

# Modelos de predicción para diagnóstico de tumores cerebrales

The background of the slide is a dark, grayscale image of a human brain. A specific region on the right side of the brain is highlighted with a red outline, indicating a tumor. The text is overlaid on this image.

---

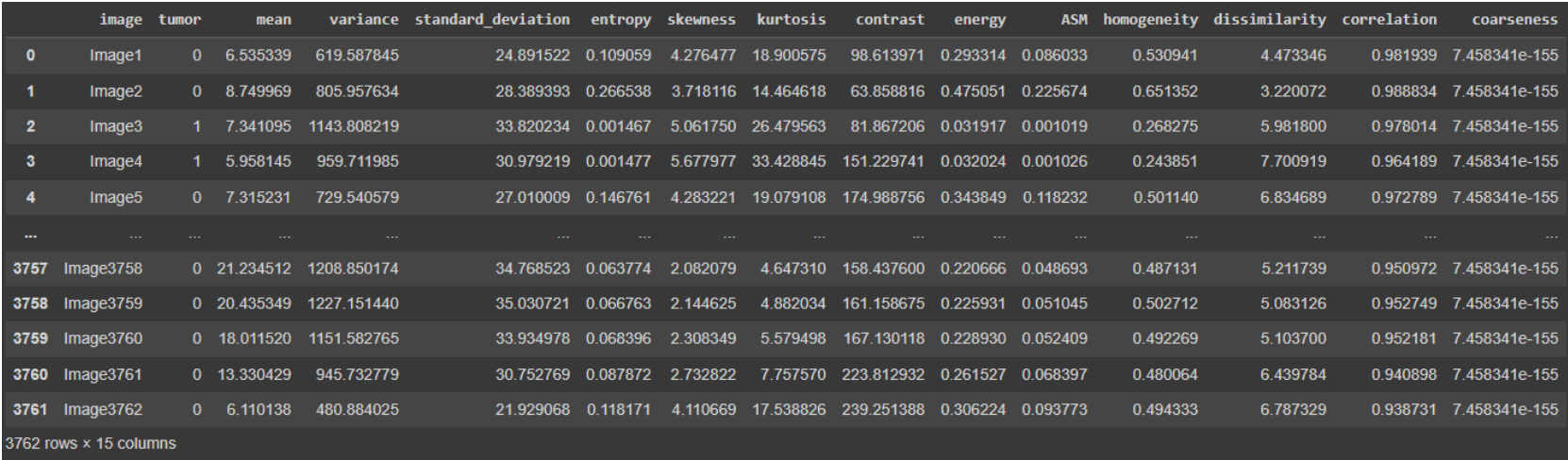
Arturo Moreno Puga

Bootcamp en Data Science & Machine Learning

ID Bootcamps

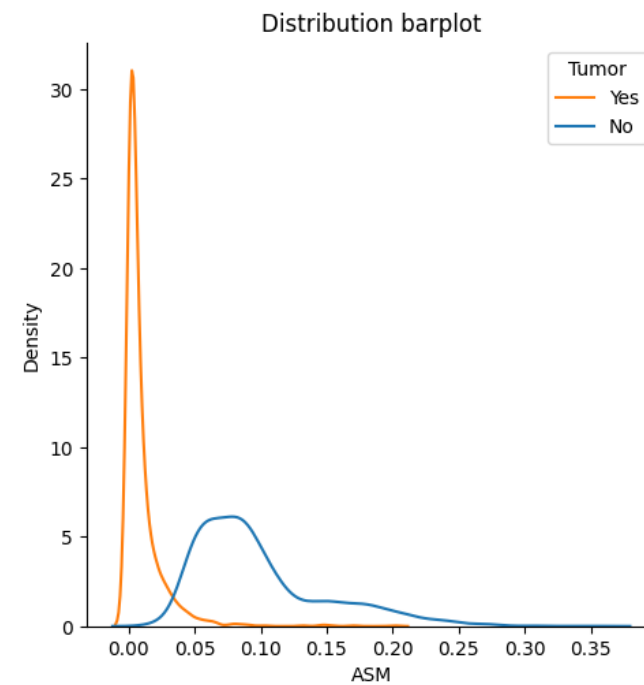
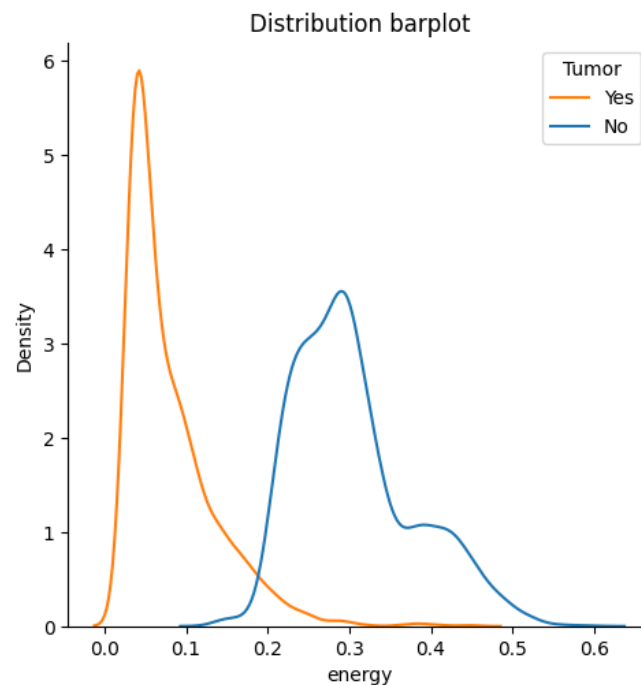
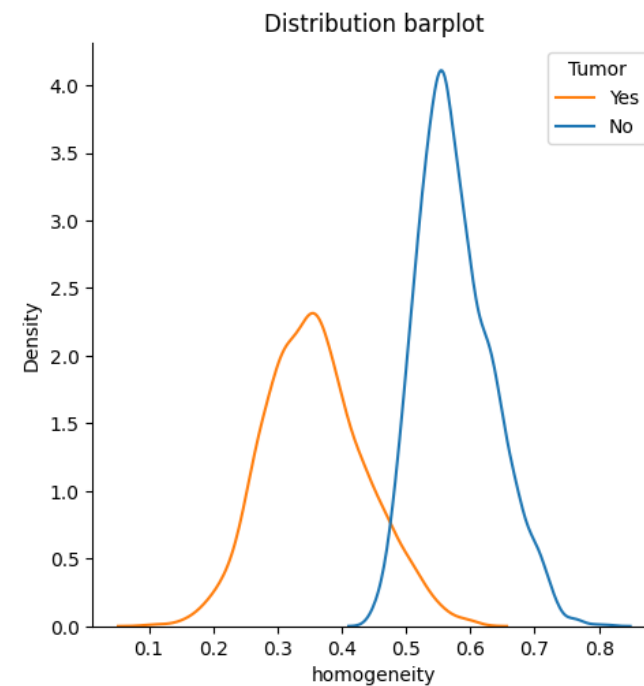
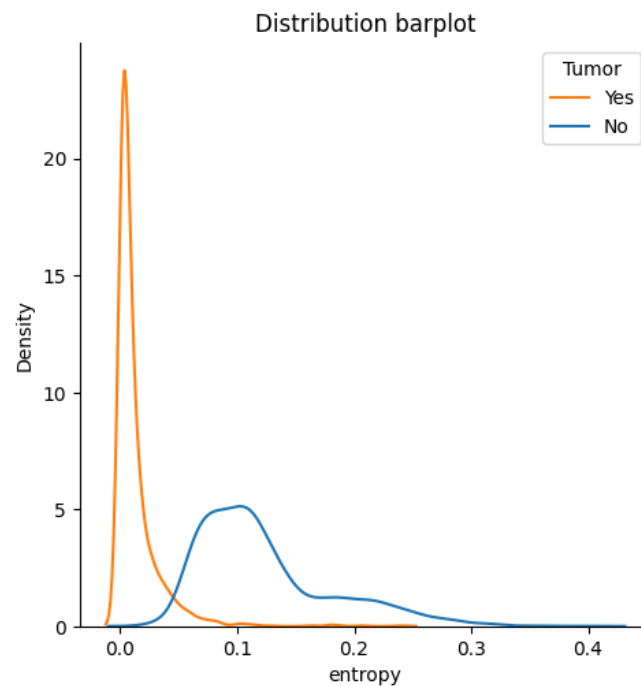
Curso 2022/2023

# Machine Learning



## Features más relevantes:

- ASM
- Energy
- Entropy
- Homogeneity



# Machine Learning

## Optimizar recall

Mayor recall



Menor cantidad de  
falsos negativos (FN)



Menor cantidad de pacientes  
que hemos diagnosticado sin  
tumor cuando realmente sí lo  
tienen

		Predicción	
		No hay tumor	Sí hay tumor
Realidad	No hay tumor	TN	FP
	Sí hay tumor	FN	TP

$$recall = \frac{TP}{TP + FN}$$

# Machine Learning

## KNN (k-Nearest Neighbors)

- n\_neighbors = 1
- weights = uniform
- leaf\_size = 10

Recall	Precision	Accuracy	F1-Score
0.9760	0.9939	0.9867	0.9849

## Decision Tree

- criterion = gini
- max\_depth = 150

Recall	Precision	Accuracy	F1-Score
0.9671	0.9788	0.9761	0.9729

# Deep Learning

## Modelo **CNN** (red neuronal convolucional)

- Nº de épocas = 100
- Regularizador L1 → reduce overfitting
- Optimizador Adam
- Función de pérdida binary\_crossentropy

→ Conv2D( filters = 16 )  
Conv2D( filters = 16 )

→ Conv2D( filters = 32 )  
Conv2D( filters = 32 )

→ Conv2D( filters = 64 )  
Conv2D( filters = 64 )

→ Capas de agrupamiento  
(MaxPooling2D)

→ Capas de aplanamiento  
(Flatten)

→ Capas completamente  
conectadas (Dense)

- 256 neuronas, ReLU
- 128 neuronas, ReLU
- 1 neurona, sigmoid

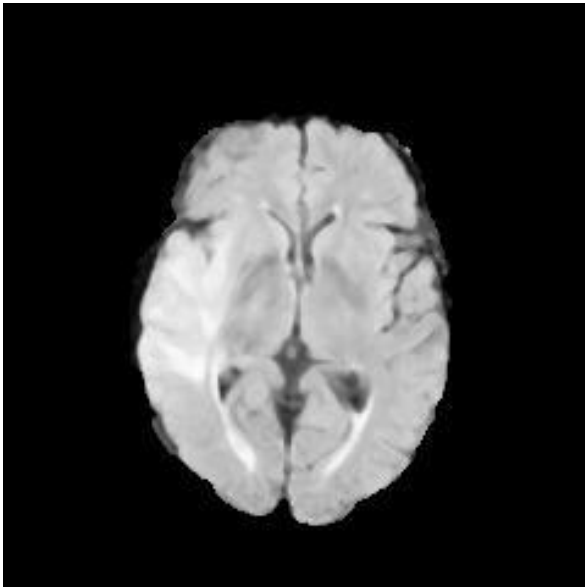
- **Épocas:** Cantidad de veces que se pasa por todo el conjunto de entrenamiento.
- **Filters:** Cantidad de filtros en cada capa. Cada filtro es una matriz de pesos que extrae características.

$$n = \underbrace{\left( \text{pesos} \right)}_{\text{filtro}} n-1$$

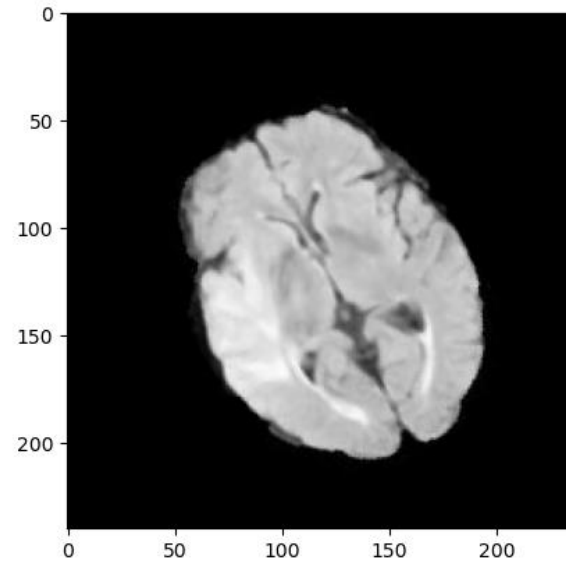
- En cada época los pesos se ajustan iterativamente.

# Deep Learning

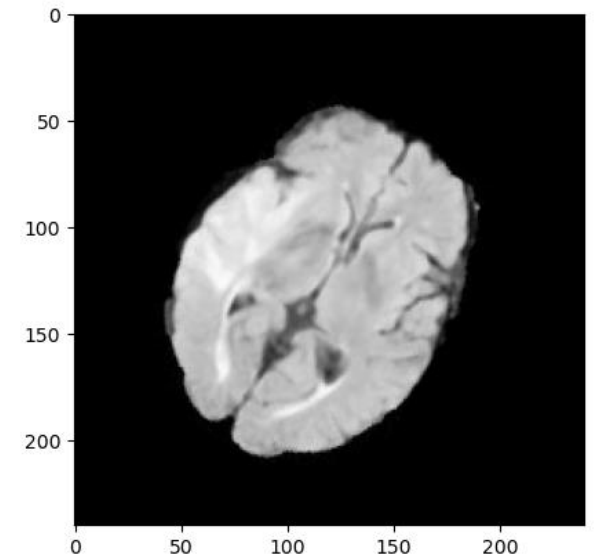
Rotaciones de  $30^\circ$  y  $-30^\circ$   $\longrightarrow$  Aumenta conjunto de entrenamiento



$0^\circ$



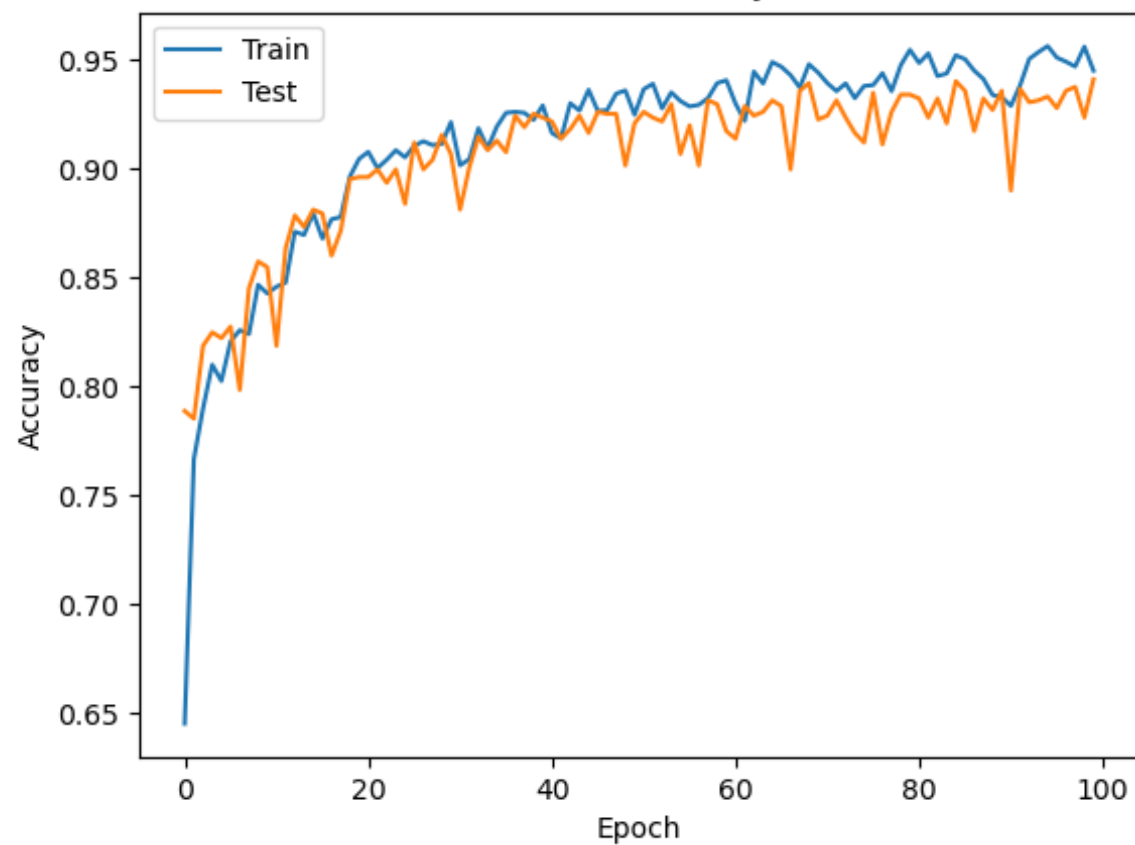
$30^\circ$



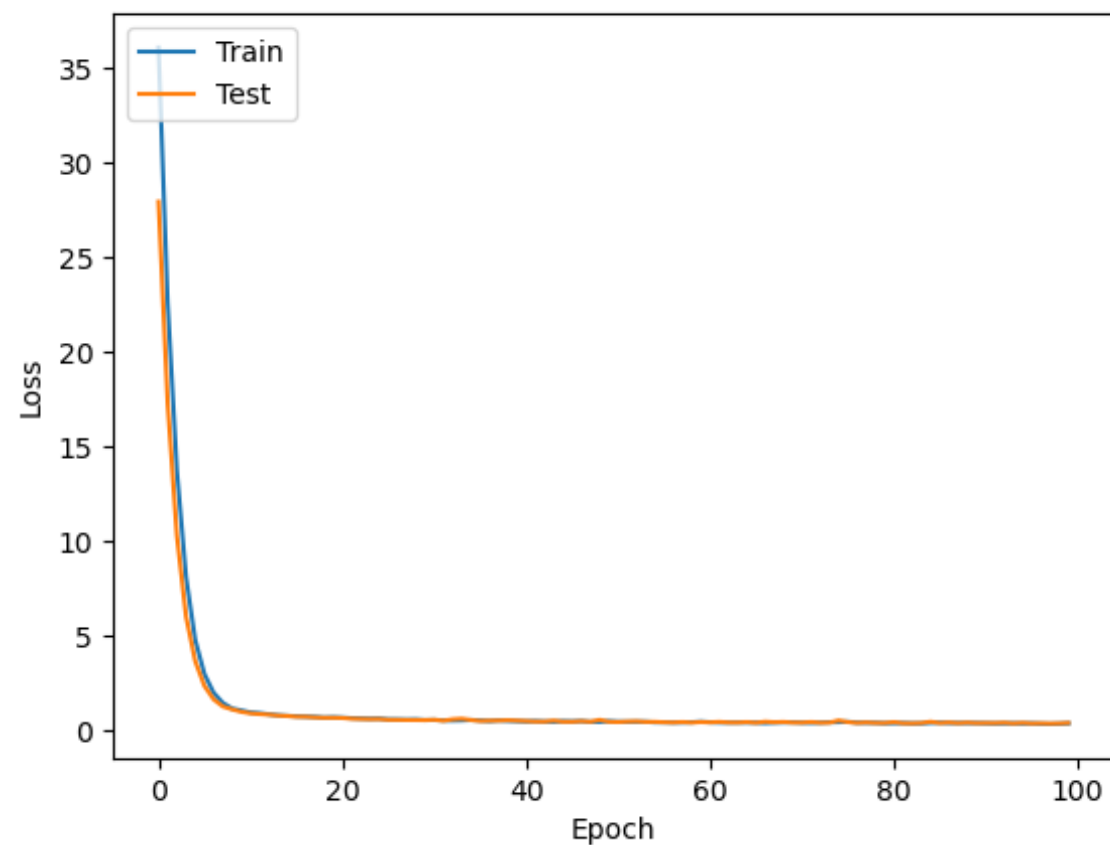
$-30^\circ$

# Deep Learning

Model accuracy

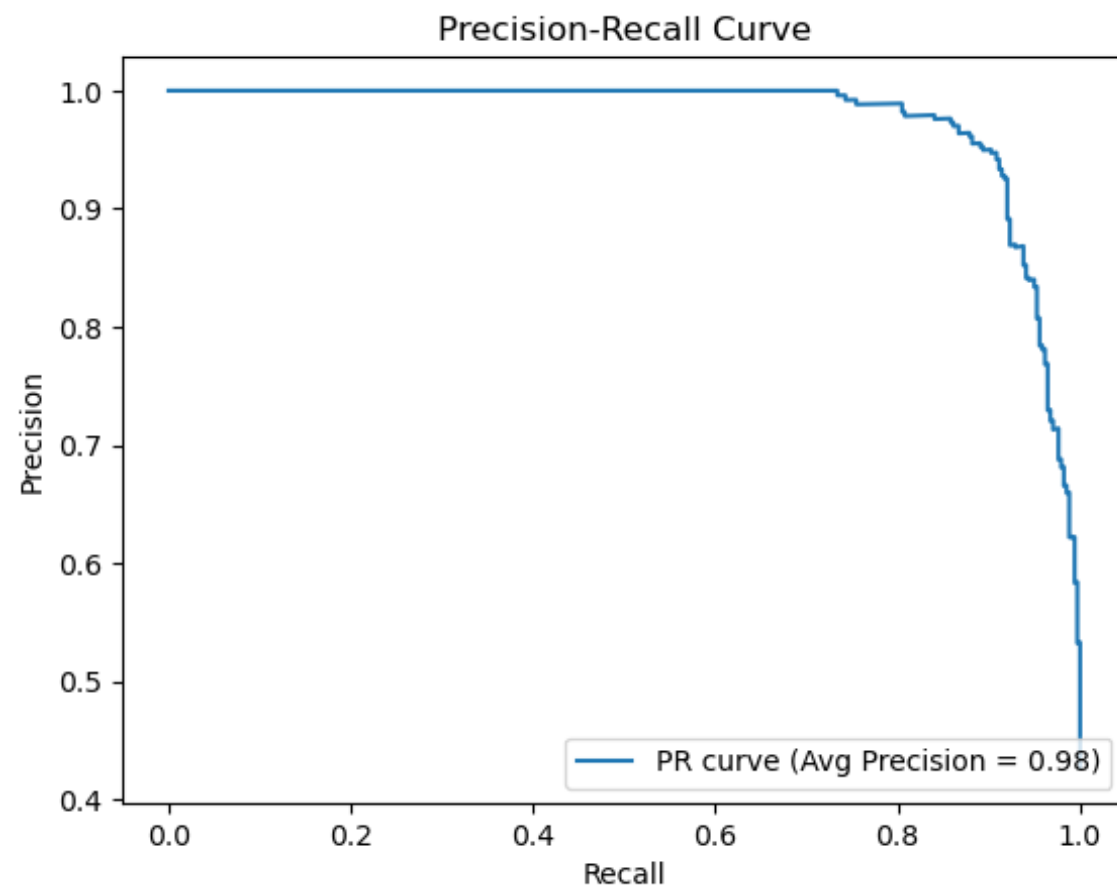
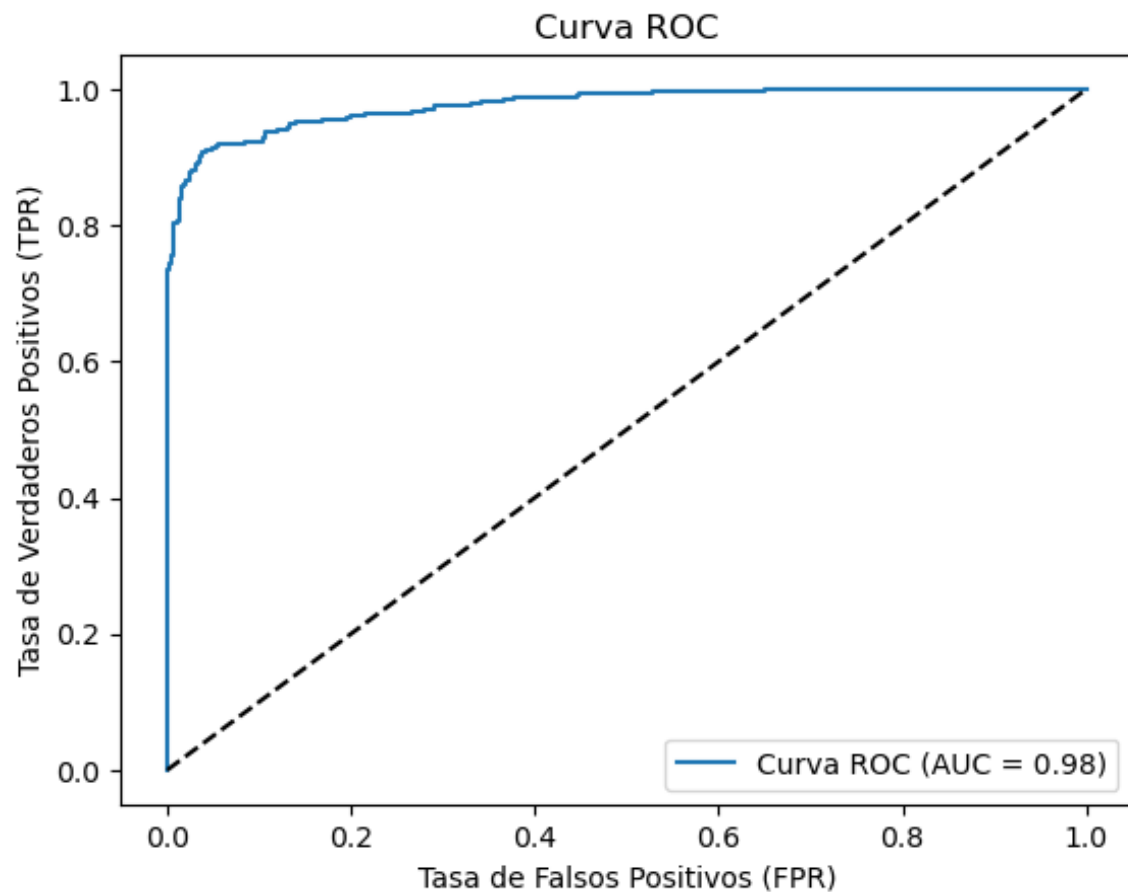


Model loss





# Deep Learning



# Resultados finales

	Recall	Precision	Accuracy	F1-Score
KNN	0.9760	0.9939	0.9867	0.9849
CNN	0.9201	0.9120	0.9281	0.9160

- Métricas para KNN mayores que para CNN.
- KNN requiere extracción de features previa, CNN la realiza internamente.
- Para ambos modelos, métricas mayores a 0.9.

# Conclusiones

- Aunque KNN presenta mejores métricas, CNN tiene una aplicación más sencilla.
- Ambos modelos muestran un buen rendimiento.
- Se requeriría un estudio posterior sobre un dataset desbalanceado para observar el comportamiento en un caso más real.



