DigitalHouse > Coding School

DATA SCIENCE

MÓDULO 4

Naïve Bayes

OBJETIVOS DE LA CLASE

1. Conocer el funcionamiento del algoritmo Naive Bayes
2. Identificar en qué casos se utiliza: como benchmark y como clasificador eficiente
3. Implementar el modelo y evaluar la performance

Definición 1.10 Sean A y B dos eventos y supongamos que B tiene probabilidad estrictamente positiva. La probabilidad condicional del evento A, dado el evento B, se denota por el símbolo $P(A \mid B)$ y se define como el cociente

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}. \tag{1.3}$$

El término P(A/B) se lee:
"La probabilidad de A dado B"

Sea $B_1, \ldots, B_n$ una partición de Ω tal que $P(B_i) \neq 0$, $i = 1, \ldots, n$. Para cualquier evento A,

$$P(A) = \sum_{i=1}^{n} P(A \mid B_i) P(B_i).$$

Cuando la partición del espacio muestral consta de únicamente los elementos B y B^c, la fórmula del teorema de probabilidad total se reduce a la expresión

$$P(A) = P(A \mid B)\, P(B) + P(A \mid B^c)\, P(B^c).$$

- ## Independencia de Eventos

La independencia entre eventos es equivalente a la situación cuando la ocurrencia de un evento no afecta la probabilidad de ocurrencia de otro evento.

Es un concepto importante que en algunos casos nos simplificará considerablemente el cálculo de probabilidades conjuntas.

Definición 1.11 Se dice que los eventos A y B son independientes si se cumple la igualdad

$$P(A \cap B) = P(A)P(B). \tag{1.4}$$

Teorema 1.2 (Teorema de Bayes) Sea $B_1, \ldots, B_n$ una partición de Ω tal que $P(B_i) \neq 0$, $i = 1, \ldots, n$. Sea A un evento tal que $P(A) \neq 0$. Entonces para cada $j = 1, 2, \ldots, n$,

$$P(B_j \mid A) = \frac{P(A \mid B_j)P(B_j)}{\sum_{i=1}^{n} P(A \mid B_i)P(B_i)}.$$

Demostración. Por definición de probabilidad condicional, y después usando el teorema de probabilidad total, tenemos que para cada $j = 1, \ldots, n$,

$$P(B_j \mid A) = \frac{P(A \cap B_j)}{P(A)} = \frac{P(A \mid B_j)P(B_j)}{\sum_{i=1}^{n} P(A \mid B_i)P(B_i)}.$$

Es una **familia de clasificadores** simples basados en la aplicación del **Teorema de Bayes**.

Podemos ver un problema de clasificación de la siguiente forma, donde L son las labels y features es la matriz de features:

$$P(L \mid \text{features}) = \frac{P(\text{features} \mid L)P(L)}{P(\text{features})}$$

En Naive Bayes suponemos que dentro de cada clase **los features son independientes entre sí**. El teorema de Bayes indica:

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y)P(x_1, \ldots x_n \mid y)}{P(x_1, \ldots, x_n)}$$

Por independencia entre los features dentro de cada clase:

$$P(x_1 \ldots x_d|y) = \underbrace{\prod_{i=1}^{d} P(x_i|x_1 \ldots x_{i-1}, y)}_{\text{chain rule (exact)}} = \underbrace{\prod_{i=1}^{d} P(x_i|y)}_{\text{independence}}$$

Para poder resolver el problema de esta forma:

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y) \prod_{i=1}^{n} P(x_i \mid y)}{P(x_1, \ldots, x_n)}$$

$$P(y \mid x_1, \ldots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

$$\Downarrow$$

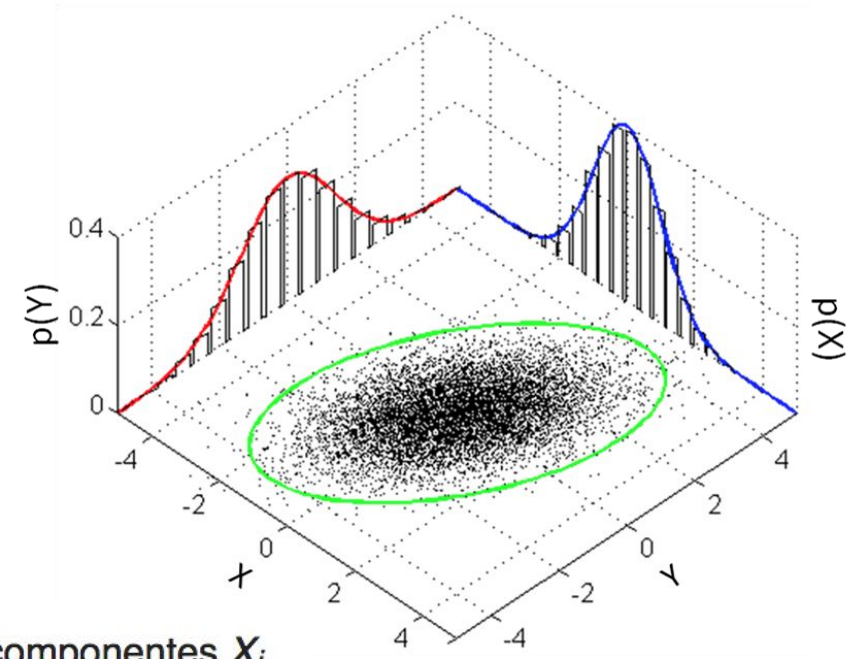
$$\hat{y} = \arg\max_y P(y) \prod_{i=1}^{n} P(x_i \mid y),$$

Como el denominador P(x1,...,xn) es constante para todas las clases, para realizar una predicción es suficiente con maximizar el numerador.

Los distintos algoritmos de Naive Bayes difieren en la distribución que suponen para $P(x_i \mid y)$.

- Gaussian Naive Bayes: Supone **distribución Gaussiana** multidimensional
- Naive Bayes Multinomial: Supone **distribución multinomial**.

Distribución Gaussiana Multivariada:

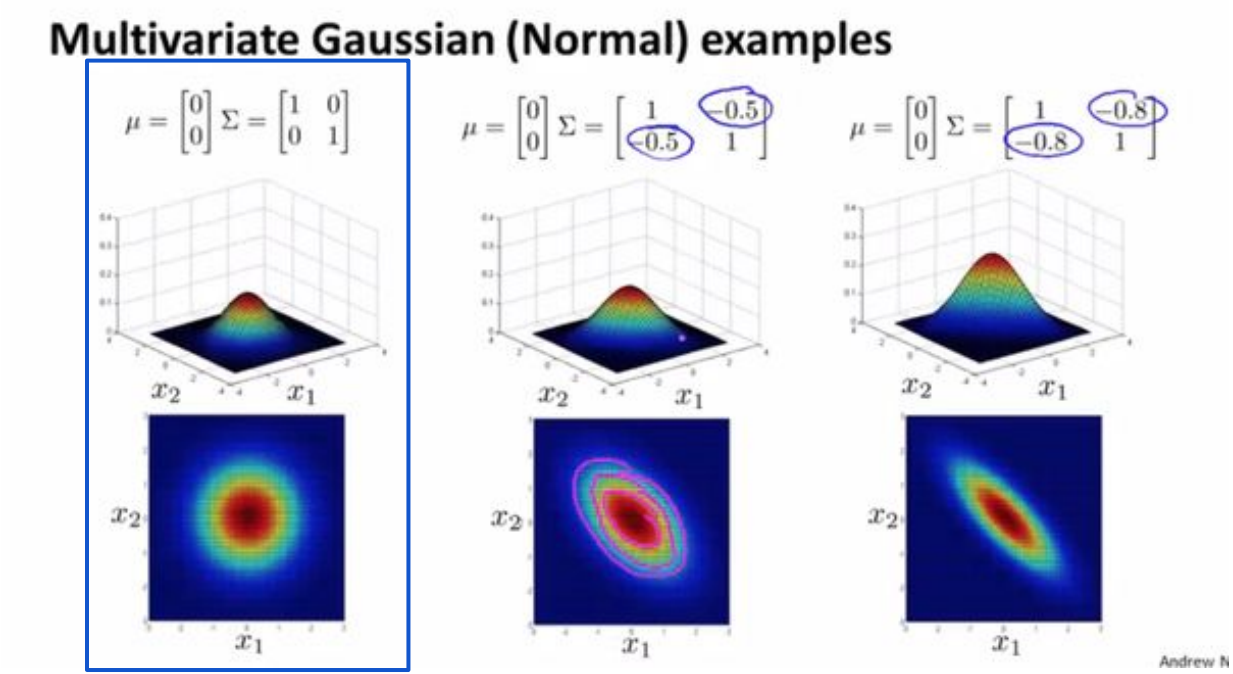


$$X \sim \mathcal{N}(\mu, \Sigma)$$

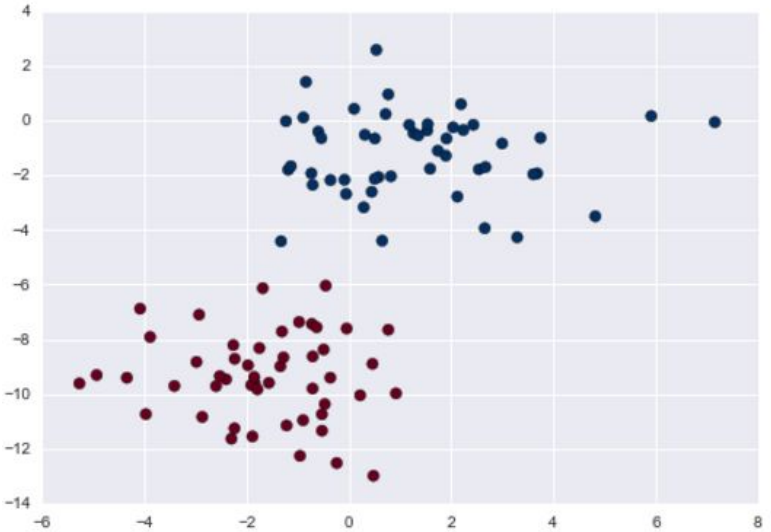
$\Sigma = AA^T$ es la matriz de covarianza de las componentes X_i.

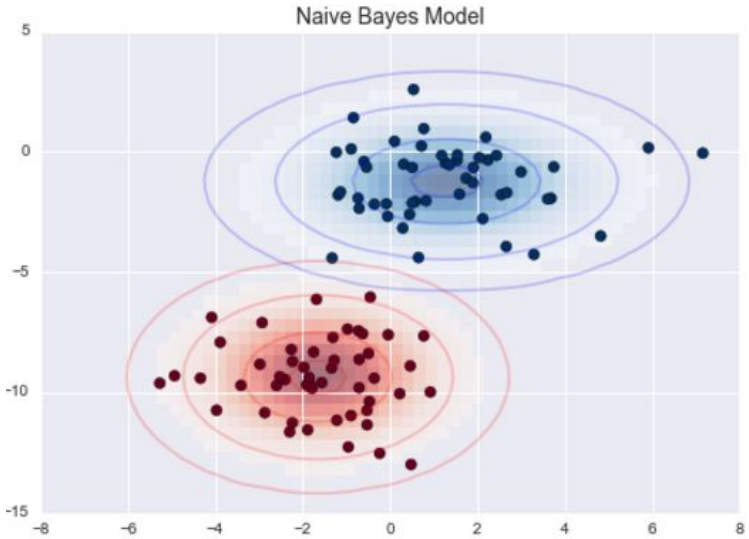
$$f_X(x_1, \ldots, x_n) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^\top \Sigma^{-1}(\mathbf{x}-\mu)\right)$$

Naive Bayes supone que cada una de las clases proviene de una distribución Gaussiana multivariada donde **las features son independientes** entre sí.

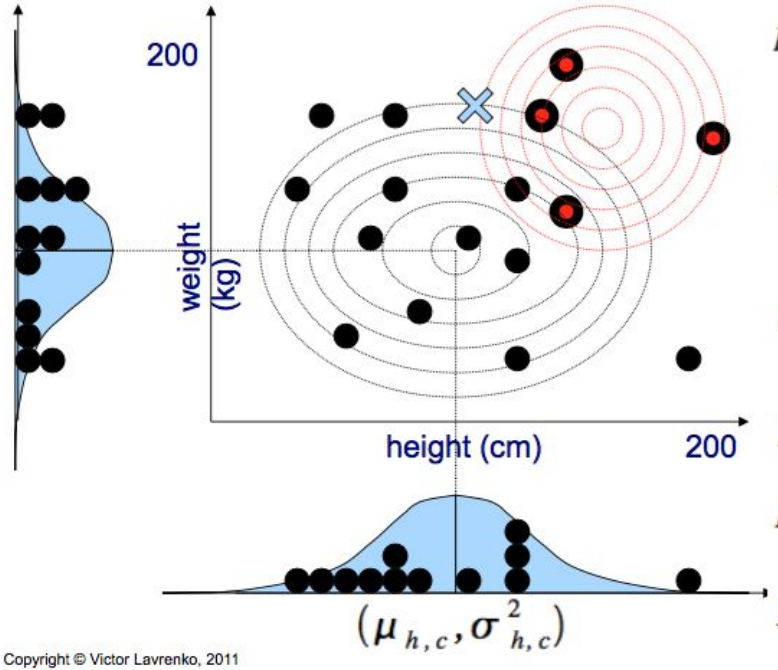


Un modelo de tipo “Gaussian Naive Bayes” toma datos como los que se ven abajo y calcula una distribución para cada una de las clases. Obtenemos para cada nuevo punto que queremos clasificar un valor de probabilidad de cada una de las clases.





$$P(a)=\frac{4}{4+12}=0.25\ ;\ P(c)=0.75$$



$$p(h_x|c)=\frac{1}{\sqrt{2\pi\,\sigma^2_{h,c}}}\exp-\frac{1}{2}\left(\frac{(h_x-\mu_{h,c})^2}{\sigma^2_{h,c}}\right)$$

$$p(w_x|c)=\frac{1}{\sqrt{2\pi\,\sigma^2_{w,c}}}\exp-\frac{1}{2}\left(\frac{(w_x-\mu_{w,c})^2}{\sigma^2_{w,c}}\right)$$

$$p(h_x|a)=\frac{1}{\sqrt{2\pi\,\sigma^2_{h,a}}}\exp-\frac{1}{2}\left(\frac{(h_x-\mu_{h,a})^2}{\sigma^2_{h,a}}\right)$$

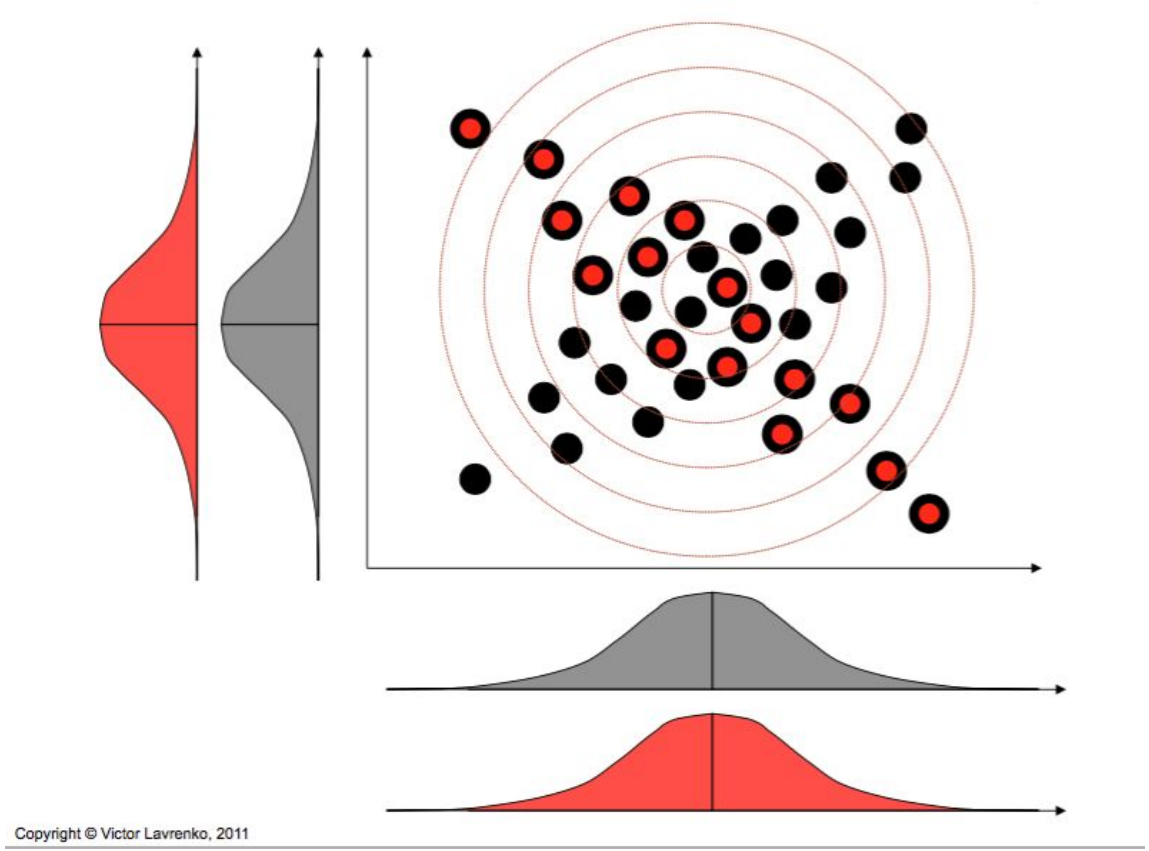
$$p(w_x|a)=\frac{1}{\sqrt{2\pi\,\sigma^2_{w,a}}}\exp-\frac{1}{2}\left(\frac{(w_x-\mu_{w,a})^2}{\sigma^2_{w,a}}\right)$$

$$P(x|a)=p(h_x|a)\,p(w_x|a)$$

$$P(x|c)=p(h_x|c)\,p(w_x|c)$$

$$P(a|x)=\frac{P(x|a)P(a)}{P(x|a)P(a)+P(x|c)P(c)}$$

Naive Bayes: problemas para el caso Gaussiano



- Separate spam from valid email, attributes = words

D1: "send us your password"	spam
D2: "send us your review"	ham
D3: "review your password"	ham
D4: "review us"	spam
D5: "send your password"	spam
D6: "send us your account"	spam

new email: "review us now"

P (spam) = 4/6 P (ham) = 2/6

spam	ham	
2/4	1/2	password
1/4	2/2	review
3/4	1/2	send
3/4	1/2	us
3/4	1/2	your
1/4	0/2	account

$$P(\text{review us}\,|\text{spam}) = P(0,1,0,1,0,0\,|\text{spam}) = (1-\frac{2}{4})(\frac{1}{4})(1-\frac{3}{4})(\frac{3}{4})(1-\frac{3}{4})(1-\frac{1}{4})$$

$$P(\text{review us}\,|\text{ham}) = P(0,1,0,1,0,0\,|\text{ham}) = (1-\frac{1}{2})(\frac{2}{2})(1-\frac{1}{2})(\frac{1}{2})(1-\frac{1}{2})(1-\frac{0}{2})$$

$$P(\text{ham}\,|\text{review us}) = \frac{0.0625\times 2/6}{0.0625\times 2/6 + 0.0044\times 4/6} = 0.87$$

Algunos problemas que puede presentar el caso discreto:

- **Problema de frecuencia cero**:
  - Por ejemplo, en el caso anterior, cualquier mail que contenga la palabra “account” va a ser considerado spam porque P(account/ham)=0/2.
  - Solución: agregar un pequeño valor positivo al conteo (**Laplace smoothing**): P(w/c) = num(w/c)+alpha / num(w) + 2*alpha
- **Asume independencia de las features** (en nuestro ejemplo, palabras):
  - se puede engañar al clasificador agregando muchas palabras asociadas con mail “no spam” a un mail de “spam”

Los clasificadores basados en Naive Bayes hacen **fuertes supuestos sobre los datos**, así que no van a tener tan buena performance si el verdadero proceso generador de los datos no cumple con los supuestos de Naive Bayes.

Dicho esto, tienen las siguientes ventajas:

- Son algoritmos **muy rápidos** tanto para entrenar como para predecir
- Brindan una **predicción probabilística** (tenemos probabilidades para cada clase)
- Son **sencillos de interpretar**
- No requieren "tunear" ningún hiperparámetro

Dado que Naive Bayes es tan fácil de optimizar y tan rápido desde el punto de vista computacional, es un buen "**baseline**" para un problema de clasificación.

Si performa bien, podemos quedarnos con este modelo y si necesitamos mejorar la precisión, tenemos una línea de base sobre la cual mejorar.

Naive Bayes tiende a funcionar bien en las siguientes situaciones:

- ***Cuando se cumplen los supuestos*** (cosa que rara vez pasa en casos reales)
- ***Cuando las clases están muy bien separadas*** y no es necesaria tanta complejidad en el modelo.
- ***Cuando tenemos datos con muy alta dimensionalidad*** (por ejemplo text mining) donde la complejidad del modelo también es menos importante porque hay mucha información para cada observación.