla 1) y contamos cuantas personas con healt.ins son TRUE para cada valor posible de cada atributo restante.

```
x1[1]<-length(which(cierto$marital.stat == "Widowed" & cierto$sex
== "F"))
x1[2]<-length(which(cierto$marital.stat == "Widowed" & cierto$sex
== "M"))
x1
```

Contamos cuántas personas con healt.ins son FALSE para cada valor posible de cada atributo restante

```
y1[1]<-length(which(falso$marital.stat == "Widowed" & falso$sex
== "F"))
y1[2]<-length(which(falso$marital.stat == "Widowed" & falso$sex
== "M"))
y1
```

Contamos cuántas personas viudas hay para cada valor posible de cada atributo restante e imprimimos las medidas de calidad.

```
z1[1]<-length(which(train$marital.stat == "Widowed" & train$sex
== "F"))
z1[2]<-length(which(train$marital.stat == "Widowed" & train$sex
== "M"))
z1
med<-medidas(x1,y1,z1,train)
med
```

Se obtuvieron los siguientes resultados:

```
     entriopia  cobertura precision
[1,]       NaN 0.05923345 1.0000000
[2,] 0.5032583 0.01567944 0.8888889
```

Debido a que obtuvimos un valor no definido escogemos la regla 2, la cual es: Si marital status=Widowed & Sex=M entonces health.ins=True.

```
regla1_2<-med[2,]
```

Como ya no hay más atributos por clasificar trabajamos con 2 reglas y las comparamos:

```
> regla1_1
 entriopia  cobertura  precision
0.15935006 0.07491289 0.97674419
> regla1_2
 entriopia  cobertura  precision
0.50325833 0.01567944 0.88888889
```

Se escogió la regla 1_1 ya que es la mejor opción en entriopía y presición

Se vuelve a realizar todo lo anterior mencionado pero dejando de evaluar aquellas tuplas cubiertas por las reglas, es decir, se crean nuevas subbases de "cierto" y "falso" en las cuales ya no se encuentran aquellas peronas que tienen estatus marital "Widowed".

Luego se cuentan las personas que cuentan con seguro y cuántas no para cada valor posible de atributo y cuántas personas hay en general para cada valor posible de atributo. Al realizar lo anterior obtenemos las siguientes medidas de calidad:

```
     entriopia cobertura precision
[1,] 0.5591652 0.4180791 0.8693694
[2,] 0.6827600 0.5819209 0.8187702
[3,] 0.5770043 0.1920904 0.8627451
[4,] 0.4860132 0.5536723 0.8945578
[5,] 0.8767163 0.2542373 0.7037037
```

Escogemos la regla 4, la cual es la mejor opción en entriopía y precisión, por lo que el candidato a regla 2 es: Si marital status=married entonces health.ins=True. Se agrega la condición marital.stat "Married" a la regla (Regla 1).

Volvemos a contar cuántas personas cuentan con seguro y cuántas no para cada valor posible de cada atributo restante y cuántas personas casadas hay para cada valor posible de cada atributo restante y obtenemos la siguiente tabla de resultados:

```
     entriopia cobertura precision
[1,] 0.3478169 0.1732580 0.9347826
[2,] 0.5400799 0.3804143 0.8762376
```

Escogemos la regla 1 siendo asi que el candidato a la regla 2 es: si marital status=Married y Sex=F entonces health.ins=True.

```
> regla2_1
entriopia cobertura precision
0.4860132 0.5536723 0.8945578
> regla2_2
entriopia cobertura precision
0.3478169 0.1732580 0.9347826
```

Como ya no hay mas atributos por clasificar, trabajamos con 2 reglas y las comparamos:

Escogemos la regla 2_2 ya que presenta la mejor opción en entriopía y presición

Volvemos a realizar todo lo anterior mencionado pero dejando de evaluar aquellas tuplas cubiertas por las reglas, es decir, se crean nuevas subbases de "cierto" y "falso" en las cuales ya no se encuentran aquellas peronas que tienen estatus marital "Married" y sexo "F".

Luego se cuentan las personas que cuentan con seguro y cuántas no para cada valor posible de atributo y cuántas personas hay en general para cada valor posible de atributo. Al realizar lo anterior obtenemos las siguientes medidas de calidad:

```
     entriopia cobertura precision
[1,] 0.6732994 0.2961276 0.8230769
[2,] 0.6827600 0.7038724 0.8187702
[3,] 0.5770043 0.2323462 0.8627451
[4,] 0.5400799 0.4601367 0.8762376
[5,] 0.8767163 0.3075171 0.7037037
```

Escogemos la regla 4, la cual es la mejor opción en entriopía y precisión. Como marital.stat= “Married” vuelve a ser la mejor opción, no tiene sentido comparar las medidad de calidad y decidimos que la regla 2 se elimine y quede como regla si marital.stat=”Married” entonces la persona cuenta con seguro de salud.

Volvemos a crear nuevas subbases de “cierto” y “falso” en las cuales ya no se encuentran aquellas peronas que tienen estatus marital “Married”.

Luego se cuentan las personas que cuentan con seguro y cuántas no para cada valor posible de atributo y cuántas personas hay en general para cada valor posible de atributo. Al realizar lo anterior obtenemos las siguientes medidas de calidad:

```
     entriopia cobertura precision
[1,] 0.6732994 0.5485232 0.8230769
[2,] 0.8683588 0.4514768 0.7102804
[3,] 0.5770043 0.4303797 0.8627451
[4,] 0.8767163 0.5696203 0.7037037
```

Escogemos la regla 3, que es la mejor opción en entriopía y precisión y la regla es: si marital.stat =="Divorced/Separated" entonces health.ins=True. Agregamos la condición marital.stat "Divorced/Separated" a la regla (Regla 1)

Volvemos a contar cuántas personas cuentan con seguro y cuántas no para cada valor posible de cada atributo restante y cuántas personas casadas hay para cada valor posible de cada atributo restante y obtenemos la siguiente tabla de resultados:

```
     entriopia  cobertura precision
[1,] 0.5032583 0.11864407 0.8888889
[2,] 0.6789539 0.07344633 0.8205128
```

Escogemos la regla 1 siendo así que el candidato a la regla 2 es: si marital status=Divorced/Separarted y Sex=F entonces health.ins=True.

```
> regla3_1
entriopia cobertura precision
0.5770043 0.4303797 0.8627451
> regla3_2
entriopia cobertura precision
0.5032583 0.1186441 0.8888889
```

Como ya no hay más atributos por clasificar, trabajamos con 2 reglas y las comparamos:

No hay mucha diferencia entre entriopia y precisión, así que escogemos la regla 3_1 porque tiene mayor cobertura.

Entonces, las reglas son:

- Si es viudo o
- SI esta casado o
- SI esta divorciado/separado
- Entonces la persona tiene seguro de vida

La matriz de confusión es la siguiente:

		Valores Reales	
		Falso	**Verdad**
Predicción	**Falso**	**11**	**27**
	Verdad	**15**	**91**

Clase positiva: Verdad

Se obtuvo una exactitud del 70.83%.

Sensitivity : 0.7712

Specificity : 0.4231

Pos Pred Value : 0.8585