

# Naive Bayes

(Clasificador Bayesiano Ingenuo)

Analítica de Datos, Universidad de San Andrés

Si encuentran algún error en el documento o hay alguna duda, mandenme un mail a [rodriguezr@udesa.edu.ar](mailto:rodriguezr@udesa.edu.ar) y lo revisamos.

## 1. Introducción a Naive Bayes

El clasificador Naive Bayes es un algoritmo de aprendizaje automático basado en el teorema de Bayes, que asume la independencia condicional entre las características dado el valor de la variable objetivo. A pesar de ser una suposición ingenua (naive), el algoritmo puede ser muy efectivo en algunas situaciones.

El teorema de Bayes establece que:

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

donde:

- $P(Y|X)$  es la probabilidad posterior
- $P(X|Y)$  es la verosimilitud
- $P(Y)$  es la probabilidad a priori
- $P(X)$  es la evidencia

En el contexto de clasificación, queremos encontrar la clase  $Y$  que maximiza  $P(Y|X)$  dado un conjunto de características  $X$ . Debido a la suposición de independencia, podemos escribir:

$$P(X|Y) = P(X_1|Y) \times P(X_2|Y) \times \cdots \times P(X_n|Y)$$

## 2. Tipos de Naive Bayes

Antes de aplicar un modelo Naive Bayes, es importante conocer las variantes que existen y cuándo conviene usar cada una, según la documentación oficial de `scikit-learn`:

- **BernoulliNB:** Utiliza variables binarias (presencia o ausencia de características). Es ideal cuando los datos son de tipo binario, por ejemplo, presencia/ausencia de palabras en texto.
- **CategoricalNB:** Se utiliza para datos categóricos, donde las características son discretas y tienen más de dos posibles valores, común en datos no numéricos.
- **MultinomialNB:** Ideal para datos discretos como conteos de palabras en texto, especialmente cuando las características tienen más de dos valores posibles y están basadas en frecuencias.
- **ComplementNB:** Variante de MultinomialNB que mejora el rendimiento en conjuntos de datos desbalanceados, ajustando las probabilidades de las clases complementarias para compensar el sesgo.
- **GaussianNB:** Asume que las características siguen una distribución normal (gaussiana) continua, siendo útil cuando las variables son continuas y tienen una forma aproximadamente normal.

## 3. Ejemplo Práctico

Para entender el funcionamiento del clasificador bayesiano ingenuo, usemos un caso de ejemplo: queremos construir un clasificador que prediga si un estudiante va a aprobar un examen, dado:

- **Tiempo de estudio:** Bajo, Moderado o Alto
- **Método de estudio:** Lectura, Práctica, o Lectura y Práctica (L y P)
- **Puntuación en exámenes anteriores:** Bajo, Promedio o Alto

Tiempo de estudio	Método de estudio	Puntuación	Resultado
Bajo	Lectura	Bajo	Desaprobó
Bajo	Práctica	Alta	Aprobó
Moderado	Lectura y Práctica	Promedio	Aprobó
Alto	Lectura y Práctica	Alta	Aprobó
Alto	Lectura	Alta	Desaprobó
Bajo	Lectura y Práctica	Baja	Desaprobó
Alto	Práctica	Alta	Aprobó
Moderado	Lectura	Alta	Aprobó
Moderado	Lectura y Práctica	Promedio	Aprobó
Moderado	Práctica	Bajo	Desaprobó

### 3.1. Tablas de Frecuencia

A partir de los datos, construimos las tablas de frecuencia para cada atributo:

#### Tiempo de Estudio:

Tiempo de estudio	Aprobó	Desaprobó	Total
Bajo	1	2	3
Moderado	3	1	4
Alto	2	1	3
Total	6	4	10

#### Método de Estudio:

Método de estudio	Aprobó	Desaprobó	Total
Lectura	1	2	3
Práctica	2	1	3
L y P	3	1	4
Total	6	4	10

#### Puntuación en exámenes anteriores:

Puntuación	Aprobó	Desaprobó	Total
Bajo	0	3	3
Promedio	2	0	2
Alto	4	1	5
Total	6	4	10

### 3.2. Aplicación del Teorema de Bayes

El clasificador bayesiano ingenuo asume que los atributos son independientes entre sí, por lo que podemos multiplicar las probabilidades. Para un nuevo estudiante, calculamos:

$$P(\text{Aprobó}) = \frac{6}{10} = 0,6$$

$$P(\text{Desaprobó}) = \frac{4}{10} = 0,4$$

Para clasificar un nuevo caso, multiplicamos las probabilidades condicionales por la probabilidad a priori. Por ejemplo, para un estudiante con:

- Tiempo de estudio: Alto
- Método: Lectura y Práctica
- Puntuación previa: Alta

$$\begin{aligned}
 P(\text{Aprobó}|X) &= P(\text{Aprobó}) \times P(\text{Alto}|\text{Aprobó}) \\
 &\quad \times P(\text{L y P}|\text{Aprobó}) \times P(\text{Alta}|\text{Aprobó}) \\
 &= 0,6 \times \frac{2}{6} \times \frac{3}{6} \times \frac{4}{6} \\
 &= 0,13
 \end{aligned}$$

$$\begin{aligned}
P(\text{Desaprobó}|X) &= P(\text{Desaprobó}) \times P(\text{Alto}|\text{Desaprobó}) \\
&\quad \times P(\text{L y P}|\text{Desaprobó}) \times P(\text{Alta}|\text{Desaprobó}) \\
&= 0,4 \times \frac{1}{4} \times \frac{1}{4} \times \frac{1}{4} \\
&= 0,006
\end{aligned}$$

Como  $0,13 > 0,006$ , el clasificador predice que el estudiante aprobará el examen.

## 4. Ventajas y Desventajas

### Ventajas

- Simple y rápido de entrenar
- Funciona bien con conjuntos de datos pequeños
- Maneja bien datos faltantes
- Fácil de interpretar

### Desventajas

- La suposición de independencia raramente se cumple en la realidad
- Sensible a características irrelevantes
- No captura relaciones complejas entre características
- Puede tener problemas con probabilidades cero (requiere suavizado)

## 5. Aplicaciones Comunes

- **Clasificación de spam:** Identificación de mails no deseados basada en el contenido y características del mensaje.
- **Análisis de sentimientos:** Clasificación de opiniones y comentarios en redes sociales, reseñas de productos o noticias como positivos, negativos o neutrales.

- **Diagnóstico médico:** Ayuda en la identificación de enfermedades basándose en síntomas y resultados de pruebas médicas.
- **Detección de fraude:** Identificación de transacciones sospechosas o actividades fraudulentas en sistemas financieros.
- **Clasificación de malware:** Identificación de software malicioso basado en sus características y comportamiento.

## 6. Suavizado de Laplace

Un problema común en Naive Bayes es cuando una probabilidad condicional es cero, lo que hace que toda la multiplicación sea cero. Para evitar esto, se utiliza el suavizado de Laplace (o suavizado aditivo), que agrega un pequeño valor  $\alpha$  (típicamente 1) a todos los conteos. Los conteos representan la frecuencia con la que aparece cada valor de una característica para una clase específica.

$$P(x_i|y) = \frac{\text{count}(x_i, y) + \alpha}{\text{count}(y) + \alpha \times n}$$

Por ejemplo, si tenemos:

- $\alpha = 1$  (suavizado de Laplace)
- $\text{count}(x_i, y) = 0$  (la palabra nunca apareció en la clase)
- $\text{count}(y) = 10$  (total de documentos en la clase)
- $n = 1000$  (cantidad de características)

Entonces:

$$P(x_i|y) = \frac{0 + 1}{10 + 1 \times 1000} = \frac{1}{1010} \approx 0,001$$

Con esto, se evita que una probabilidad condicional sea cero, haciendo que toda la multiplicación sea cero.

## 7. Implementación en Python

```
1 from sklearn.naive_bayes import GaussianNB
2 from sklearn.naive_bayes import MultinomialNB
3 from sklearn.naive_bayes import BernoulliNB
4
5 # Para datos continuos (distribucion normal)
6 model = GaussianNB()
7
8 # Para datos discretos (conteos)
9 model = MultinomialNB()
10
11 # Para datos binarios (presencia/ausencia)
12 model = BernoulliNB()
13
14 # Para datos categoricos
15 model = CategoricalNB()
16
17 # Para datos desbalanceados
18 model = ComplementNB()
19
20 # Entrenamiento
21 model.fit(X_train, y_train)
22
23 # Prediccion
24 y_pred = model.predict(X_test)
```