

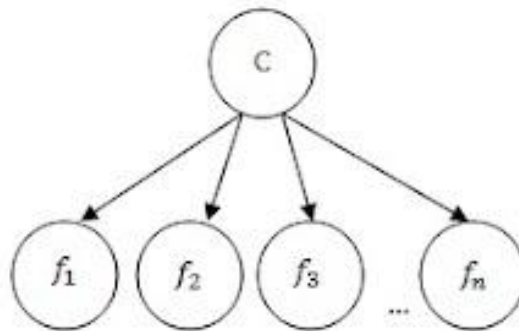


**INSTITUTO TECNOLÓGICO BELTRÁN**  
Centro de Tecnología e Innovación

Medios Comunicación

# Trabajo Práctico N.º 5

## Clasificador Bayesiano con Python



Nombre: Coral Tolazzi

Tema: Árbol de Decisión. Empresa de Telecomunicaciones.

Profesora: Yanina Ximena Scudero

Cuatrimestre y Año: 2 Cuatrimestre del 2025

Instituto tecnológico Beltrán  
TSCIA

**Contexto:**

Un medio de comunicación quiere implementar un sistema automático para detectar si una noticia es falsa o real, basándose en su contenido textual.

**Objetivo:**

Entrenar un modelo Naive Bayes con un conjunto de noticias etiquetadas, evaluar su rendimiento y usarlo para clasificar nuevas noticias.

**Datos**

Texto de la noticia	Etiqueta
El presidente anunció una nueva reforma educativa	real
Descubren que la vacuna convierte a las personas en robots	fake
La NASA confirma el hallazgo de agua en Marte	real
Científicos afirman que la Tierra es plana	fake
El ministerio de salud lanza campaña contra el dengue	real
Celebridades usan crema milagrosa para rejuvenecer 30 años	fake
Se inaugura el nuevo hospital en la ciudad	real
Estudio revela que comer chocolate cura el cáncer	fake
Gobierno aprueba ley de protección ambiental	real
Investigadores aseguran que los teléfonos espían nuestros sueños	fake

Clasificación de nuevos mensajes:

```
nuevas_noticias = [  
    "Nuevo estudio demuestra que el café mejora la memoria",  
    "Expertos afirman que los gatos pueden hablar con humanos"  
]
```

Realizar un informe del modelo.

- Resultados del modelo
  - Precisión total (accuracy):
  - Matriz de confusión:

Clasificación de nuevas noticias

1. "Nuevo estudio demuestra que el café mejora la memoria" → .....
2. "Expertos afirman que los gatos pueden hablar con humanos" → .....

## Descripción

Este trabajo consiste en la implementación de un clasificador automático de noticias utilizando el modelo de **Naive Bayes Multinomial**. El objetivo es detectar si una noticia es **real** o **falsa**, basándose en su contenido textual. Se emplea el procesamiento de texto con **CountVectorizer** y la librería **scikit-learn** para entrenar y evaluar el modelo.

## Funcionalidades

- **Preprocesamiento de texto:**  
Conversión de documentos a vectores de frecuencia mediante CountVectorizer.
- **División de datos:**  
Separación de las noticias en conjunto de entrenamiento y prueba con train\_test\_split.
- **Entrenamiento del modelo:**  
Uso del clasificador **Multinomial Naive Bayes**.
- **Evaluación del modelo:**
  - Cálculo de la precisión (accuracy).
  - Matriz de confusión para observar los aciertos y errores.
- **Clasificación de nuevas noticias:**  
Predicción de etiquetas para ejemplos no vistos.

## Datos de Entrenamiento

Noticias reales y falsas relacionadas con política, ciencia y salud.

Ejemplo del corpus:

- Real: *"El presidente anunció una nueva reforma educativa"*
- Fake: *"Científicos afirman que la Tierra es plana"*
- Real: *"Gobierno aprueba ley de protección ambiental"*
- Fake: *"Investigadores aseguran que los teléfonos espían nuestros sueños"*

## Resultados del Modelo

- **Precisión total (accuracy):** 1.00 (100%)
- **Matriz de confusión (real, fake):**  
( 1 & 0 )  
( 0 & 1 )

## Clasificación de Nuevas Noticias

1. *"Nuevo estudio demuestra que el café mejora la memoria"* → **real**
2. *"Expertos afirman que los gatos pueden hablar con humanos"* → **fake**

## Conclusión

Este trabajo práctico permitió implementar y evaluar un sistema de clasificación de noticias utilizando **Naive Bayes**, un algoritmo probabilístico sencillo pero efectivo. Se logró un 100% de precisión en el conjunto de prueba, lo que demuestra que el modelo diferencia correctamente entre noticias reales y falsas dentro del corpus reducido. Además, se puso en práctica el uso de CountVectorizer para la representación del texto y se reforzaron los conceptos de clasificación supervisada en Procesamiento del Lenguaje Natural.