### **1. GaussianNB:**

* **Cuándo usarlo:** Para datos continuos que siguen una distribución normal.
* **Ejemplo:** Clasificación de alturas o pesos.

### **2. MultinomialNB:**

* **Cuándo usarlo:** Para datos discretos basados en conteos o frecuencias.
* **Ejemplo:** Clasificación de texto (spam/no spam).

### **3. BernoulliNB:**

* **Cuándo usarlo:** Para datos binarios (0 o 1), indicando presencia/ausencia.
* **Ejemplo:** Clasificación de correos según presencia de palabras clave.

## GaussianNB:

GaussianNB (para datos continuos)

**from sklearn.naive\_bayes import GaussianNB**

**from sklearn.model\_selection import train\_test\_split**

**from sklearn.metrics import accuracy\_score**

# Datos de ejemplo

**X = artists\_cartelera.drop(['top'], axis=1)**

**y = artists\_cartelera['top']**

# División en datos de entrenamiento y prueba

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.25, random\_state=42)**

# Crear el modelo GaussianNB

**model = GaussianNB(priors=[0.4, 0.6], var\_smoothing=1e-9)**

# Ajustar el modelo

**model.fit(X\_train, y\_train)**

# Predecir con el modelo

**y\_pred = model.predict(X\_test)**

## 

## MultinomialNB (para datos categóricos)

**from sklearn.naive\_bayes import MultinomialNB**

**from sklearn.model\_selection import train\_test\_split**

**from sklearn.metrics import accuracy\_score**

# Datos de ejemplo (características discretas)

**X = [[2, 3], [1, 1], [3, 5], [4, 5]] # Características**

**y = [0, 1, 0, 1] # Etiquetas (0 o 1)**

# División en datos de entrenamiento y prueba

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.25, random\_state=42)**

# Crear el modelo MultinomialNB

**model = MultinomialNB(alpha=1.0, fit\_prior=True, class\_prior=None)**

# Ajustar el modelo

**model.fit(X\_train, y\_train)**

# Predecir con el modelo

**y\_pred = model.predict(X\_test)**

## BernoulliNB (para datos binarios)

**from sklearn.naive\_bayes import BernoulliNB**

**from sklearn.model\_selection import train\_test\_split**

**from sklearn.metrics import accuracy\_score**

# Datos de ejemplo (características binarias)

**X = [[1, 0], [0, 1], [1, 1], [0, 0]] # Características**

**y = [0, 1, 0, 1] # Etiquetas (0 o 1)**

# División en datos de entrenamiento y prueba

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.25, random\_state=42)**

# Crear el modelo BernoulliNB

**model = BernoulliNB(alpha=1.0, binarize=0.0, fit\_prior=True, class\_prior=None)**

# Ajustar el modelo

**model.fit(X\_train, y\_train)**

# Predecir con el modelo

**y\_pred = model.predict(X\_test)**