



**AWAKELAB**

**BASECAMP**

Ciencia de Datos

## Módulo: Aprendizaje de Máquina Supervisado

---

### Aprendizaje Esperado

---

7. Elabora un modelo predictivo aplicando el algoritmo clasificador Bayesiano para resolver un problema de clasificación utilizando lenguaje Python.

---

### Naive Bayes o Bayes ingenuo

The diagram illustrates Bayes' Theorem with the formula  $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$ . The components are annotated as follows:

- LIKELIHOOD** (orange text): the probability of "B" being TRUE given that "A" is TRUE. An arrow points to the term  $P(B|A)$  in the numerator.
- PRIOR** (green text): the probability of "A" being TRUE. An arrow points to the term  $P(A)$  in the numerator.
- POSTERIOR** (green text): the probability of "A" being TRUE given that "B" is TRUE. An arrow points to the term  $P(A|B)$  on the left side of the equation.
- Marginal Probability** (pink text): The probability of "B" being TRUE. An arrow points to the term  $P(B)$  in the denominator.

Corresponde a una técnica de Machine Learning de tipo Supervisado. Se denomina como un clasificador estadístico pues se basa en el famoso Teorema de Bayes, predice las probabilidades de pertenecer a las distintas clases y luego predice la clase dependiendo de cuál es más probable.

Se llama Bayes ingenuo pues asume independencia condicional en los features. Es fácil de construir, preciso y rápido para conjuntos de datos muy grandes.

### Teorema de Bayes

El teorema de Bayes (también conocido como regla de Bayes) es una fórmula engañosamente simple que se usa para calcular la probabilidad condicional. El Teorema lleva el nombre del matemático inglés Thomas Bayes (1701-1761). La definición formal de la regla es:

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

En la mayoría de los casos, no puede simplemente insertar números en una ecuación; primero debe averiguar cuáles son sus "pruebas" y "eventos". Para dos eventos, A y B, el teorema de Bayes le permite calcular  $P(A|B)$  (la probabilidad de que ocurra el evento A, dado que la prueba B fue positiva) a partir de  $P(B|A)$  (la probabilidad de que la prueba B sucedió, dado que sucedió el evento A). Puede ser un poco complicado entenderlo, ya que técnicamente estás trabajando al revés; es posible que deba cambiar sus pruebas y eventos, lo que puede resultar confuso. Un ejemplo debería aclarar lo que quiero decir con "cambiar las pruebas y los eventos".

### Naive Bayes: Independencia condicional

En este modelo, los features (inputs)  $X_1, \dots, X_m$  y la clase  $Y$  se asume que cumplen independencia condicional:

$$P(X_i|X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_m, Y) = P(X_i|Y)$$

$$P(Y) = \prod_{i=1}^m P(X_i|Y)$$

### ¿Por qué asumir la independencia condicional?

Estimar funciones multivariantes del tipo  $P(Y)$  es matemáticamente más complejo que estimar funciones univariantes del tipo  $P(X_i|Y)$ .

Las funciones univariantes necesitan menos datos para ajustar bien.

Estimadores univariantes difieren mucho menos que los estimadores multivariantes (menos varianza) aunque posiblemente estos estimadores no sean los correctos (más sesgo).

## El modelo de Naive Bayes

La idea es encontrar:  $P(Y = y|X_1 = x_1, \dots, X_m = x_m)$  (la probabilidad condicional de la variable target dados los predictores).

Usando el teorema de Bayes y el supuesto de independencia condicional, tenemos que:

$$\begin{aligned} & P(Y = y|X_1 = x_1, \dots, X_m = x_m) \\ & \propto P(Y = y)P(X_1 = x_1, \dots, X_m = x_m|Y = y) \\ & \approx P(Y = y) \prod_{i=1}^m P(X_i = x_i|Y = y) \end{aligned}$$

Luego, la predicción será la clase que maximiza la probabilidad condicional:

$$\hat{y}^{NB} = \underset{y}{\operatorname{argmax}} \left\{ P(Y = y) \prod_{i=1}^m P(X_i = x_{\{test,i\}}|Y = y) \right\}$$

## Procedimiento

1. Convertir el conjunto de datos en una tabla de frecuencias.
2. Crear una tabla de probabilidad marginal para cada uno de los posibles casos.
3. Usar Bayes para calcular la probabilidad posterior de cada clase.
4. La clase con la probabilidad posterior más alta es el resultado de la predicción.

### Ejemplo: Predecir según el clima si un jugador debería jugar o no

Utilice los siguientes datos y use la idea del Naive Bayes para predecir dado que el clima es soleado si el jugador debería jugar o no.

**Paso 1:** Debemos convertir el conjunto de datos en una tabla de frecuencias.

	Yes	No
Overcast	5	0
Rainy	2	2
Sunny	3	2
Total	10	5

Index	Outlook	Play
0	Rainy	Yes
1	Sunny	Yes
2	Overcast	Yes
3	Overcast	Yes
4	Sunny	No
5	Rainy	Yes
6	Sunny	Yes
7	Overcast	Yes
8	Rainy	No
9	Sunny	No
10	Sunny	Yes
11	Rainy	No
12	Overcast	Yes
13	Overcast	Yes

**Paso 2:** Crear una tabla de probabilidad marginal para cada uno de los posibles casos.

	Yes	No	
Overcast	5	0	$\frac{5}{14} = 0.35$
Rainy	2	2	$\frac{4}{14} = 0.29$
Sunny	3	2	$\frac{5}{14} = 0.35$
All	$\frac{10}{14} = 0.71$	$\frac{4}{14} = 0.29$	1

Index	Outlook	Play
0	Rainy	Yes
1	Sunny	Yes
2	Overcast	Yes
3	Overcast	Yes
4	Sunny	No
5	Rainy	Yes
6	Sunny	Yes
7	Overcast	Yes
8	Rainy	No
9	Sunny	No
10	Sunny	Yes
11	Rainy	No
12	Overcast	Yes
13	Overcast	Yes

**Paso 3:** Usar Bayes para calcular la probabilidad posterior de cada clase.

Necesitamos calcular  $P(\text{Yes} | \text{Sunny})$  y  $P(\text{No} | \text{Sunny})$ . Usando Bayes tenemos que:

$$P(\text{Yes} | \text{Sunny}) = \frac{P(\text{Sunny} | \text{Yes})P(\text{Yes})}{P(\text{Sunny})}$$

Y sabemos que:

- $P(\text{Sunny}) = \frac{5}{14}$
- $P(\text{Yes}) = \frac{10}{14}$
- $P(\text{Sunny} | \text{Yes}) = \frac{3}{10}$

Luego reemplazamos y obtenemos que:  $P(\text{Yes} | \text{Sunny}) = 0.6$

De modo análogo:

$$P(\text{No} \mid \text{Sunny}) = \frac{P(\text{Sunny} \mid \text{No})P(\text{No})}{P(\text{Sunny})}$$

Y sabemos que:

- $P(\text{Sunny}) = \frac{5}{14}$
- $P(\text{Yes}) = \frac{4}{14}$
- $P(\text{Sunny} \mid \text{Yes}) = \frac{2}{4}$

Luego reemplazamos y obtenemos que:  $P(\text{No} \mid \text{Sunny}) = 0.4$ .

Como  $P(\text{Yes} \mid \text{Sunny}) > P(\text{No} \mid \text{Sunny})$  se predice que el jugador debería jugar. Este proceso se extrapola a más predictores usando independencia condicional.

Ventajas	Desventajas
Puede no estimar las probabilidades de clase con exactitud, pero suele tomar decisiones correctamente.	No le va muy bien cuando hay muchos datos y pocos predictores.
Han demostrado tener alta velocidad al aplicarse a grandes bases de datos.	Dado el supuesto de independencia, el modelo probablemente no refleja cómo son los datos en el mundo real.

**Ejemplo en Python**

La base de datos iris contiene distintas características de flores, las cuales están etiquetadas con su especie (setosa=0, versicolor=1, virginica=2).

```
import numpy as np

import pandas as pd

from sklearn import datasets

data = datasets.load_iris()

X = pd.DataFrame(data['data'],
                  columns=data['feature_names'])

y = data['target']

from sklearn.model_selection import train_test_split

X_train,X_test,y_train,y_test=train_test_split(X,y,te
st_size=0.3,random_state=0) #split de entrenamiento y
prueba

from sklearn.naive_bayes import MultinomialNB

model=MultinomialNB(fit_prior=False).fit(X_train,
y_train) #ajustamos el modelo

y_predicted = model.predict(X_test) #Obtenemos
predicciones en el set de prueba

pd.crosstab(y_test, y_hat) #Matriz de confusión
```



col_0	0	1	2
row_0			
0	16	0	0
1	0	17	1
2	0	0	11

## Referencias

[1] Teorema de Bayes

<https://economipedia.com/definiciones/teorema-de-bayes.html>

[2] Naive Bayes

[https://scikit-learn.org/stable/modules/naive\\_bayes.html](https://scikit-learn.org/stable/modules/naive_bayes.html)

[3] Clasificador bayesiano

<https://medium.com/datos-y-ciencia/algoritmos-naive-bayes-fundamentos-e-implementaci%C3%B3n-4bcb24b307f>

## Material Complementario

[1] Teorema de Bayes

<https://www.youtube.com/watch?v=CP4ToX5Tyvw>

[2] Qué es Naive Bayes

<https://www.youtube.com/watch?v=949tYJgRvRg>