

Benemérita Universidad Autónoma de Puebla  
Facultad de Ciencias de la Computación

# Máquinas de aprendizaje

## Reporte: Tarea 2 - Naïve Bayes



# BUAP

**Docente: Abraham Sánchez López**

**Alumno**

Taisen Romero Bañuelos

**Matrícula**

202055209

## Tarea 2 - Naive Bayes

A) Bueno, lo primero que se pide es crear una tabla dinámica

```
> #Crear tabla dinámica con table
> d_table <- table(
+   Online = data_train$Online,
+   CreditCard = data_train$CreditCard,
+   Personal.Loan = data_train$Personal.Loan
+ )
> d_table
, , Personal.Loan = 0

      CreditCard
Online    0     1
0      781   293
1     1140   477

, , Personal.Loan = 1

      CreditCard
Online    0     1
0       81    35
1      139    54

> |
```

¿Qué podemos observar a primera vista?, que la mayoría de los clientes no tienen un préstamo (Personal.Loan=0), ya que se distribuyen en mayor medida en esa tabla/matriz. También, hay muchos clientes que usan servicios en línea, para ambos casos de Personal.Loan (aunque lógicamente, al haber más clientes sin préstamo vemos que hay más en el caso de Personal.Loan=0). Dentro del conjunto de las personas con Personal.Loan=1 podemos decir que la mayoría de clientes no usa tarjeta de crédito, por lo que el perfil de quienes aceptan un préstamo no tiene mucha correlación con la posesión de una tarjeta de crédito.

B) Para calcular la probabilidad que se nos pide podríamos ver si funciona la fórmula del primer acetato sobre Bayes

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Pero supongo que el chiste de la práctica es usar el modelo con R, así que el resultado se vería así

```
$ Online      : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 2 1 2 1 ...
$ CreditCard  : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 2 1 1 ...
> new_client <- data.frame(CreditCard = factor(1, levels = levels(data_train$CreditCard)),
+                           Online = factor(1, levels = levels(data_train$Online)))
> modelo<-naiveBayes(Personal.Loan ~ CreditCard + Online, data = data_train)
> pred<-predict(modelo, new_client, type = "raw") #type = "raw" devuelve probabilidades
> pred
           0           1
[1,] 0.8927424 0.1072576
> |
```

Entonces, tenemos que lo más probable es que no nos acepte la oferta el cliente (89% de probabilidades de que no acepte). Una mala noticia para el sistema bancario.

**C)** Sobre la tabla para comparar los préstamos con las otras variables, tenemos que notar que hay más influencia de la posesión de una tarjeta de crédito que de usar servicios Online, un dato a tener en cuenta.

```
> # ==== Inciso C ====
> tab_loan_vs_online<-table(Prestamo=data_train$Personal.Loan, Online=data_train$Online)
> tab_loan_vs_online
      Online
Prestamo  0    1
0  1074 1617
1   116  193
> tab_loan_vs_cc<-table(Prestamo=data_train$Personal.Loan, CC=data_train$CreditCard)
> tab_loan_vs_cc
      CC
Prestamo  0    1
0  1921  770
1   220   89
```

**D)** Bueno, otra vez podríamos resolver lo que se pide con la fórmula del acetato, pero al igual que la vez pasada, supondré que habrá que usar R.

Nota:

Se dice que dos eventos  $A$  y  $B$  son independientes cuando la probabilidad de  $A$  no depende de la de  $B$ . En otras palabras:

$$p(A|B) = p(A)$$

i) La probabilidad es del 28%

```
> #i) P(CC = 1 | Loan = 1)
> modelo<-naiveBayes(Personal.Loan ~ CreditCard, data=data_train)
> modelo

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
  0    1
0.897 0.103

Conditional probabilities:
CreditCard
Y      0      1
0 0.7138610 0.2861390
1 0.7119741 0.2880259

> modelo$tables$CreditCard
CreditCard
Y      0      1
0 0.7138610 0.2861390
1 0.7119741 0.2880259
```

ii) La probabilidad es del 62%

```
> #ii) P(Online = 1 | Loan = 1)
> modelo<-naiveBayes(Personal.Loan ~ Online, data=data_train)
> modelo$tables$Online
  Online
Y      0      1
0 0.3991081 0.6008919
1 0.3754045 0.6245955
```

iii) La probabilidad es del 10%

```
> #iii) P(Loan = 1) (la proporción de aceptantes de préstamos)
> modelo<-naiveBayes(Personal.Loan ~ ., data=data_train)
> modelo

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y      0      1
0.897 0.103
```

iv) La probabilidad es del 28% (se resuelve con la misma salida de los comandos anteriores en (y=0, CC=1)).

v) La probabilidad es del 60% (se resuelve con la misma salida de los comandos anteriores en (y=0, Online=1)).

vi) La probabilidad es del 89% (se resuelve con la misma salida del comando anterior).

**E)** Bueno, esta vez no supe cómo resolverlo con R sin hacer directamente la ecuación en el programa. Estos fueron mis resultados

```
> p_Loan1_given_CC1_Online1
[1] 0.1073403
```

**F)** Es difícil determinar cuál es más difícil cuando ambas probabilidades son muy parecidas, pero supongo que la clave está en el número de datos con los que se entrenó al modelo. Pocos son los modelos que se benefician de tener pocos datos, creo que Bayes no es uno de ellos, por lo que no veo descabellado afirmar que la tabla dinámica es más precisa ya que está basada en Bayes.

**G)** Bueno, el resultado es algo menor con respecto al del inciso E, pero no lo veo descabellado ya que, como mencioné, Bayes funciona mejor con muchos datos y hasta cierto punto le veo sentido a que disminuya el porcentaje ya que desde un inicio sabíamos que poca gente tenía un préstamo.

```
> modelo<-naiveBayes(Personal.Loan ~ CreditCard + Online, data=data_train)
> modelo

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
  0    1
0.897 0.103
```