UNIVERSIDAD DE SANTIAGO DE CHILE
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA INFORMÁTICA

Laboratorio N 4

Integrantes: Cristóbal Donoso, Shalini Ramchandani

Curso: Análisis de datos

Profesor(a): Max Chacón Pacheco

Ayudante: Ignacio Ibañez

16 de Enero de 2019

Tabla de contenidos

1. Introducción

En el mundo cada elemento es clasificado según sus características, ya sea que un objeto sea grande, pequeño, mediano, pesado, liviano, etc. Por ejemplo, si se quisiera saber si una fruta o planta sea comestible, para esto se debería clasificar a dicha fruta o planta según sus características físicas ya que de alguna forma que sea comestible o no seguirá un patrón acorde a su clasificación.

Es por esto que, en este laboratorio, se procederá a clasificar a todos los atributos que componen al set de datos obtenidos para el estudio de la toxicidad de las setas.

1.1. Problema

Hacer uso de un clasificador bayesiano ingenuo para clasificar en comestible o venenoso a los atributos de una seta.

1.2. Objetivos

Los objetivos principales para esta entrega son:

- Hacer uso del clasificador bayesiano
- Analizar resultados.
- Concluir al respecto.

1.3. Estructura del informe

El documento consta de 5 capítulos: Marco teórico, donde se explica que es el clasificador bayesiano ingenuo y las probabilidades a priori y posteriori. Luego, se obtiene el clasificador bayesiano ingenuo. Además, se realiza un análisis de los resultados obtenidos por el clasificador desarrollado. Finalmente, se obtienen conclusiones respecto al problema en sí y el análisis realizado.

2. Marco teórico

2.1. Clasificador bayesiano

Técnica de clasificación que consiste en construir modelos que predicen probabilidad de un posible resultado. Dicho clasificador está basado en el Teorema de bayes en el cuál se puede calcular una probabilidad de un suceso 'A' condicionada por un suceso 'B'. Para lograr lo anterior, es necesario entrenar el modelo.

2.2. Probabilidad a priori

La probabilidad a priori P(C_i) es la probabilidad de que un suceso 'A' sea clasificado en C_i. La fórmula para obtener dicha probabilidad es la siguiente:

$$P(C_i) = \lim_{x \to \infty} \frac{n_i}{n} \tag{1}$$

2.3. Probabilidad a posteriori

La probabilidad a posteriori P(C_i/x) es la probabilidad de que un suceso 'A' pertenezca a la clase C_i dado que el valor del nivel sérico es x. La fórmula para obtener dicha probabilidad es la siguiente:

$$P(C_i/x) = \frac{P(x/C_i)P(C_i)}{P(x)} \tag{2}$$

,donde x es un suceso cualquiera, por lo que $P(x/C_i)$ es conocido y por ende P(x) también es conocido. Además $P(C_i)$ es la probabilidad a priori.

3. Obtención del clasificador

Para efectos de este laboratorio, se decide eliminar el atributo veil-type, esto ya que en el laboratorio 1 se analizó y se llegó a la conclusión de que es un atributo de tipo constante por lo que no aporta información relevante al estudio.

Por otro lado, el atributo stalk-root contiene 2480 valores desconocidos por lo que se decide eliminar solamente dichos valores ya que podría producir una disminución en la precisión del estudio, se mantienen el resto de valores para este atributo.

```
> filtered.datos <- datos[datos$stalk.root != "?",]
> filtered.datos <- subset(filtered.datos, select = -c(veil.type))
>
```

Figura 1: Filtrado datos

Ahora bien, se generan 5 conjuntos que se caracterizan por tener 20 % de los datos (1128), 40 % de los datos (2257), 60 % de los datos (3386), 80 % de los datos (4515) y 95 % de los datos (5361), esto para luego efectuar una comparación de la precisión obtenida por el clasificador bayesiano ingenuo. Cabe recordar que el número total de datos es de 5644 datos (100 %).

Luego, se procede a realizar el clasificador bayesiano con cada uno de los conjuntos, cabe destacar que dichos conjuntos van a sufrir variaciones en los datos ya que escogerá aleatoriamente valores acorde al tamaño del conjunto que se le otorgue como parámetro a la función, esto implica que los resultados por lo general también sufrirán cambios, es por eso que luego se calculará un promedio a la precisión.

```
> mush_test <- sample(1:nrow(filtered.datos), nrow(filtered.datos)*0.95)
> nb_mush <- naiveBayes(filtered.datos[-mush_test, columnas], filtered.datos[-mush_test, "class"])
>
```

Figura 2: Clasificador bayesiano ingenuo con un tamaño de datos del 95 %

Además, luego de realizar el clasificador, se crean tablas de contingencia para mostrar los datos clasificados correctamente e incorrectamente.

```
> datosTabla <- filtered.datos[mush_test,]
> tabla1.c <- table(pred_mush,datosTabla$class)
> tabla1.c
```

Figura 3: Tablas de contingencia para cada conjunto

3.1. Resultados obtenidos

Los resultados obtenidos para cada uno de los conjuntos son los siguientes:

Tamaño conjunto [%]	**Correctamente clasificados**	**Total de datos**	**Precisión**
20	1074	1129	95.15
40	2138	2258	94.70
60	3271	3386	96.59
80	4326	4515	95.81
95	5028	5362	93.77

Cuadro 1: Precisión obtenida en cada conjunto

A continuación se mostrarán las tablas de contingencia obtenidas:

	Edible [e]	**Poisonous [p]**	**Total**
Edible [e]	3300	317	3617
Poisonous [p]	16	1728	1744
Total	3316	2045	5361

Cuadro 2: Tabla de contingencia conjunto del 95 %

	Edible [e]	**Poisonous [p]**	**Total**
Edible [e]	2767	173	2940
Poisonous [p]	16	1559	1575
Total	2783	1732	4515

Cuadro 3: Tabla de contingencia conjunto del 80 %

	Edible [e]	**Poisonous [p]**	**Total**
Edible [e]	2099	112	2211
Poisonous [p]	3	1172	1175
Total	2102	1284	3386

Cuadro 4: Tabla de contingencia conjunto del 60 %

	Edible [e]	**Poisonous [p]**	**Total**
Edible [e]	1374	113	1487
Poisonous [p]	6	764	770
Total	1380	877	2257

Cuadro 5: Tabla de contingencia conjunto del 40 %

	Edible [e]	**Poisonous [p]**	**Total**
Edible [e]	683	51	734
Poisonous [p]	3	391	394
Total	686	442	1128

Cuadro 6: Tabla de contingencia conjunto del 20 %

4. Análisis de los resultados

En esta sección se expondrá el análisis de resultados para el desarrollo del clasificador bayesiano ingenuo implementado para el laboratorio 4 de análisis de datos.

Como se puede observar en el Cuadro 1, la precisión del clasificador bayesiano ingenuo se mantuvo por sobre el 90 % en todos los conjuntos. Se puede observar que la mejor clasificación la obtiene el conjunto con tamaño del 60 % de los datos, implicando un tamaño de datos igual a 3386, lo cual es un tamaño considerable para un estudio de ésta embergadura. Ahora bien, el menor porcentaje de precisión lo obtiene el conjunto de tamaño del 95 %, con un número de datos igual a 5362, esto se puede deber a la gran variedad de datos que existen con este tamaño. A pesar de lo anterior, la diferencia de precisión entre ambos casos mencionados anteriormente solo es de un 1.82 %, lo cuál es bastante poco para un aumento de 1976 datos.

Por otro lado, se realiza una comparación con el algoritmo de clustering k-means realizado en el laboratorio 2, el cuál entregó un porcentaje de precisión igual a 94.42 % para el caso en que se utilizaban cinco clusters.

Clasificador	**Precisión** [%]
Clasificador bayesiano ingenuo 95 %	93.77
Clasificador bayesiano ingenuo 60 %	96.59
Algoritmo de clustering k-means (5 clústers)	94.42

Cuadro 7: Comparación clasificador bayesiano ingenuo con algoritmo de clustering k-means

Debido a lo anterior, se puede deducir que, para un caso en el que el tamaño de datos igual al 60 % del total de datos para la base de datos mushroom, el mejor clasificador es el clasificador bayesiano ingenuo con una precisión total del 96.59 %.

5. Conclusiones

Se hace uso del clasificador bayesiano ingenuo para poder realizar una correcta clasificación de la base de datos mushroom seleccionada para este laboratorio. Se crean cinco conjuntos de los cuáles destaca el conjunto que tiene un tamaño de datos igual al 60 % del total de datos con una precisión igual al 96.59 %.

Debido al análisis anterior, se procede a realizar una comparación con el algoritmo de clustering k-means realizado en el laboratorio 2 y se concluye que el clasificador bayesiano con el tamaño del conjunto mencionado anteriormente otorga una mayor precisión al clasificar, la diferencia es de un 2.17 %.

Por lo tanto, se puede concluir que el laboratorio fue desarrollado con éxito, cumpliéndose así todos los objetivos planteados con anterioridad.

Bibliografía

[1] CHristianCH (02/05/2013). Clasificador Naïve Bayes. ¿Cómo funciona? naivebayes.blogspot Recuperado de http://naivebayes.blogspot.com/

6. Anexo

```r
library("ggplot2")
library("e1071")
library("corrplot")

#LECTURA DE DATOS
columnas <- c("class","cap-shape","cap-surface","cap-color","bruises","odor","gill-attachment","gill-spacing","gill-size","gill-color"
              ,"stalk-shape","stalk-root","stalk-surface-above-ring","stalk-surface-below-ring","stalk-color-above-ring"
              ,"stalk-color-below-ring","veil-type","veil-color","ring-number","ring-type","spore-print-color","population","habitat")
datos <- read.csv("D:/Universidad/Analisis de datos/Lab 4/agaricus-lepiota.data", header=FALSE,
                  sep=",", col.names = columnas)

#Porcentaje de datos inicial:
#   - edible: 51.8% (4208)
#   - poisonous: 48.2% (3916)

#LIMPIEZA DE DATOS
#Se eliminan las tuplas o registros que en el atributo "stalk-root" no tienen ning?n valor registrado.

#Se filtran 2480 datos, es decir, nos quedamos con el 69,5% de los datos.
#El porcentaje de datos por cada variable de clase, despu?s de filtrado:
#   - edible: 61,8% (3488)
#   - poisonous: 38.2% (2156)
filtered.datos <- datos[datos$stalk.root != "?",]

#Se elimina el atributo veil-type ya que es una elemento de tipo constante por lo que no aporta al estudio.
#Se mantiene el número total de datos, no hay variación absoluta en este.
filtered.datos <- subset(filtered.datos, select = -c(veil.type))


columnas <- c("cap.shape","cap.surface","cap.color","bruises","odor","gill.attachment","gill.spacing","gill.size","gill.color"
              ,"stalk.shape","stalk.root","stalk.surface.above.ring","stalk.surface.below.ring","stalk.color.above.ring"
              ,"stalk.color.below.ring","veil.color","ring.number","ring.type","spore.print.color","population","habitat")


#Conjunto de entrenamiento del 95%
mush_test <- sample(1:nrow(filtered.datos), nrow(filtered.datos)*0.95)
nb_mush <- naiveBayes(filtered.datos[-mush_test, columnas], filtered.datos[-mush_test, "class"])
pred_mush <- predict(nb_mush, filtered.datos[mush_test, columnas])

datosTabla <- filtered.datos[mush_test,]
tabla1.c <- table(pred_mush,datosTabla$class)
tabla1.c

clasificadosCorrectamente <- sum(pred_mush==filtered.datos[mush_test, "class"])
totalDatos <- nrow(filtered.datos)*0.95
clasificadosCorrectamente
totalDatos
precision <- (clasificadosCorrectamente / totalDatos) * 100
precision

#Conjunto de entrenamiento del 80%
mush_test <- sample(1:nrow(filtered.datos), nrow(filtered.datos)*0.8)
nb_mush <- naiveBayes(filtered.datos[-mush_test, columnas], filtered.datos[-mush_test, "class"])
pred_mush <- predict(nb_mush, filtered.datos[mush_test, columnas])

datosTabla <- filtered.datos[mush_test,]
tabla1.c <- table(pred_mush,datosTabla$class)
tabla1.c

clasificadosCorrectamente <- sum(pred_mush==filtered.datos[mush_test, "class"])
totalDatos <- nrow(filtered.datos)*0.8
clasificadosCorrectamente
totalDatos
precision <- (clasificadosCorrectamente / totalDatos) * 100
precision
```

```
66
67  #Conjunto de entrenamiento del 60%
68  mush_test <- sample(1:nrow(filtered.datos), nrow(filtered.datos)*0.6)
69  nb_mush <- naiveBayes(filtered.datos[-mush_test, columnas], filtered.datos[-mush_test, "class"])
70  pred_mush <- predict(nb_mush, filtered.datos[mush_test, columnas])
71
72  datosTabla <- filtered.datos[mush_test,]
73  tabla1.c <- table(pred_mush,datosTabla$class)
74  tabla1.c
75
76  clasificadosCorrectamente <- sum(pred_mush==filtered.datos[mush_test, "class"])
77  totalDatos <- nrow(filtered.datos)*0.6
78  clasificadosCorrectamente
79  totalDatos
80  precision <- (clasificadosCorrectamente / totalDatos) * 100
81  precision
82
83  #Conjunto de entrenamiento del 40%
84  mush_test <- sample(1:nrow(filtered.datos), nrow(filtered.datos)*0.4)
85  nb_mush <- naiveBayes(filtered.datos[-mush_test, columnas], filtered.datos[-mush_test, "class"])
86  pred_mush <- predict(nb_mush, filtered.datos[mush_test, columnas])
87
88  datosTabla <- filtered.datos[mush_test,]
89  tabla1.c <- table(pred_mush,datosTabla$class)
90  tabla1.c
91
92  clasificadosCorrectamente <- sum(pred_mush==filtered.datos[mush_test, "class"])
93  totalDatos <- nrow(filtered.datos)*0.4
94  clasificadosCorrectamente
95  totalDatos
96  precision <- (clasificadosCorrectamente / totalDatos) * 100
97  precision
```

```
 99  #Conjunto de entrenamiento del 20%
100  mush_test <- sample(1:nrow(filtered.datos), nrow(filtered.datos)*0.2)
101  nb_mush <- naiveBayes(filtered.datos[-mush_test, columnas], filtered.datos[-mush_test, "class"])
102  pred_mush <- predict(nb_mush, filtered.datos[mush_test, columnas])
103
104  datosTabla <- filtered.datos[mush_test,]
105  tabla1.c <- table(pred_mush,datosTabla$class)
106  tabla1.c
107
108  clasificadosCorrectamente <- sum(pred_mush==filtered.datos[mush_test, "class"])
109  totalDatos <- nrow(filtered.datos)*0.2
110  clasificadosCorrectamente
111  totalDatos
112  precision <- (clasificadosCorrectamente / totalDatos) * 100
113  precision
```