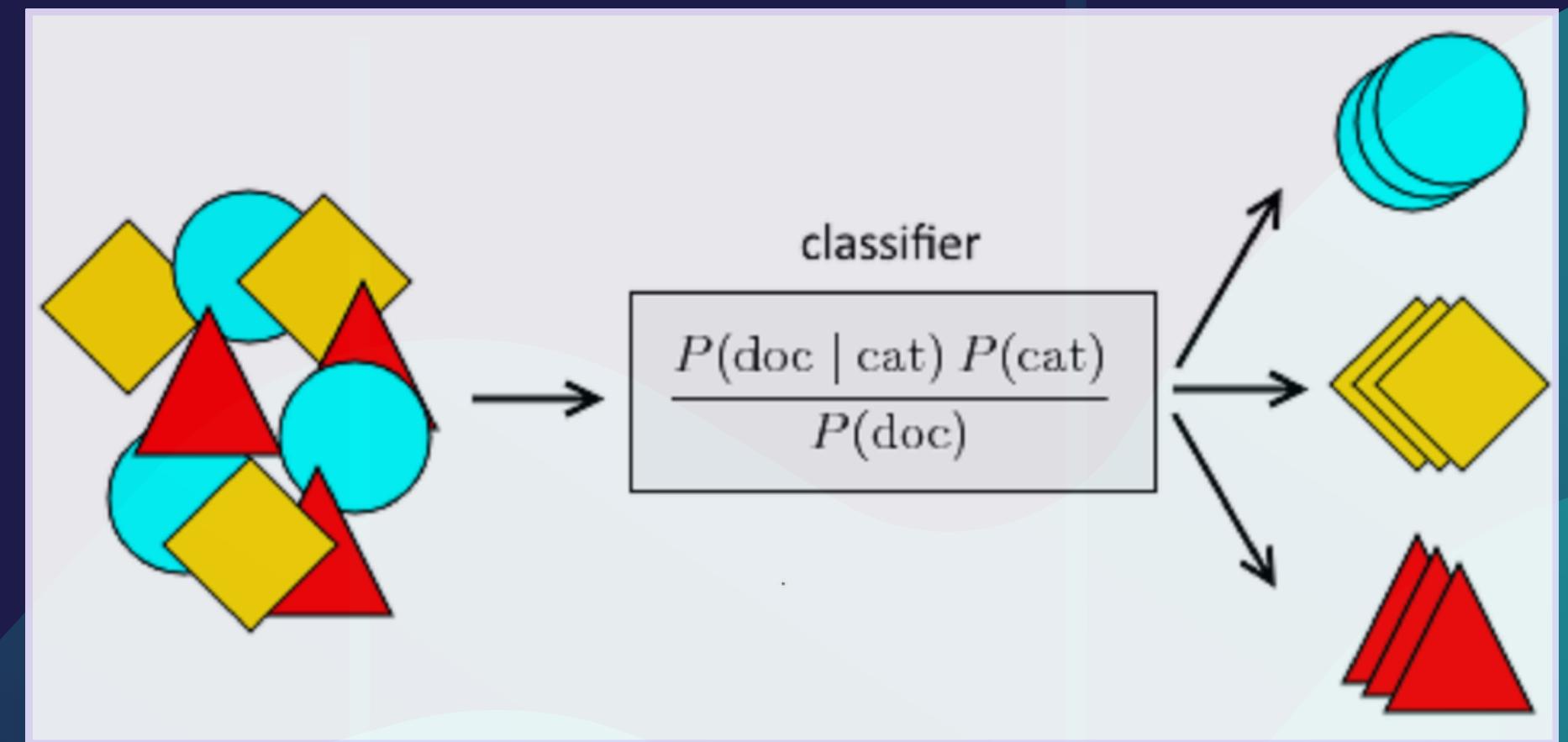
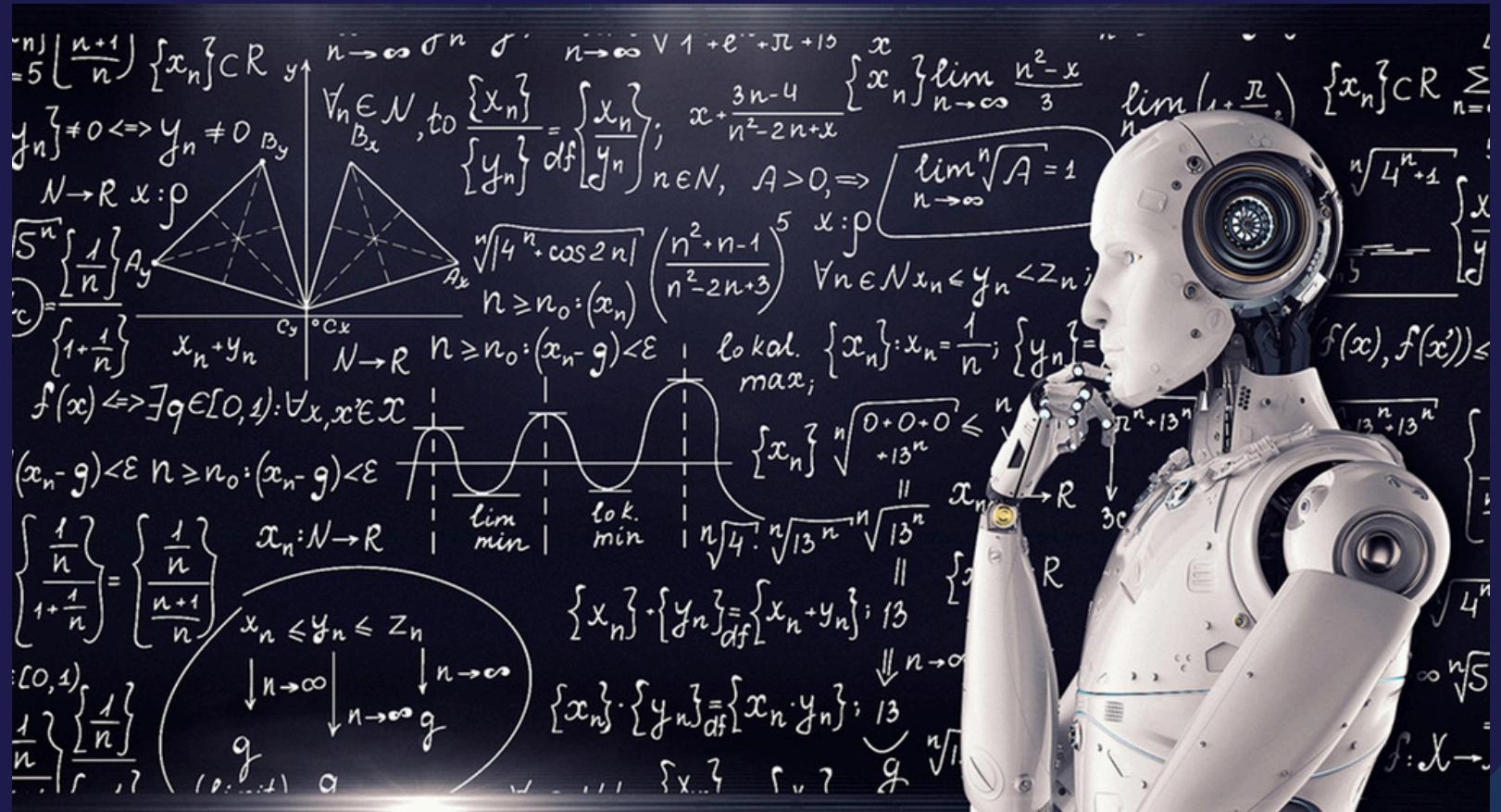


Naive Bayes



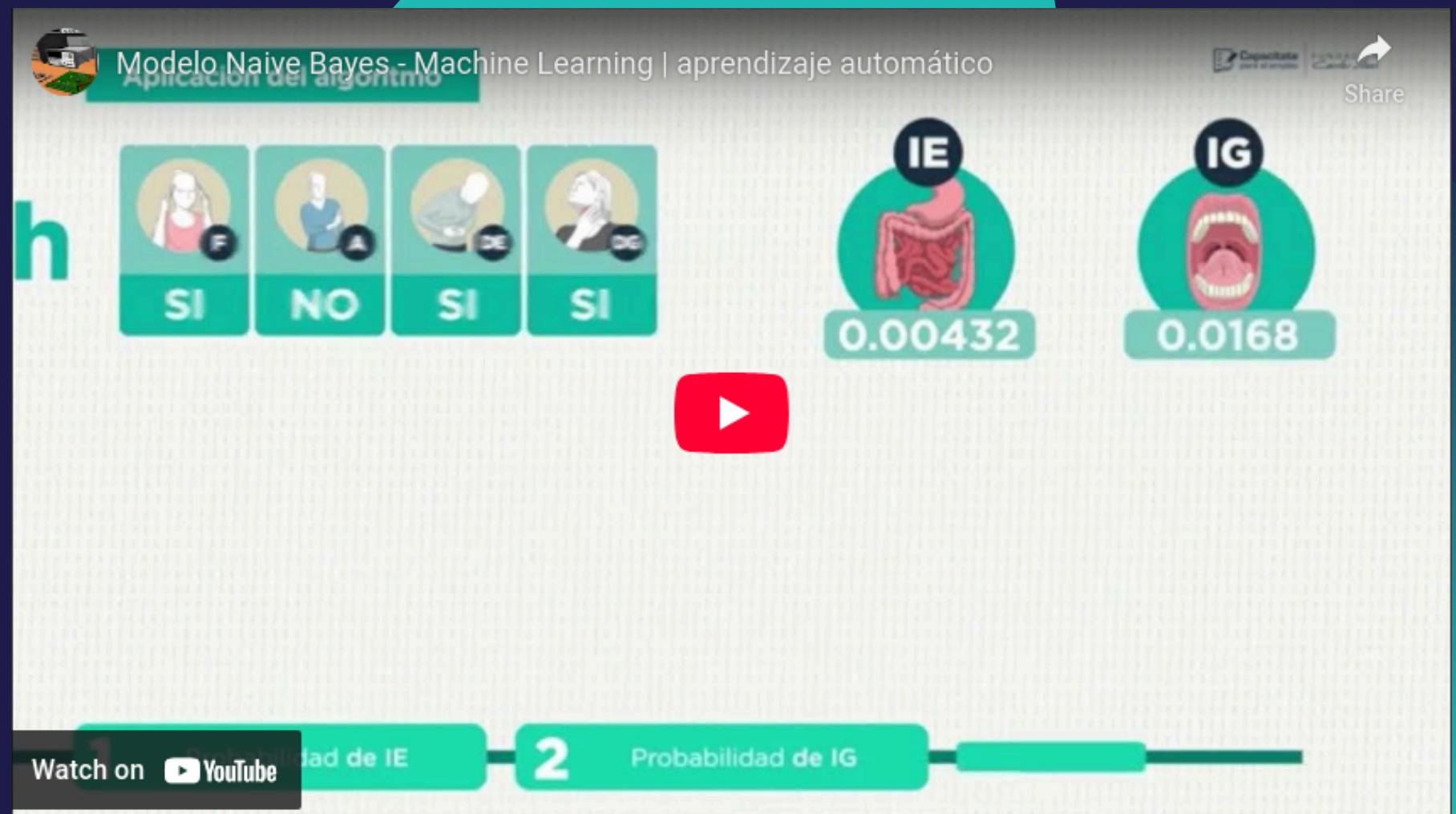
Agenda

- ✓ Naïve Bayes como Clasificador
- ✓ Clasificacion Basada en Modelos
- ✓ Ejemplos de clasificacion y aplicaciones
- ✓ Algoritmo Naïve Bayes
- ✓ Maximum likelihood
- ✓ Over-fitting
- ✓ K-fold
- ✓ Laplace Smoothing
- ✓ Suma de Logaritmos

Naïve Bayes

Es una clase especial de algoritmo de clasificación de Aprendizaje Automatico, o Machine Learning

Proporcionan una manera fácil de construir modelos con un comportamiento muy bueno debido a su simplicidad



Naïve Bayes

El principio de parsimonia o navaja de Ockham en el contexto de Machine Learning y clasificadores como Naive Bayes, se refleja en la preferencia por modelos simples que generalicen bien en lugar de modelos complejos que puedan sobreajustarse a los datos.

En este caso se asume por simplificación que:

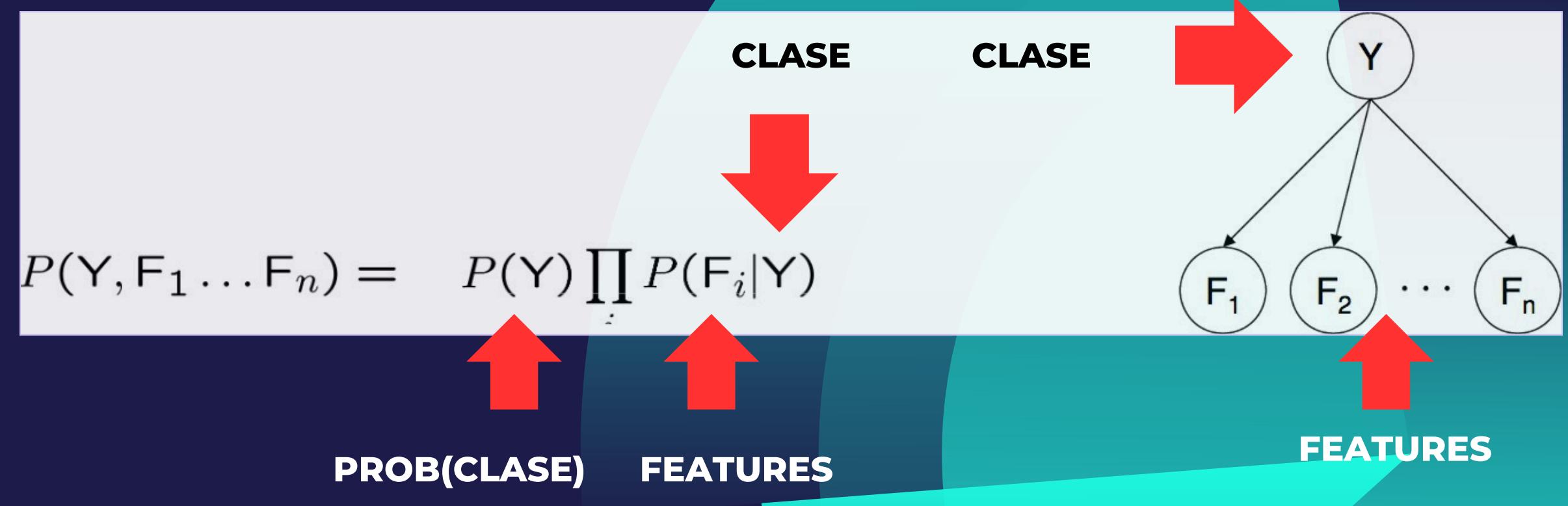
Las características (features) son independientes entre sí, dado el resultado (clase).

Principio de simplicidad o parsimonia:
"En igualdad de condiciones, la explicación más sencilla, suele ser la correcta".



Naïve Bayes

Naïve Bayes, asume una red bayesiana donde los atributos (*features*) son independientes entre si, pero las features dependen de la clase



- La **clase** representa la clasificación a la que puede pertenecer una instancia
- La **feature** representa alguna de las características o parámetros que se tienen de esa instancia.

ES COMO UN CHEF QUE USA UNA RECETA (EL TEOREMA DE BAYES) PARA COMBINAR INGREDIENTES (CARACTERÍSTICAS) Y HACER UN PLATO (LA CLASIFICACIÓN)

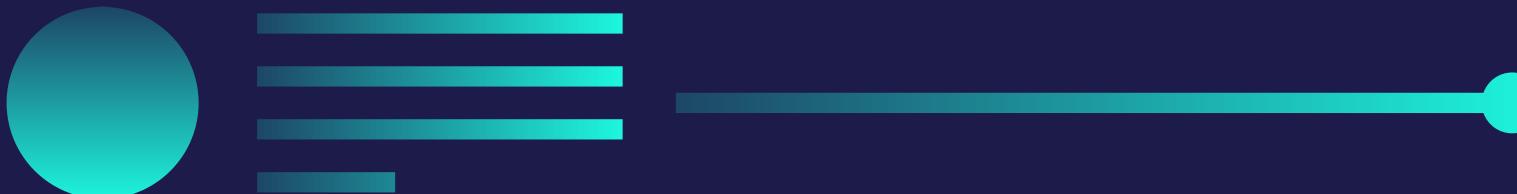
Clasificacion basada en Modelos





HASTA AHORA

*¿Como hemos modelado
sistemas con redes
bayesianas?*



Modelo



Variables

Definir parámetros o variables (evidencia, consulta, ocultas)



Estructura

Definir estructura mediante una Red bayesiana (BN) y Grafos



Probabilidades

Se usan las reglas de probabilidad condicional y el teorema de Bayes



Inferencia

Una vez modelada la red, se pueden responder las preguntas consultadas

Clasificación

Consiste en asignar una etiqueta o clase a una muestra de datos en función de sus características.

Los clasificadores se entrena con datos etiquetados para aprender a predecir la clase de nuevas muestras.

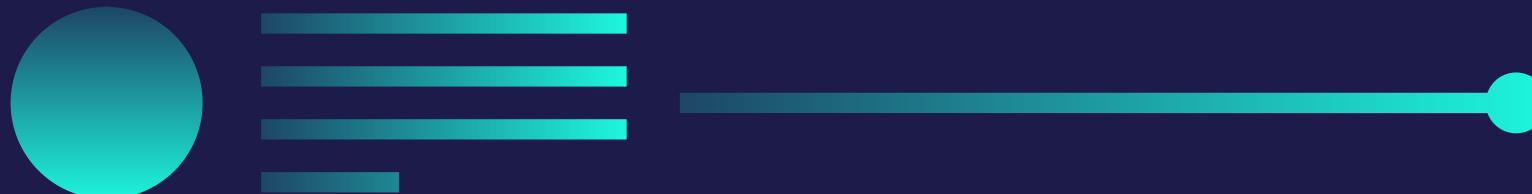
Para clasificar necesitamos:

1. Recopilación de Datos
2. Preprocesamiento de Datos
 - a. Limpiar los datos
 - b. Escalar o normalizar las características
3. División del Conjunto de Datos
 - a. Conjunto de entrenamiento (70%)
 - b. Conjunto de prueba (15%)
 - c. Conjunto de validacion (15%)
4. Selección del Modelo (Naive Bayes)
5. Entrenamiento del Modelo
6. Evaluación del Modelo



PERO...

*¿Cómo evaluarían si un
modelo es bueno o malo
para clasificar?*



¿Qué son las métricas de evaluación?



Definición

Herramientas para medir el rendimiento de un modelo.



Generalización

Un modelo con alta precisión en los datos de entrenamiento no necesariamente generaliza bien.



Decidir

Permiten comparar diferentes modelos y elegir el mejor.



Guían

Informan qué aspectos del modelo deben ajustarse para optimizar su rendimiento.

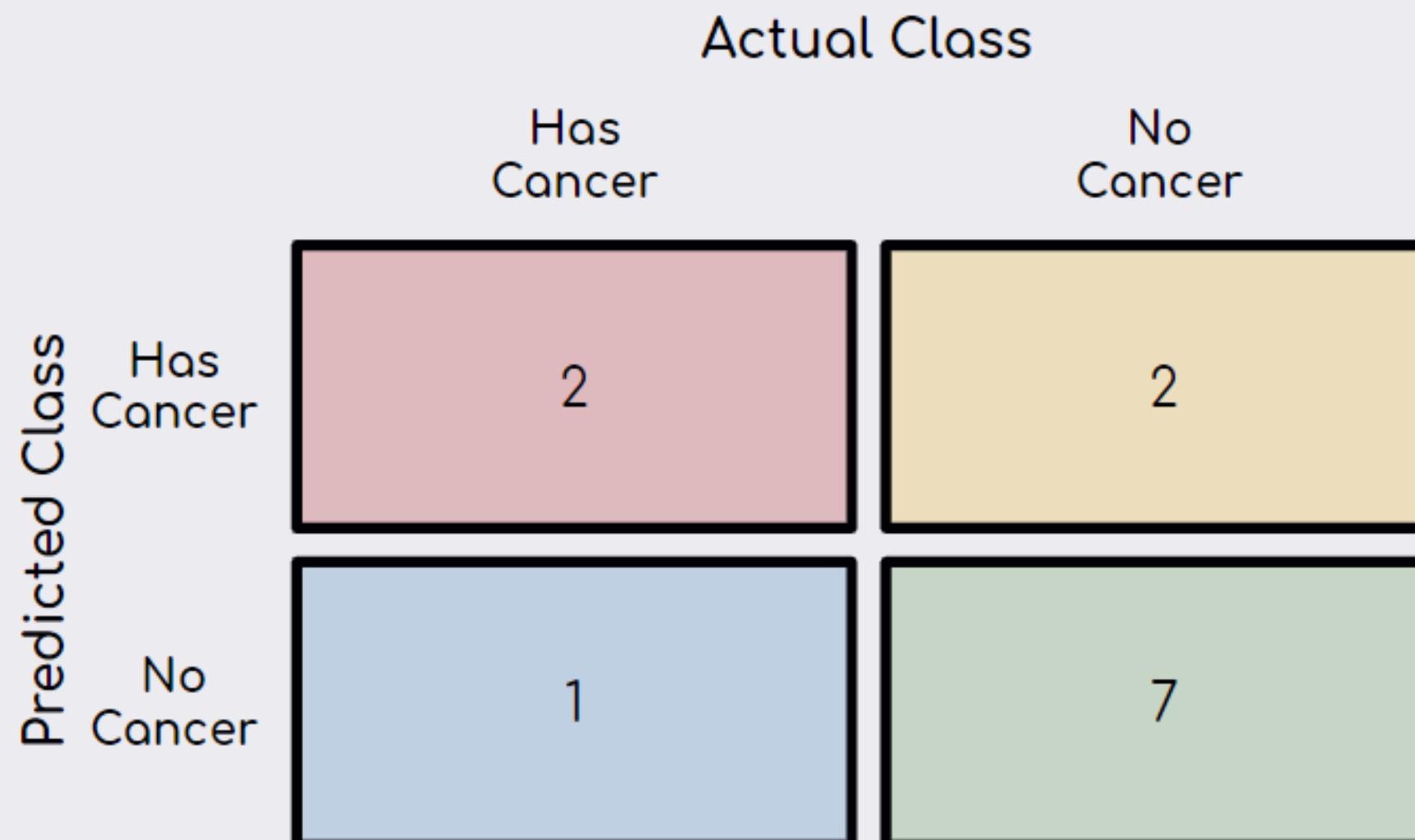
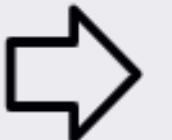
Matriz de Confusión

Una tabla que muestra las predicciones del modelo versus las clases reales.

		Valor Actual o Real	
		Positivo	Negativo
Valor Predicho	Positivo		
	Negativo		

Matriz de Confusión

Predicted Class	Actual Class
No Cancer	No Cancer
Has Cancer	Has Cancer
No Cancer	No Cancer
No Cancer	No Cancer
No Cancer	Has Cancer
Has Cancer	Has Cancer
No Cancer	No Cancer
Has Cancer	No Cancer
No Cancer	No Cancer
No Cancer	No Cancer
Has Cancer	No Cancer
No Cancer	No Cancer



Evaluación del Modelo

Usar métricas como:

- **Exactitud (Accuracy)**: mide qué tan bien el modelo clasifica correctamente

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- TP (True Positive): Clasificado correctamente como positivo.
- TN (True Negative): Clasificado correctamente como negativo.
- FP (False Positive): Clasificado incorrectamente como positivo (falso positivo).
- FN (False Negative): Clasificado incorrectamente como negativo (falso negativo).

- **Precisión**: mide la proporción de ejemplos positivos predichos que realmente son positivos.

$$\text{Precision} = \frac{TP}{TP + FP}$$

¿Qué métrica usarías si te importa más evitar falsos positivos?

Evaluación del Modelo

Ejemplo comparativo

Tenemos un modelo de clasificación que predice si un tumor es maligno o benigno.

- Maligno: Tumor peligroso.
- Benigno: Tumor no peligroso.

Supongamos que tenemos 1000 tumores y el modelo realiza las siguientes predicciones:

1000 casos	Predictión: Maligno	Predictión: Benigno
Real: Maligno	30	70
Real: Benigno	20	880

Clasificación

Ejemplo comparativo

El **Accuracy** mide qué tan bien el modelo clasifica correctamente en general, tanto malignos como benignos.

$$\text{Accuracy} = \frac{70 + 880}{70 + 880 + 20 + 30} = \frac{950}{1000} = 0.95 \quad (95\%)$$

El modelo acierta en el 95% de los casos totales.

Si el conjunto de datos está muy desbalanceado (por ejemplo, 950 benignos y 50 malignos), el modelo puede lograr un accuracy alto simplemente prediciendo todo como benigno, por tanto,

puede ser engañoso

Clasificación

Ejemplo comparativo

La **Precision** mide qué porcentaje de los tumores clasificados como malignos realmente lo son.

$$\text{Precision} = \frac{70}{70 + 20} = \frac{70}{90} = 0.78 \quad (78\%)$$

De todos los tumores que el modelo clasificó como malignos, el 78% realmente lo eran.

Evaluación del Modelo

Usar métricas como:

- **Recall (Sensibilidad o Tasa de Verdaderos Positivos)**: mide la capacidad del modelo para

detectar correctamente los casos positivos (malignos).

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Recall} = \frac{70}{70 + 30} = \frac{70}{100} = 0.70 \quad (70\%)$$

El modelo identifica correctamente el 70% de los tumores malignos

- **F1-Score**: combina la **Precision** y el **Recall** en una única métrica para equilibrar ambos aspectos.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{F1-Score} = 2 \times \frac{0.78 \times 0.70}{0.78 + 0.70} = 2 \times \frac{0.546}{1.48} = 0.74 \quad (74\%)$$

Indica que el modelo logra un equilibrio razonable entre precisión y sensibilidad, útil en casos de clases desbalanceadas

Evaluación del Modelo

		Predicted Class		Sensitivity $\frac{TP}{(TP + FN)}$	Specificity $\frac{TN}{(TN + FP)}$	Accuracy $\frac{TP + TN}{(TN + FP + FP + FN)}$
		Positive	Negative			
Actual Class	Positive	True Positive (TP)	False Negative (FN) Type II Error			
	Negative	False Positive (FP) Type I Error	True Negative (TN)			
	Precision	$\frac{TP}{(TP + FP)}$	Negative Predictive Value	$\frac{TN}{(TN + FN)}$		

Evaluación del Modelo

¿Qué métrica usarías si te importa más evitar falsos positivos?

La **precisión** mide qué proporción de las predicciones positivas realizadas por el modelo son correctas. Penaliza los falsos positivos, ya que estos afectan directamente su valor.

- **Ejemplo:** En un sistema de detección de fraudes, es importante minimizar los falsos positivos para no bloquear transacciones legítimas.

¿Qué métrica priorizarías en un sistema médico donde es crítico detectar todas las enfermedades?

El **recall** mide qué proporción de los casos positivos reales son correctamente identificados por el modelo. En un sistema médico, detectar todas las enfermedades (minimizar los falsos negativos) es crucial, incluso si eso implica aceptar algunos falsos positivos.

- **Ejemplo:** En la detección de cáncer, es preferible que el modelo clasifique erróneamente algunos casos como positivos (aunque no lo sean) a que pase por alto un caso real de cáncer.

Resumen

Entonces:

Accuracy

Útil cuando las clases están equilibradas

Precision

Importante cuando el costo de un **falso positivo** es alto (como falsos diagnósticos positivos).

Recall

Importante cuando el costo de un **falso negativo** es alto (como pasar por alto un tumor maligno).

F1-Score

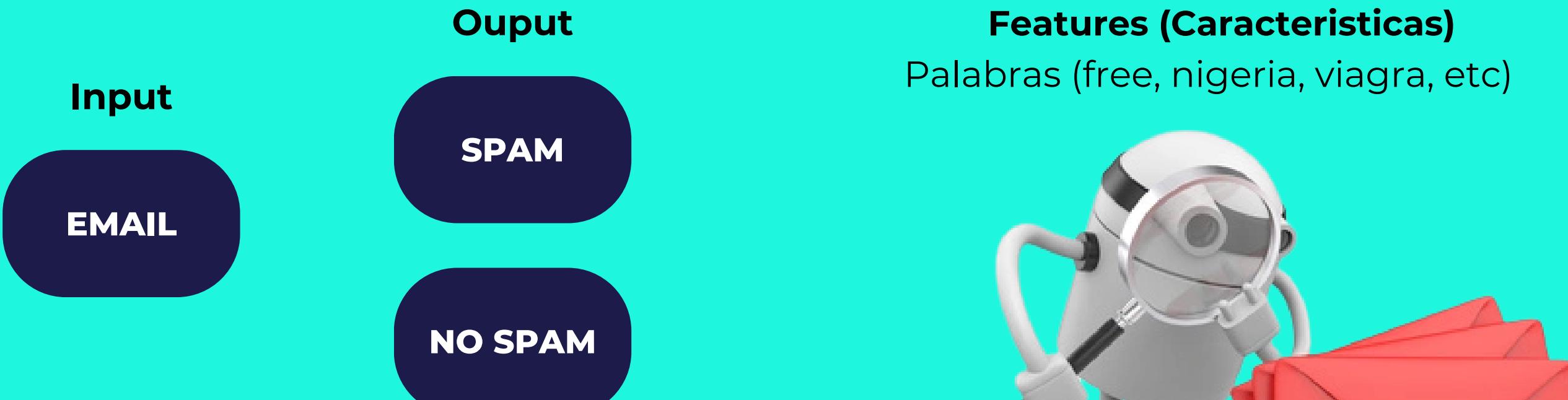
Útil cuando se necesita un equilibrio entre precisión y sensibilidad.

Ejemplo de clasificación



Ejemplo SPAM

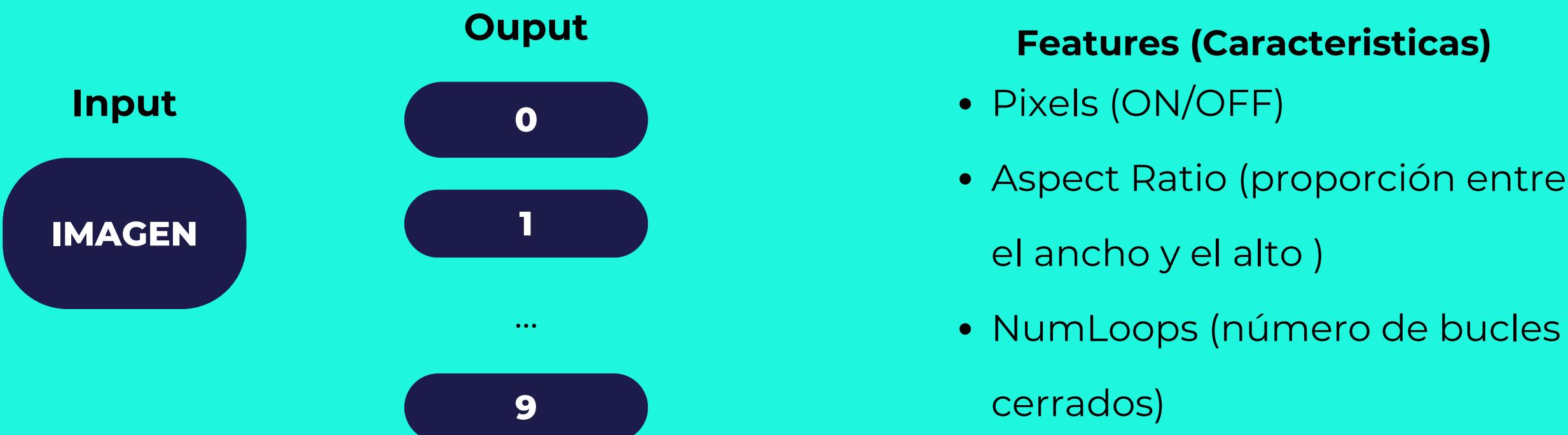
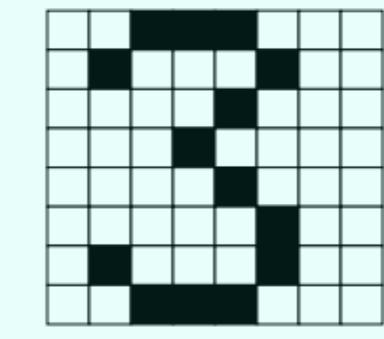
Supongamos que tenemos un correo electronico en el cual queremos identificar correo valido y correo SPAM



OBTENER UNA GRAN CANTIDAD DE EMAILS, CLASIFICAR CADA UNO DE ELLOS COMO SPAM O HAM, APRENDER DE LOS EMAILS PREVIAMENTE CLASIFICADOS

Ejemplo Reconocimiento de Números en Imágenes

Supongamos que queremos identificar números dentro de una imagen



OBTENER UNA GRAN CANTIDAD DE MUESTRAS DE NÚMEROS PREVIAMENTE CLASIFICADOS

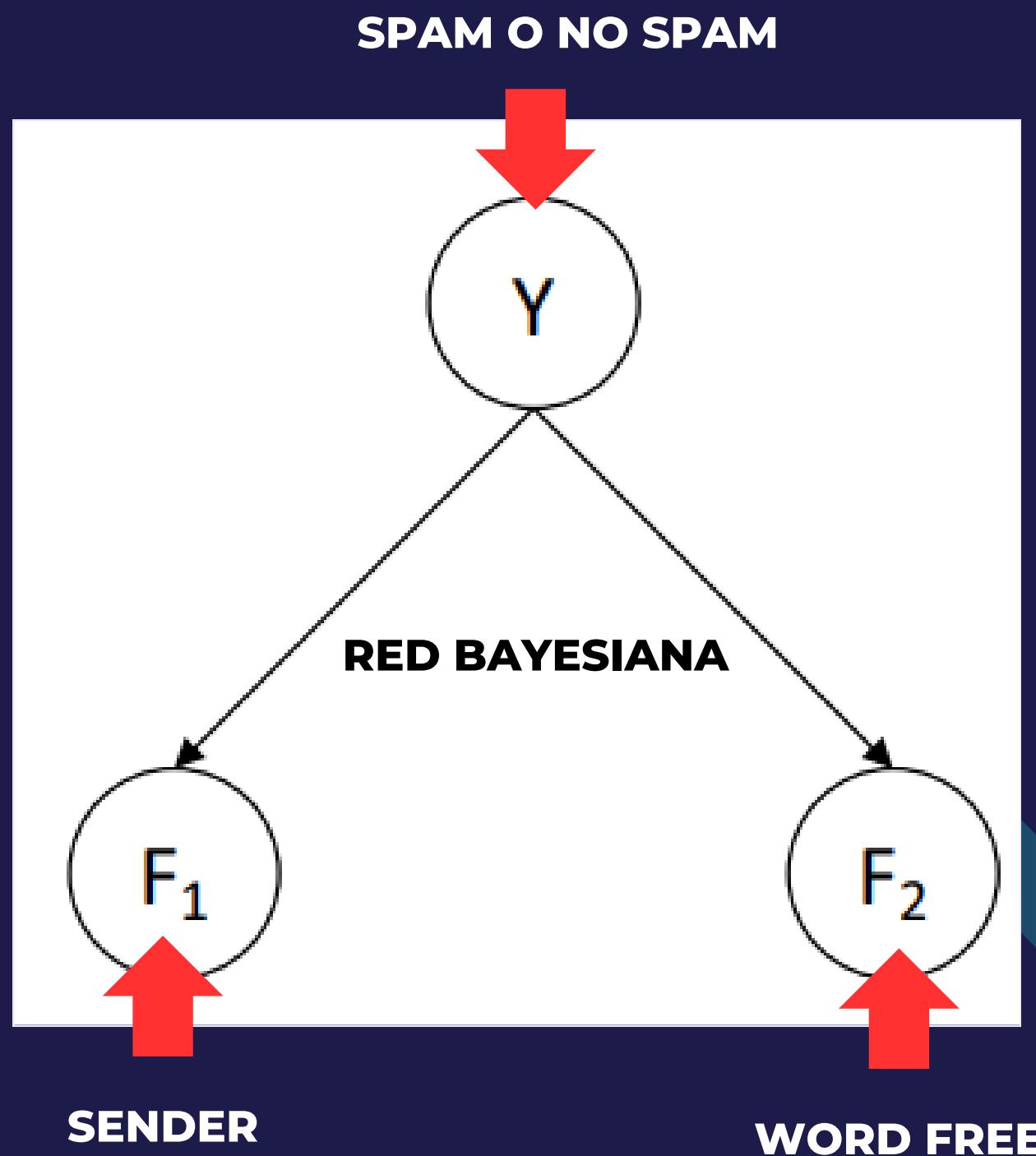
Aplicaciones de Clasificación

- ✓ Detección spam
- ✓ OCR
- ✓ Diagnósticos médicos
- ✓ Detección de fraudes
- ✓ Ruteo de quejas
- ✓ Reconocimiento de Voz
- ✓ Segmentación de Clientes en Marketing
- ✓ Clasificación de Noticias
- ✓ etc...

Algortimo Naïve Bayes



Naïve Bayes



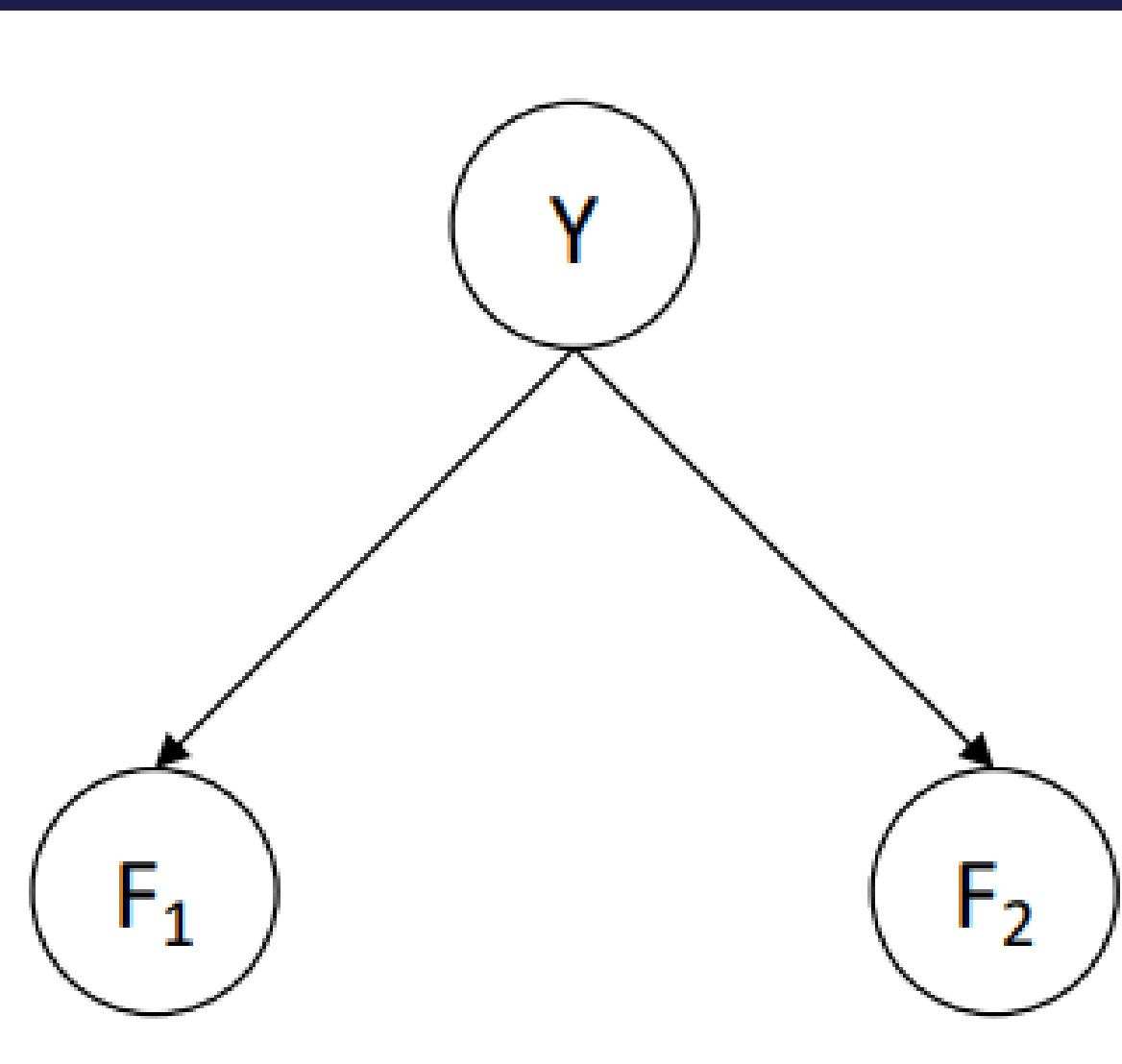
Para el correo electronico sabemos que hay dos clasificaciones esperadas: **SPAM y NO SPAM**

También para este ejemplo en los datos sabremos dos características:

- Quién envía el correo (F_1)
- Numero de veces que aparece la palabra *FREE* (F_2)

Consideramos ambas *features* independientes entre ellas

Naïve Bayes



F ₁ : A feature (do I know the sender?)		
F ₁	Y	P(F ₁ Y)
yes	ham	?
no	ham	?
yes	spam	?
no	spam	?

F ₂ : Another feature (# of occurrences of FREE)		
F ₂	Y	P(F ₂ Y)
0	ham	?
1	ham	?
2	ham	?
0	spam	?
1	spam	?
2	spam	?

Y: The label (spam or ham)	
Y	P(Y)
ham	?
spam	?

Naïve Bayes

DATOS DE ENTRENAMIENTO

Training Data		
#	Email Text	Label
1	Attached is my portfolio.	ham
2	Are you free for a meeting tomorrow?	ham
3	Free unlimited credit cards!!!!	spam
4	Mail \$10,000 check to this address	spam
5	Sign up now for 1 free Bitcoin	spam
6	Free money free money	spam

Total de correos: **6**

Veces que aparece ham: **2**

$$P(\text{ham}) = 2/6 = 1/3 = 0.3333$$

Veces que aparece *FREE* en un correo etiquetado como ham:

- 0 veces: **1** $P(\text{f2}|\text{ham}) = 1/2 = 0.5$
- 1 veces: **1** $P(\text{f2}|\text{ham}) = 1/2 = 0.5$
- 2 veces: **0** $P(\text{f2}|\text{ham}) = 0$

Naïve Bayes

DATOS DE ENTRENAMIENTO

Training Data		
#	Email Text	Label
1	Attached is my portfolio.	ham
2	Are you free for a meeting tomorrow?	ham
3	Free unlimited credit cards!!!!	spam
4	Mail \$10,000 check to this address	spam
5	Sign up now for 1 free Bitcoin	spam
6	Free money free money	spam

Total de correos: **6**

Veces que aparece spam: 4

$$P(\text{spam}) = 4/6 = 2/3 = 0.6666$$

Veces que aparece *FREE* en un correo etiquetado como spam:

- 0 veces: **1** $P(f2|\text{spam}) = 1/4 = 0.25$
- 1 veces: **2** $P(f2|\text{spam}) = 2/4 = 0.5$
- 2 veces: **1** $P(f2|\text{spam}) = 1/4 = 0.25$

Naïve Bayes

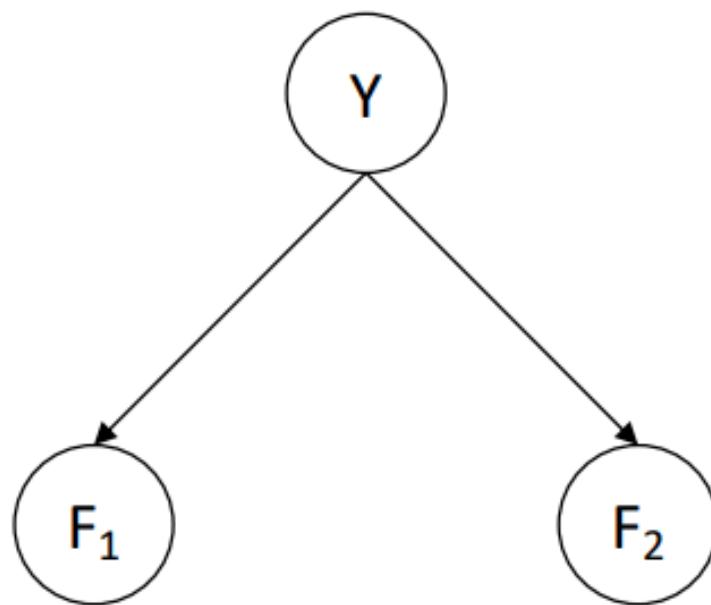
F_2 : # of occurrences of FREE

F_2	Y	$P(F_2 Y)$
0	ham	0.5
1	ham	0.5
2	ham	0.0
0	spam	0.25
1	spam	0.50
2	spam	0.25

Training Data

#	Email Text	Label
1	Attached is my portfolio.	ham
2	Are you free for a meeting tomorrow?	ham
3	Free unlimited credit cards!!!!	spam
4	Mail \$10,000 check to this address	spam
5	Sign up now for 1 free Bitcoin	spam
6	Free money free money	spam

Naïve Bayes



Y: The label (spam or ham)	
	P(Y)
ham	0.6
spam	0.4

F ₁ : A feature (do I know the sender?)		
F ₁	Y	P(F ₁ Y)
yes	ham	0.7
no	ham	0.3
yes	spam	0.1
no	spam	0.9

F ₂ : Another feature (# of occurrences of FREE)		
F ₂	Y	P(F ₂ Y)
0	ham	0.85
1	ham	0.07
2	ham	0.08
0	spam	0.75
1	spam	0.12
2	spam	0.13

Naïve Bayes

Suponga el mensaje conociendo la persona que envió (contacto):

“Free food in Soda 430 today”

Identifique si es spam o no spam

Sabemos que:

F1 = yes

F2 = 1

Probabilidad Conjunta

Que el correo sea Spam, sabiendo quien lo envio y que tiene la palabra Free una vez en el mensaje

$$P(Y = \text{spam}, F1 = \text{yes}, F2 = 1) = P(Y = \text{spam}) * P(F1 = \text{yes} | Y = \text{spam})$$

$$* P(F2 = 1 | Y = \text{spam}) = 0.4 * 0.1 * 0.12 = \mathbf{0.0048}$$

Que el correo sea No Spam, sabiendo quien lo envio y que tiene la palabra Free una vez en el mensaje

$$P(Y = \text{ham}, F1 = \text{yes}, F2 = 1) = P(Y = \text{ham}) * P(F1 = \text{yes} | Y = \text{ham}) *$$

$$P(F2 = 1 | Y = \text{ham}) = 0.6 * 0.7 * 0.07 = \mathbf{0.0294}$$

Naïve Bayes

Suponga el mensaje conociendo la persona que envió (contacto):

“Free food in Soda 430 today”

Identifique si es spam o no spam

Sabemos que:

F1 = yes

F2 = 1

Normalizar

$$P(Y = \text{spam} | F1 = \text{yes}, F2 = 1) = 0.0048 / (0.0048 + 0.0294) = \mathbf{0.14}$$

$$P(Y = \text{ham} | F1 = \text{yes}, F2 = 1) = 0.0294 / (0.0048 + 0.0294) = \mathbf{0.86}$$

14% de posibilidades de que el correo electrónico sea spam.

86% de posibilidades de que sea ham.

Maximum likelihood en Naïve Bayes



Maximum Likelihood

El algoritmo de Naive Bayes utiliza el principio de Máxima Verosimilitud (Maximum Likelihood) para estimar las probabilidades necesarias para realizar la clasificación.

El objetivo es encontrar la clase **C** que maximice la probabilidad posterior dado un conjunto de características **X=(x₁,x₂,...,x_n)**:

$$P(C | X) = \frac{P(X | C) \cdot P(C)}{P(X)}$$

Dado que P(X) es constante para todas las clases, basta con maximizar el numerador (o sea no necesitamos normalizar para identificar la clasificación, basta con encontrar el mayor valor en el numerador):

$$P(C | X) \propto P(X | C) \cdot P(C)$$

Over-fitting



Over-fitting

Ocurre cuando un modelo de machine learning se ajusta demasiado bien a los datos de entrenamiento, capturando tanto los patrones reales como el ruido o las anomalías presentes en el conjunto de datos.

Como resultado, el modelo funciona muy bien con los datos de entrenamiento pero falla al generalizar con nuevos datos

¿Por qué ocurre el overfitting?

- Modelo demasiado complejo
- Demasiadas características
- Pocos datos de entrenamiento
- Entrenamiento excesivo

¿Cómo detectar el overfitting?



- **Diferencia de rendimiento entre entrenamiento y prueba:**
 - Alta exactitud (accuracy) en entrenamiento (por ejemplo, 98%).
 - Baja exactitud (accuracy) en prueba (por ejemplo, 60%).
- **Curvas de aprendizaje:**
 - La precisión de entrenamiento sigue aumentando, pero la precisión de validación se estanca o disminuye.

¿Cómo evitar el overfitting?

Aunque el algoritmo de Naive Bayes generalmente es menos propenso al overfitting en comparación con modelos más complejos como redes neuronales, aún puede ocurrir en ciertos escenarios, para esto se puede aplicar las siguientes técnicas:

- **Suavizado de Laplace (Laplace Smoothing)**
- **Selección de características:** remover features irrelevantes o ruidosas que hagan aprender patrones falsos
- **Eliminación de características redundantes o correlacionadas** (features realmente dependientes)
- **Limpieza y preprocesamiento de datos:** Eliminar valores atípicos (outliers) y normalizar los datos puede reducir el riesgo.

¿Cómo evitar el overfitting?

- **Validación cruzada (Cross-Validation):** Realizar una validación cruzada, como **K-Fold***, ayuda a evaluar el modelo en diferentes subconjuntos de datos para detectar el overfitting de manera temprana.
- **Aumento de datos (Data Augmentation):** Si tienes pocos datos, el modelo puede memorizar ejemplos específicos. Generar ejemplos adicionales, especialmente en tareas de clasificación de imágenes o texto, puede ayudar a mejorar la generalización.
- **Priorización de clases balanceadas:** Si el conjunto de datos está muy desbalanceado, el modelo podría sobreajustarse a la clase mayoritaria.

K-Fold Cross-Validation



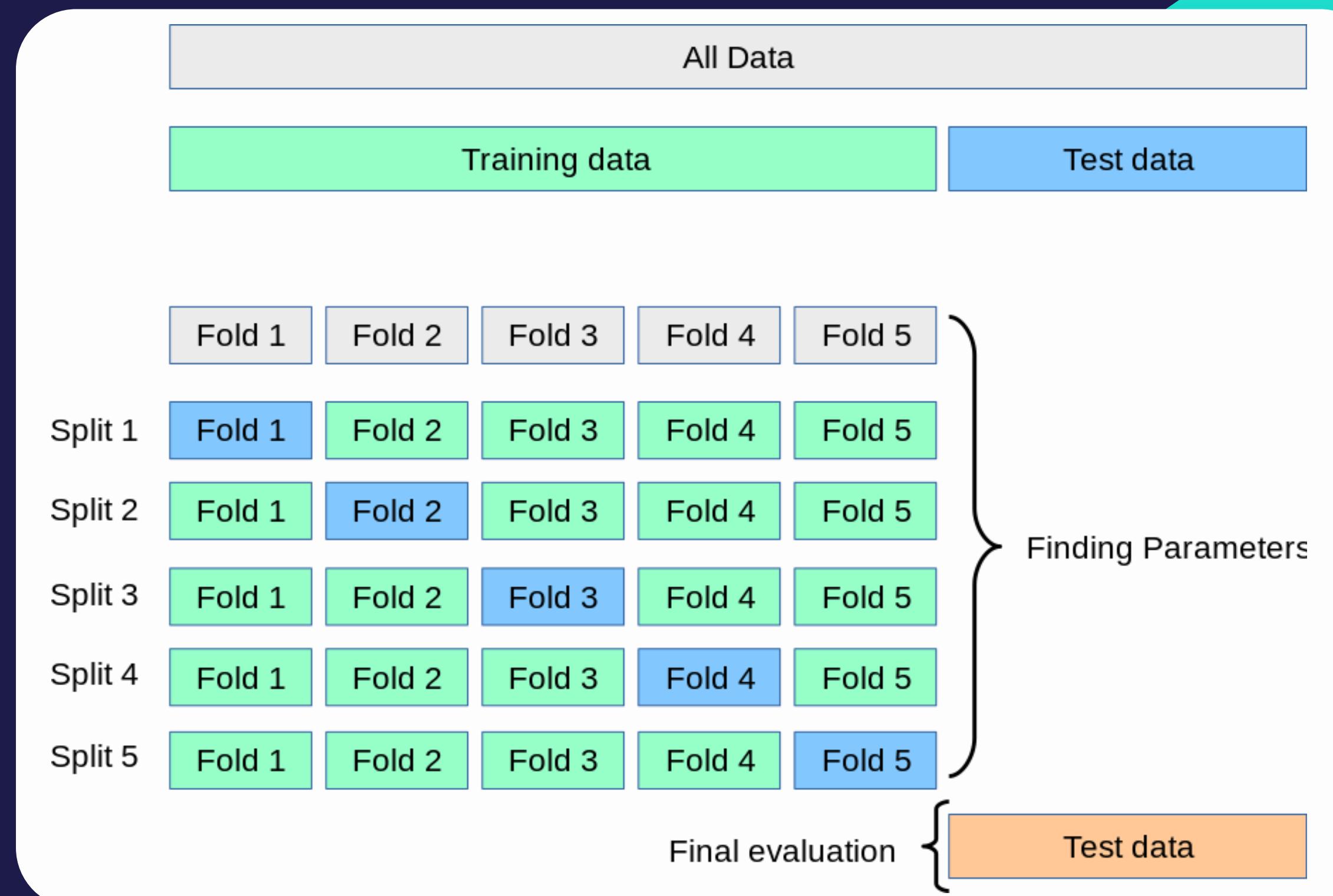
K-Fold Cross-Validation

Es una técnica utilizada para evaluar el rendimiento de un modelo de machine learning y evitar problemas como el overfitting. Su objetivo principal es asegurarse de que el modelo generalice bien a datos nuevos.

Pasos

1. **Dividir los datos en K grupos (o "folds") de tamaño aproximadamente igual.**
 - a. Por ejemplo, si se tienen 1000 datos y se elige K=5, cada grupo tendrá 200 datos
2. **Entrenamiento y prueba repetidos:**
 - a. Se realizan K iteraciones en total.
 - b. En cada iteración:
 - i. (K-1) folds se utilizan para entrenar el modelo.
 - ii. 1 fold se utiliza para probar el modelo.
3. **Promedio de resultados:**
 - a. Después de entrenar y evaluar el modelo K veces, se promedian las métricas (como precisión, recall, F1-score, etc.) para obtener una estimación más robusta del rendimiento.

K-Fold Cross-Validation



Laplace Smoothing



Laplace Smoothing

Es una técnica que se utiliza en el algoritmo Naive Bayes para evitar el problema de las probabilidades nulas

¿Cuál es el problema?

Sabemos que:

$$P(\text{clase} \mid \text{característica}) = \frac{P(\text{característica} \mid \text{clase}) \cdot P(\text{clase})}{P(\text{característica})}$$

El mayor problema surge al calcular $P(\text{característica} \mid \text{clase})$, ya que si alguna característica no aparece en una clase, el resultado será cero.

Si la **frecuencia** es cero, la probabilidad será cero, y eso hará que toda la probabilidad de la clase también sea cero.

Laplace Smoothing

El suavizado de Laplace añade un valor constante (generalmente 1) a cada recuento para asegurarse de que ninguna probabilidad sea cero.

$$P(\text{característica} \mid \text{clase}) = \frac{\text{Frecuencia}(\text{característica} \mid \text{clase}) + 1}{\text{Total de características en la clase} + V}$$

- V es el número total de características posibles.
- El valor "1" garantiza que incluso las características no observadas tengan una pequeña probabilidad.

Ejemplo Bag of Words usando Laplace Smoothing

Supongamos que tenemos un conjunto de datos de correos electrónicos clasificados como spam y no spam.

Email	Spam (1) / No Spam (0)	Palabras en el email
"oferta exclusiva"	1	oferta, exclusiva
"gana dinero fácil"	1	gana, dinero, facil
"saludos cordiales"	0	saludos, cordiales
"reunión de trabajo"	0	reunion, de, trabajo

Ejemplo Bag of Words usando Laplace Smoothing

Contemos las frecuencias de las palabras en cada clase (spam y no spam)

Spam:

- "oferta": 1
- "exclusiva": 1
- "gana": 1
- "dinero": 1
- "fácil": 1

No Spam:

- "saludos": 1
- "cordiales": 1
- "reunión": 1
- "trabajo": 1

Total de palabras en Spam: 5

Total de palabras en No Spam: 4

Ejemplo Bag of Words usando Laplace Smoothing

Probabilidades sin suavizado

Palabra	Frecuencia	P(Palabra)
"oferta"	1	1/5
"exclusiva"	1	1/5
"gana"	1	1/5
"dinero"	1	1/5
"fácil"	1	1/5

Ejemplo Bag of Words usando Laplace Smoothing

Probabilidades sin suavizado

Ham	Palabra	Frecuencia	P(Palabra)
	"saludos"	1	1/4
	"cordiales"	1	1/4
	"reunión"	1	1/4
	"trabajo"	1	1/4

Vocabulario Total:

$$V = \{"oferta", "exclusiva", "gana", "dinero", "fácil", "saludos", "cordiales", "reunión", "trabajo"\}$$

9 palabras unicas

Ejemplo Bag of Words usando Laplace Smoothing

Aplicando Suavizado

$$P(\text{palabra} \mid \text{clase}) = \frac{\text{Frecuencia de la palabra} + 1}{\text{Total de palabras en la clase} + V}$$

Spam:

$$P(\text{"oferta"} \mid \text{spam}) = \frac{1 + 1}{5 + 9} = \frac{2}{14} \approx 0.1429$$

$$P(\text{"exclusiva"} \mid \text{spam}) = \frac{1 + 1}{5 + 9} = \frac{2}{14} \approx 0.1429$$

$$P(\text{"gana"} \mid \text{spam}) = \frac{1 + 1}{5 + 9} = \frac{2}{14} \approx 0.1429$$

No Spam:

$$P(\text{"saludos"} \mid \text{no spam}) = \frac{1 + 1}{4 + 9} = \frac{2}{13} \approx 0.1538$$

$$P(\text{"cordiales"} \mid \text{no spam}) = \frac{1 + 1}{4 + 9} = \frac{2}{13} \approx 0.1538$$

Laplace Smoothing

Con nuestro proceso previamente calculado, tenemos la probabilidades ahora, supongamos que queremos clasificar un nuevo email: “**gana dinero**”

Queremos saber si es spam o no spam, usando Naive Bayes:

$$P(\text{Spam} \mid \text{gana, dinero}) = P(\text{Spam}) \cdot P(\text{gana} \mid \text{Spam}) \cdot P(\text{dinero} \mid \text{Spam})$$

$$P(\text{Spam}) = \frac{2}{4} = 0.5$$

$$P(\text{gana} \mid \text{Spam}) = 0.1429$$

$$P(\text{dinero} \mid \text{Spam}) = 0.1429$$

$$P(\text{Spam} \mid \text{gana, dinero}) = 0.5 \cdot 0.1429 \cdot 0.1429 \approx 0.0102$$

$$P(\text{No Spam}) = \frac{2}{4} = 0.5$$

$$P(\text{gana} \mid \text{No Spam}) = \frac{1}{13} \approx 0.0769$$

$$P(\text{dinero} \mid \text{No Spam}) = \frac{1}{13} \approx 0.0769$$

$$P(\text{No Spam} \mid \text{gana, dinero}) = 0.5 \cdot 0.0769 \cdot 0.0769 \approx 0.00295$$

Por Maximum Likelihood podemos concluir $P(\text{Spam} \mid \text{gana, dinero}) > P(\text{No Spam} \mid \text{gana, dinero})$
por tanto podemos concluir que el correo es SPAM

Suma de Logaritmos



Suma de Logaritmos

Los números muy pequeños pueden desaparecer (es decir, ser redondeados a cero) o pueden causar errores de precisión, lo que hace que los cálculos sean inexactos, el valor final puede ser tan pequeño que se pierde la información.

Una técnica comúnmente utilizada para evitar estos problemas es trabajar con logaritmos de las probabilidades en lugar de las probabilidades directamente.

Normal

$$P(\text{clase} \mid \text{características}) = \frac{P(\text{característica}_1 \mid \text{clase}) \cdot P(\text{característica}_2 \mid \text{clase}) \cdot \dots}{P(\text{características})}$$

Logaritmo

$$P(\text{clase} \mid \text{características}) = \log P(\text{característica}_1 \mid \text{clase}) + \log P(\text{característica}_2 \mid \text{clase}) + \dots$$