

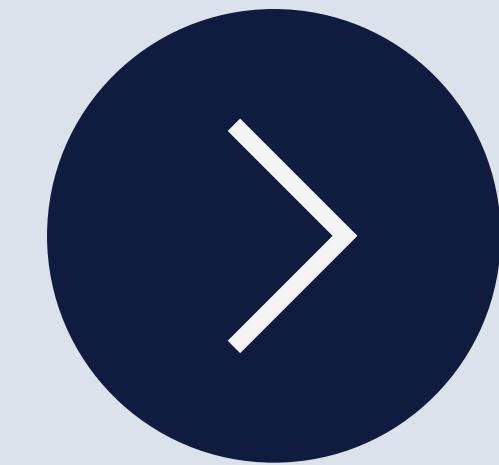
Inteligencia Artificial

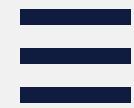
Naive Bayes con Estimación KDE para Mantenimiento Predictivo



Profesor: Manuel Cerón

Por: Rigoberto Zelayandia
Mario Morales

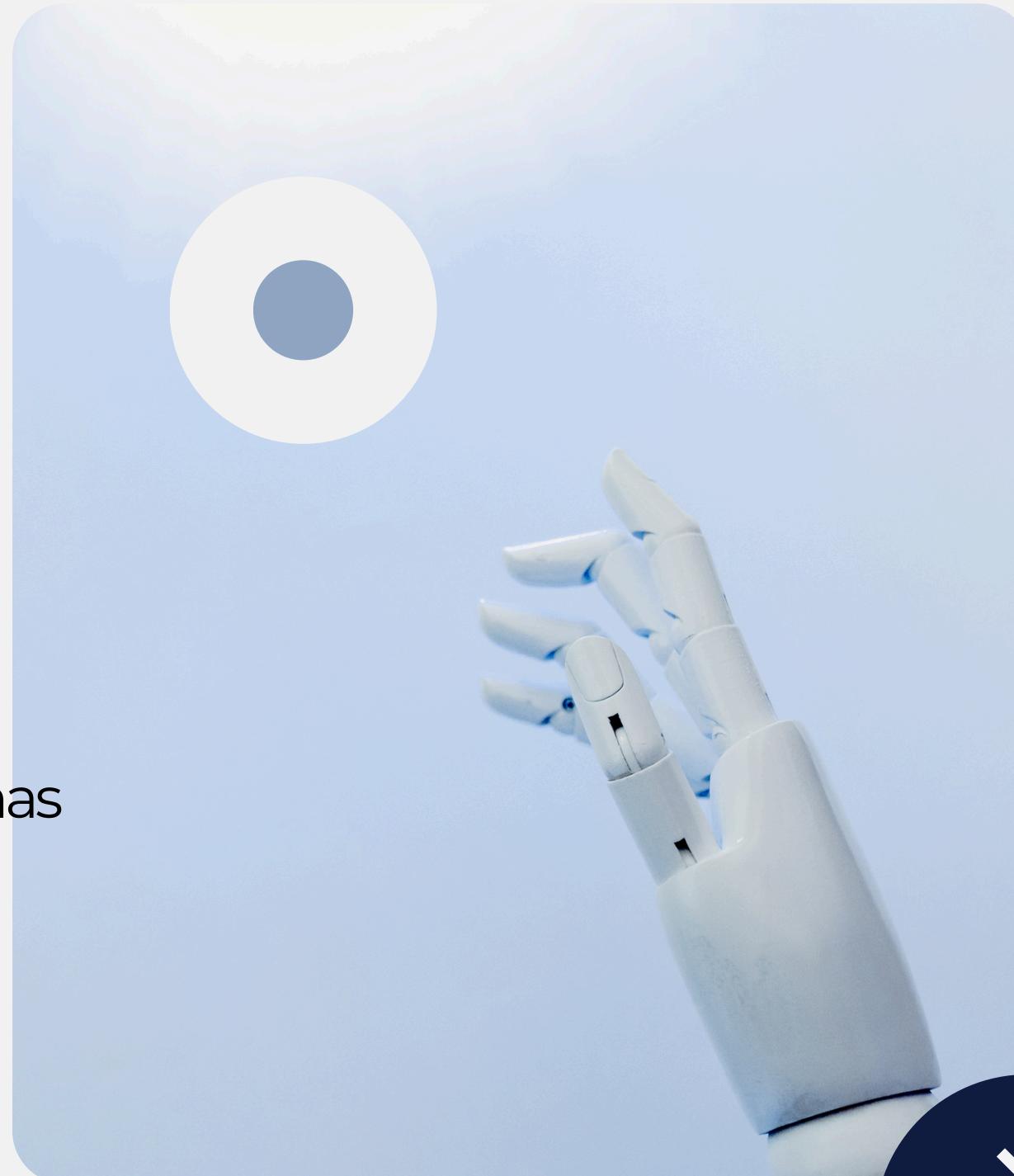
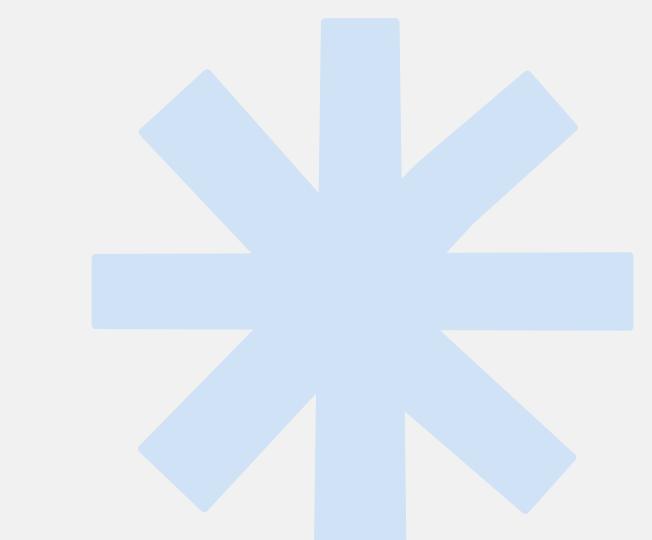




Objetivo.

Diseñar e implementar un clasificador propio basado en Naive Bayes, en el que la verosimilitud $P(X_i | y)$ se estime utilizando técnicas de Kernel Density Estimation (KDE) en lugar de asumir una distribución normal.

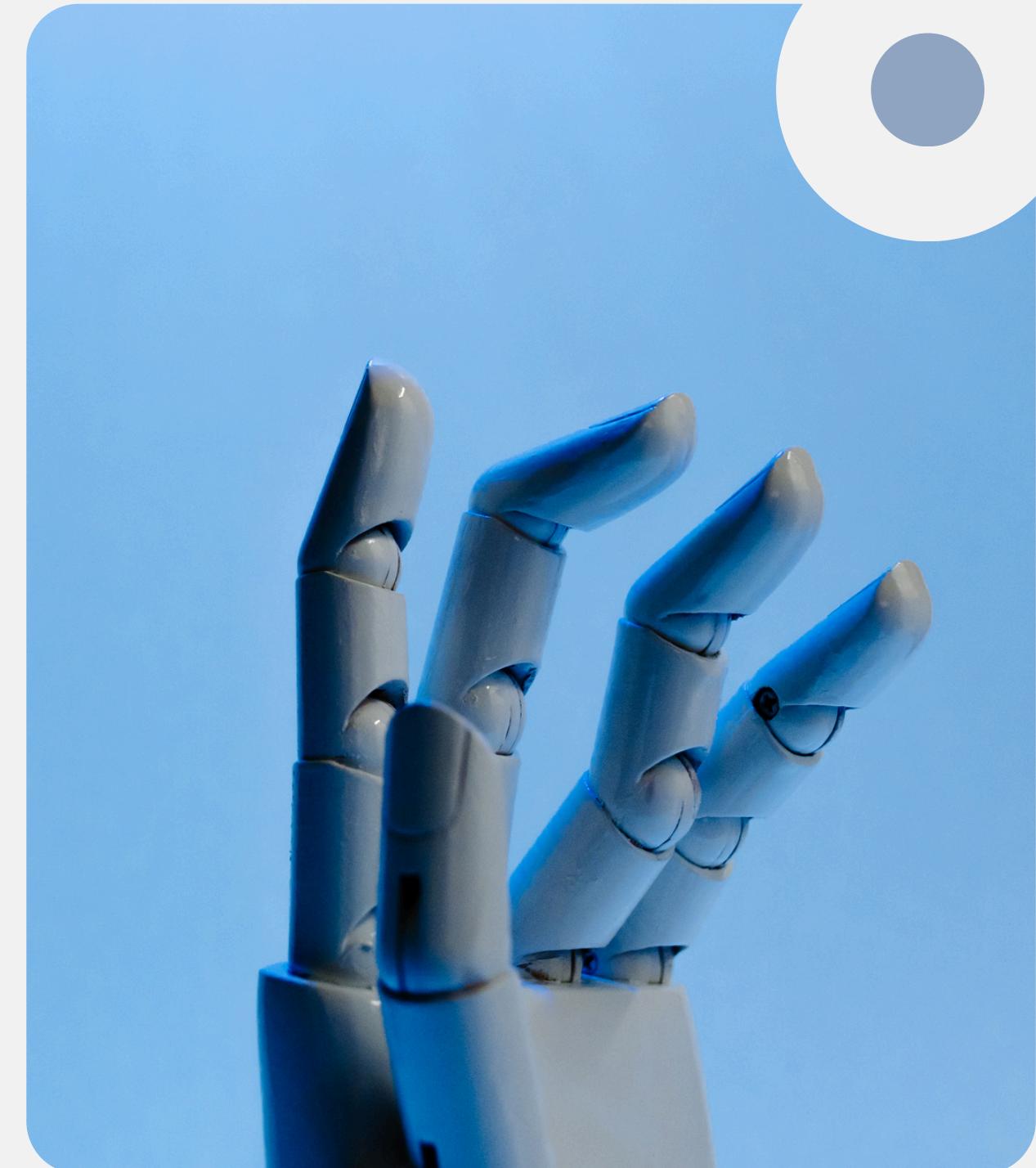
Un enfoque flexible para anticipar fallos en sistemas industriales



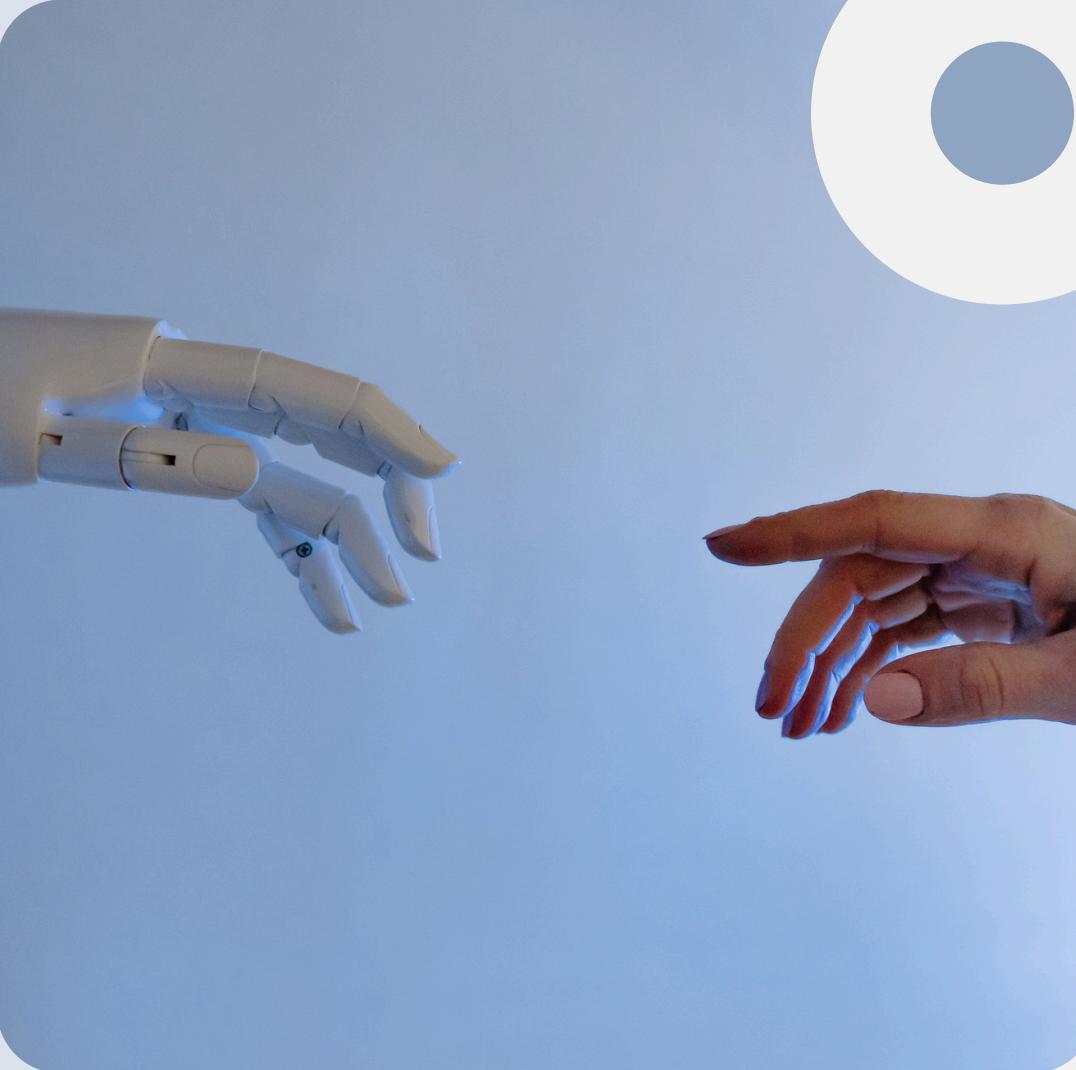
Introducción al Mantenimiento Predictivo

El mantenimiento predictivo permite anticipar fallas antes de que sucedan al monitorear variables como vibración, temperatura y consumo de energía. Esto ayuda a programar revisiones, evitar interrupciones y tomar decisiones con datos confiables.

Utiliza datos de sensores para tomar decisiones proactivas: predice fallas, optimiza mantenimiento y alarga la vida útil de los equipos.



¿Qué es Naive Bayes?



Un clasificador probabilístico basado en el teorema de Bayes.

$$P(y | \mathbf{x}) = \frac{P(y) \cdot P(\mathbf{x} | y)}{P(\mathbf{x})}$$

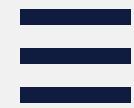
Dado que $P(\mathbf{x})$ es constante para todas las clases en un problema de clasificación, la predicción se basa en:

$$\hat{y} = \arg \max_y P(y) \prod_{i=1}^n P(x_i | y)$$

Asume que las variables son independientes entre sí.

Es simple, rápido y fácil de interpretar.



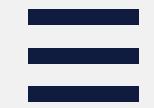


Introducción a KDE

La estimación de densidad por núcleo (KDE) es un método no paramétrico para estimar la función de densidad de probabilidad de una variable.

No asume una forma específica para la distribución de los datos, lo que lo hace flexible.





Combinando Bayes y KDE

Naive

Usar KDE para estimar las distribuciones de probabilidad en lugar de suponer distribuciones gaussianas.

Un clasificador más robusto para datos reales de mantenimiento.

Recolección de datos, preprocessamiento, entrenamiento del modelo y predicción.

Adaptable a diferentes tipos de datos y fácil de implementar.

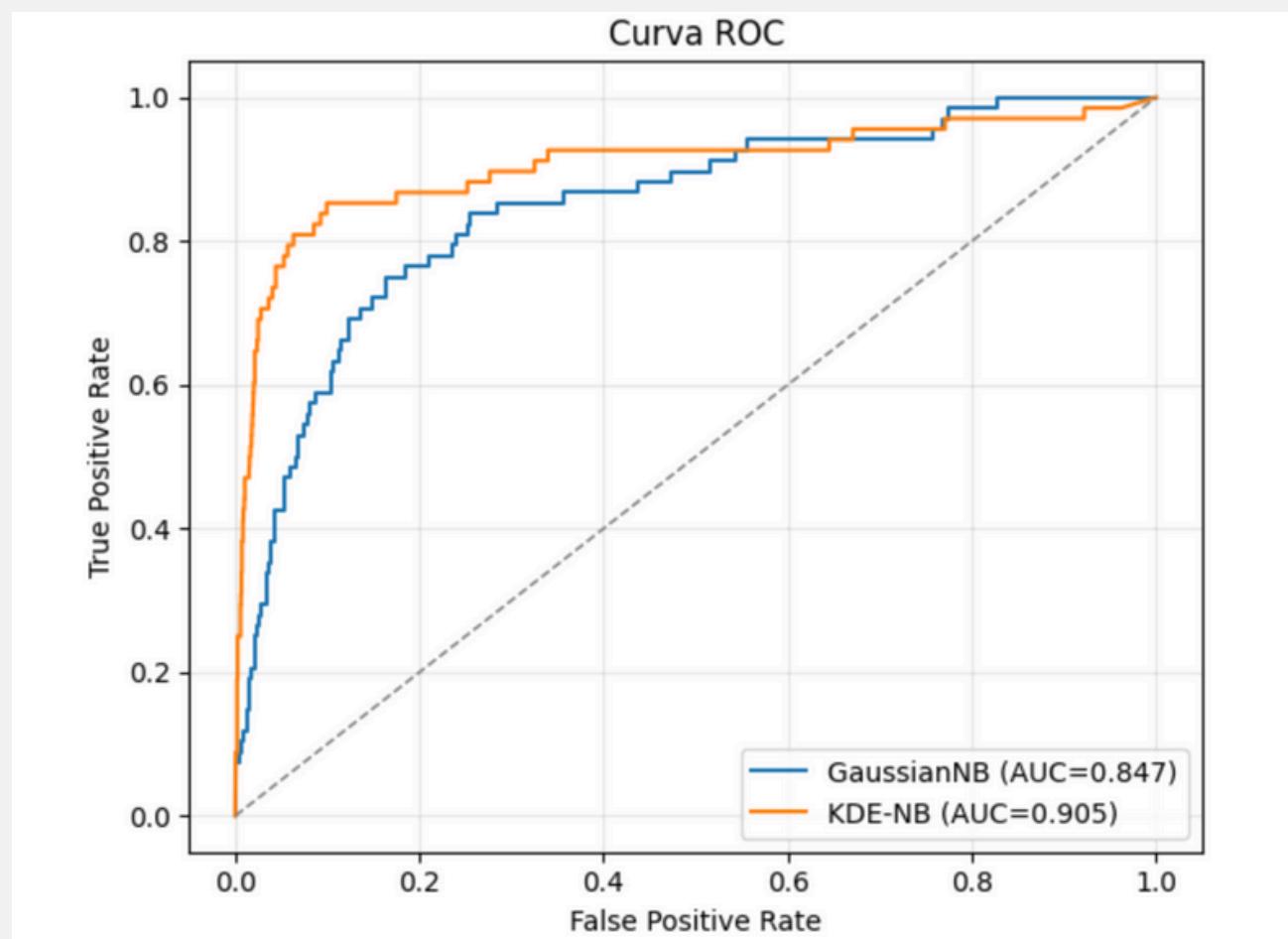


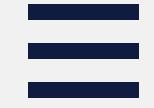
Desarollo Proyecto

del

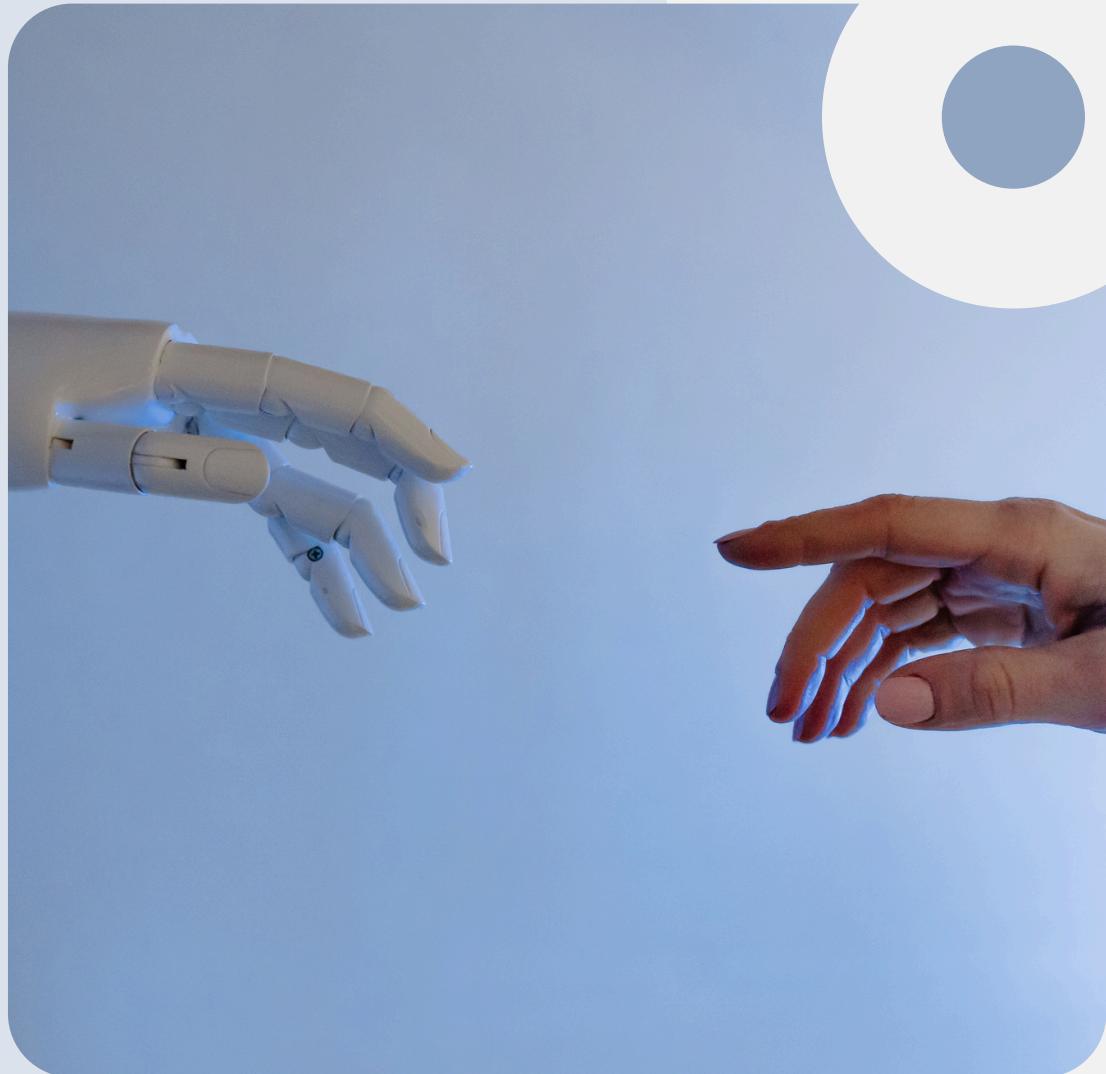
Curva ROC (AUC)

- Qué muestra: la sensibilidad (TPR) frente a la tasa de falsos positivos (FPR) para todos los umbrales.
- Lectura: cuanto más arriba e izquierda esté la curva, mejor. El AUC resume la capacidad de ranking del modelo (1.0 perfecto, 0.5 azar).
- Hallazgo: El KDE-NB se mantiene por encima de GaussianNB, lo que indica mejor discriminación global entre “falla” y “no falla”.





¿Usar KDE mejora la clasificación frente al modelo gaussiano tradicional? ¿bajo qué condiciones?



Depende de las características de tus datos. KDE mejora significativamente cuando las distribuciones NO son gaussianas, pero puede ser igual o peor cuando las distribuciones son aproximadamente normales.

KDE mejora cuando tus datos tienen "personalidad" (picos, valles, asimetrías, múltiples grupos).

Gaussian NB es suficiente cuando tus datos son "aburridos" (campana perfecta, bien comportados).

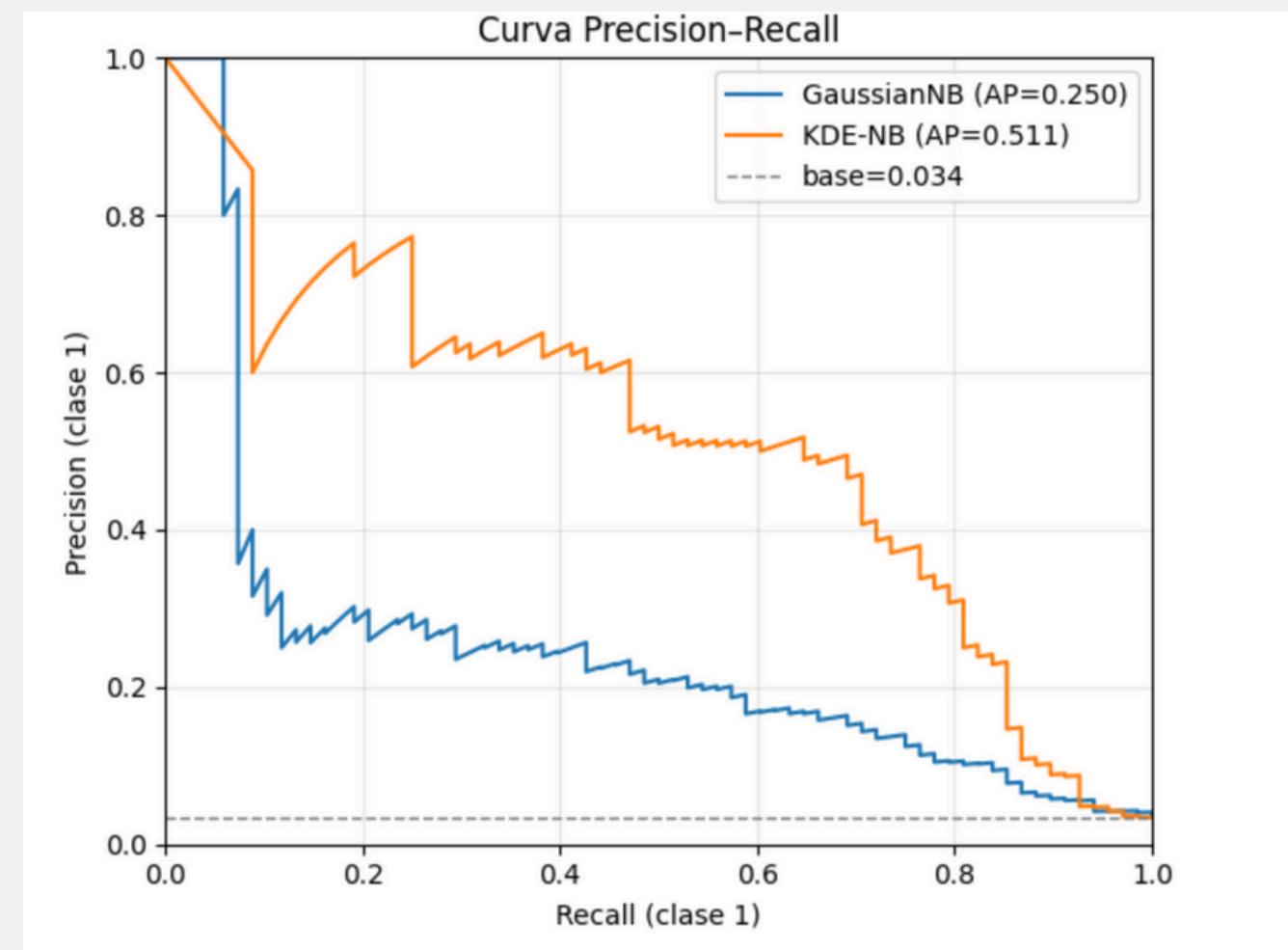
En mantenimiento predictivo, los datos suelen tener "personalidad" (comportamientos normales vs pre-falla), por eso KDE generalmente ayuda.

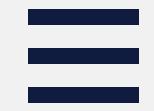


Curva de Precision Recall

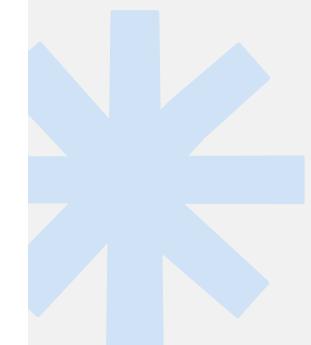
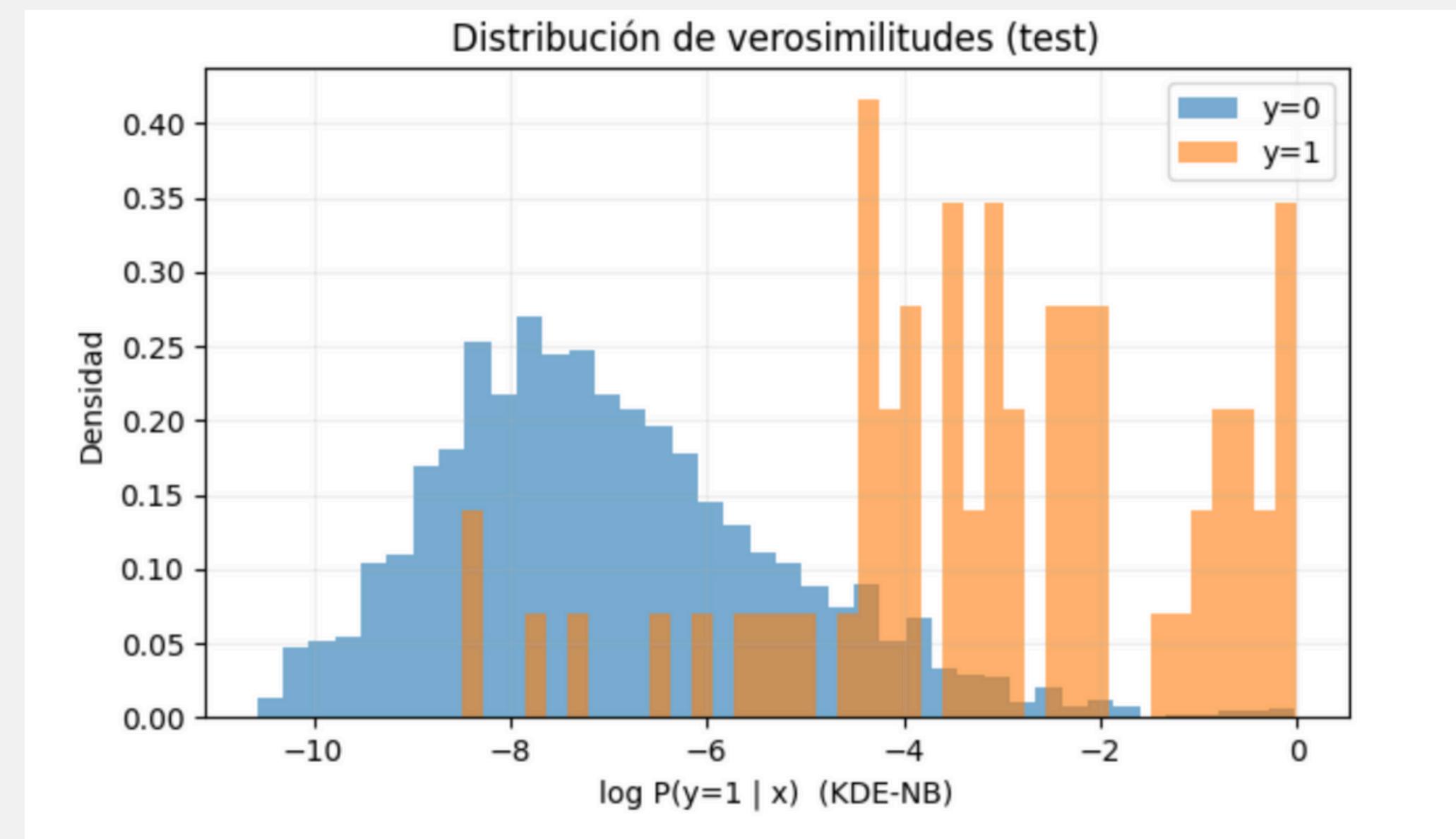
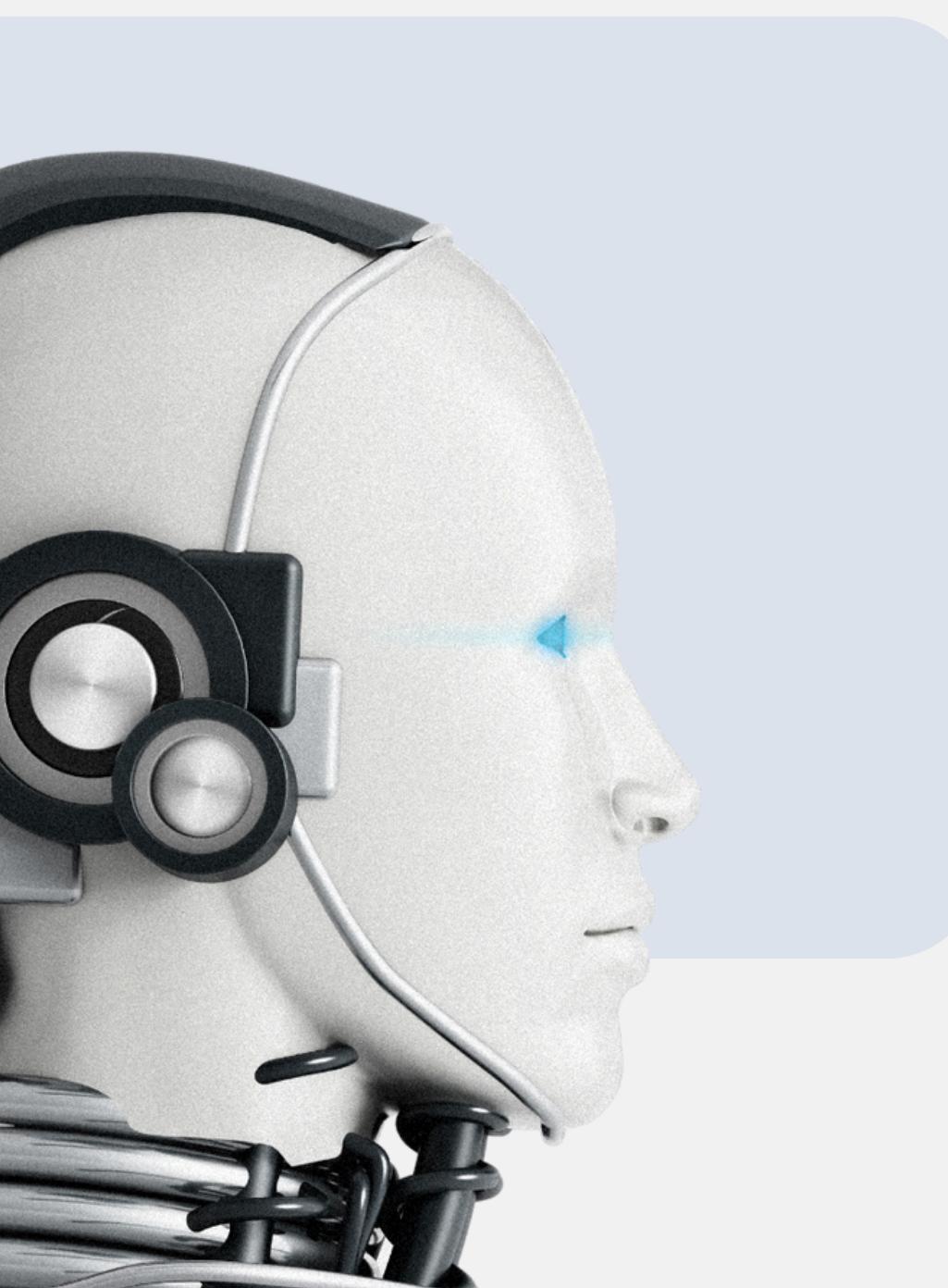
Curva Precision–Recall (AP)

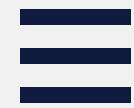
- Qué muestra: rendimiento sobre la clase positiva (fallas), crucial en datos desbalanceados.
- Línea base: coincide con la prevalencia de fallas en el set de test.
- Hallazgo: KDE-NB presenta AP mayor \Rightarrow mejor calidad para detectar fallas con menos falsos positivos a lo largo de los umbrales.



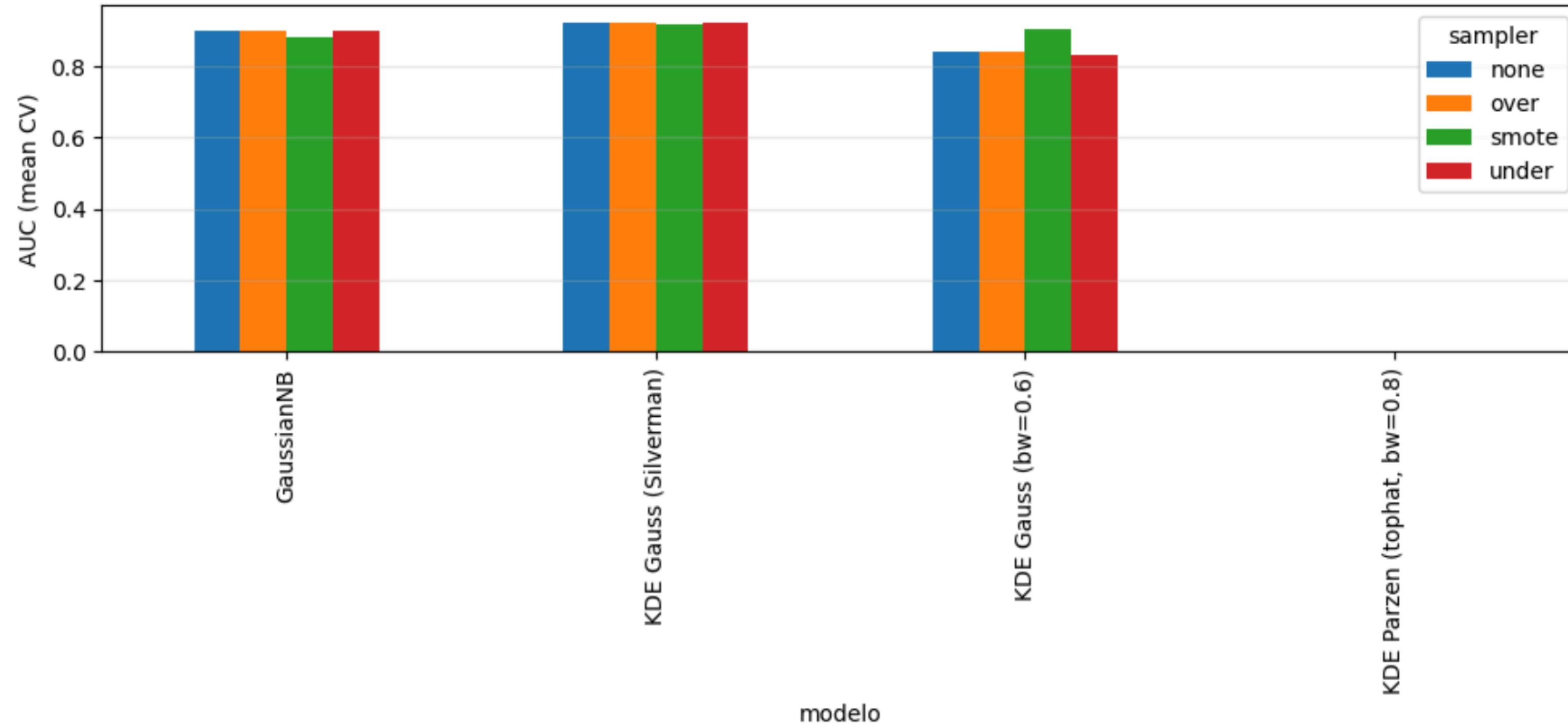


Distribuciones de Verosimilitud





AUC por modelo y estrategia de balanceo





KDE PARZEN Y KDE GAUSS CON SILVERMAN (HOLDOUT + UMBRAL ÓPTIMO)

```
==== KDE Parzen (bw=0.8, thr=0.5) ===
```

```
Accuracy: 0.9610
```

```
Matriz de confusión:
```

```
[[1908 24]
 [ 54 14]]
```

```
Reporte:
```

	precision	recall	f1-score	support
--	-----------	--------	----------	---------

0	0.9725	0.9876	0.9800	1932
1	0.3684	0.2059	0.2642	68

accuracy			0.9610	2000
----------	--	--	--------	------

macro avg	0.6704	0.5967	0.6221	2000
-----------	--------	--------	--------	------

weighted avg	0.9519	0.9610	0.9556	2000
--------------	--------	--------	--------	------

```
--- Umbral óptimo (Parzen) ---
```

```
Umbral: 0.330 | F1*: 0.3016
```

```
==== KDE Parzen (umbral óptimo) ===
```

```
Accuracy: 0.9560
```

```
Matriz de confusión:
```

```
[[1893 39]
 [ 49 19]]
```

```
Reporte:
```

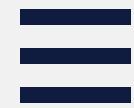
	precision	recall	f1-score	support
--	-----------	--------	----------	---------

0	0.9748	0.9798	0.9773	1932
1	0.3276	0.2794	0.3016	68

accuracy			0.9560	2000
----------	--	--	--------	------

macro avg	0.6512	0.6296	0.6394	2000
-----------	--------	--------	--------	------

weighted avg	0.9528	0.9560	0.9543	2000
--------------	--------	--------	--------	------



KDE PARZEN Y KDE GAUSS CON SILVERMAN (HOLDOUT + UMBRAL ÓPTIMO)

```
==== KDE Gaussian (Silverman, thr=0.5) ===
```

```
Accuracy: 0.9680
```

```
Matriz de confusión:
```

```
[[1926  6]
 [ 58 10]]
```

```
Reporte:
```

	precision	recall	f1-score	support
0	0.9708	0.9969	0.9837	1932
1	0.6250	0.1471	0.2381	68
accuracy			0.9680	2000
macro avg	0.7979	0.5720	0.6109	2000
weighted avg	0.9590	0.9680	0.9583	2000

```
--- Umbral óptimo (Silverman) ---
```

```
Umbral: 0.050 | F1*: 0.4638
```

```
==== KDE Gaussian (Silverman, umbral óptimo) ===
```

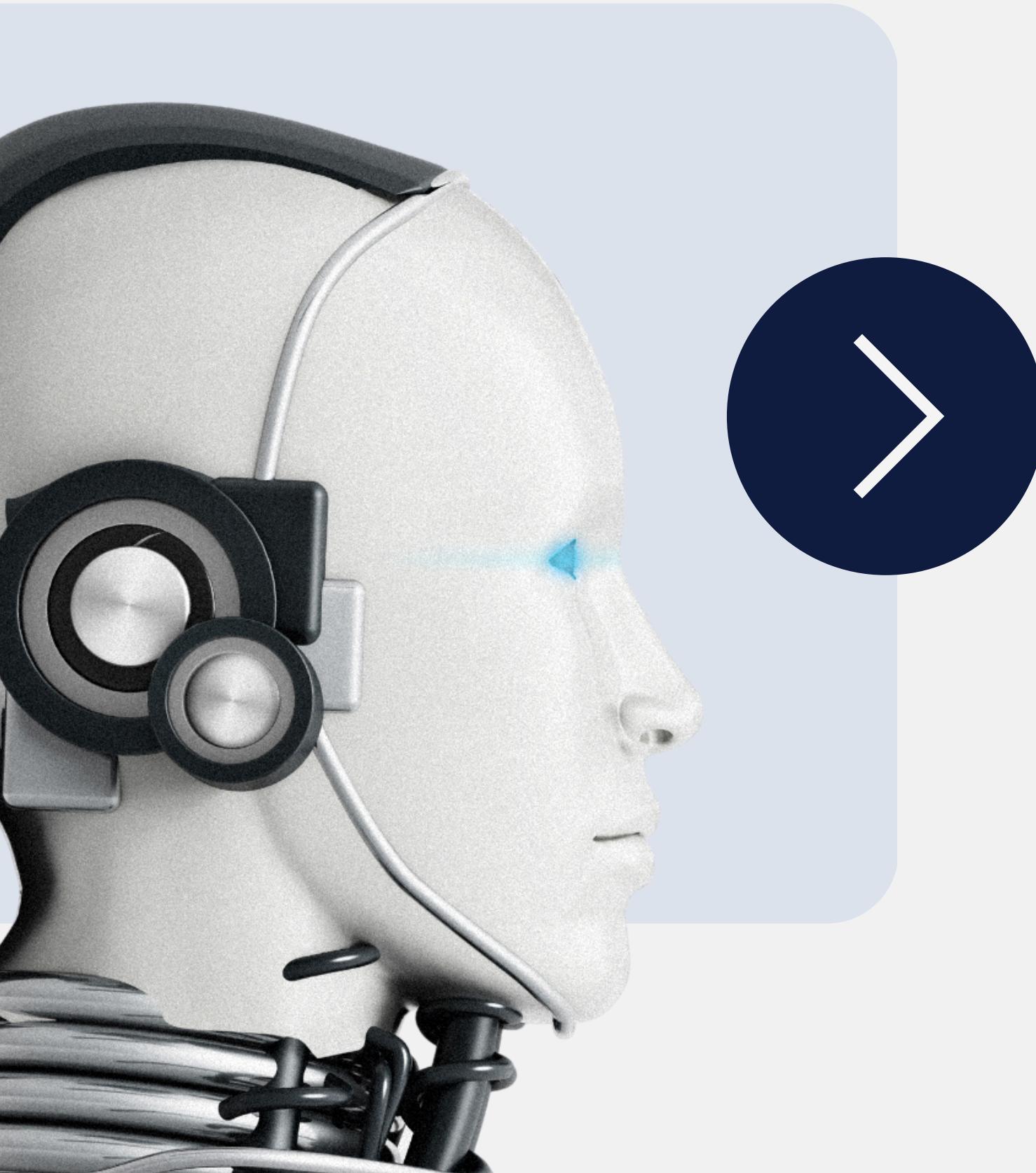
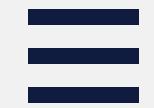
```
Accuracy: 0.9630
```

```
Matriz de confusión:
```

```
[[1894 38]
 [ 36 32]]
```

```
Reporte:
```

	precision	recall	f1-score	support
0	0.9813	0.9803	0.9808	1932
1	0.4571	0.4706	0.4638	68
accuracy			0.9630	2000
macro avg	0.7192	0.7255	0.7223	2000
weighted avg	0.9635	0.9630	0.9633	2000



Conclusiones

- Naive Bayes con KDE es una solución simple y flexible para mantenimiento predictivo.
- Mejora la precisión y permite anticipar fallas con mayor confianza.

