



Modelos de predicción para diagnóstico de tumores cerebrales

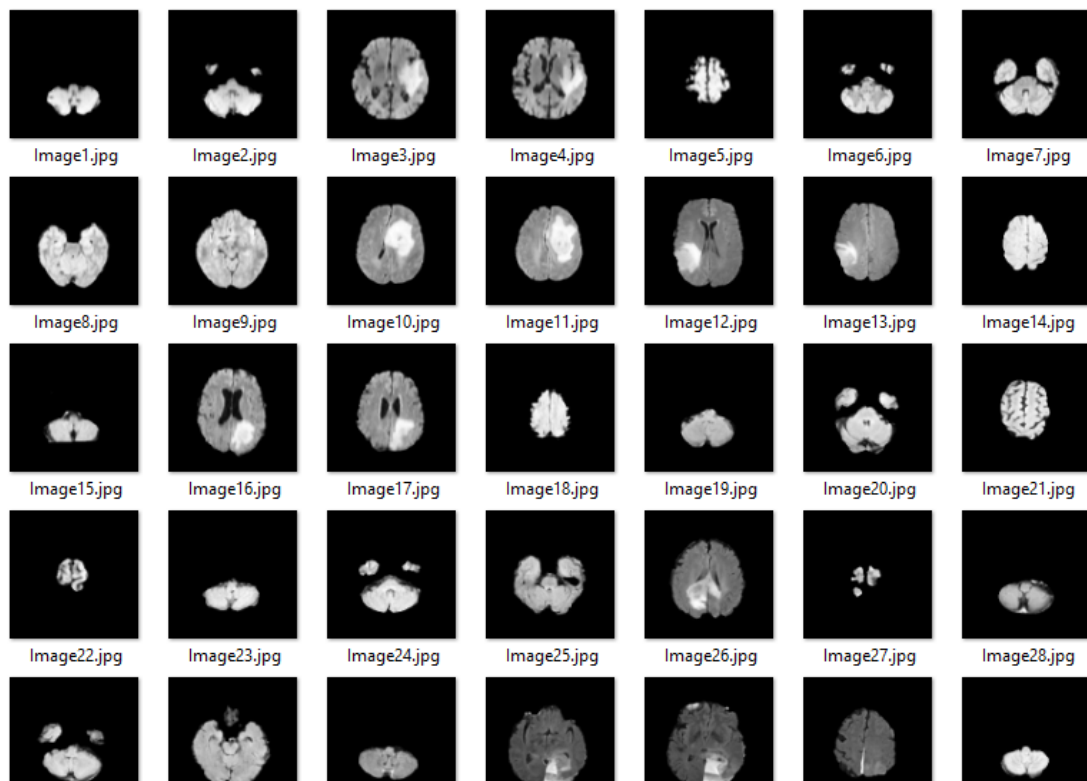
Arturo Moreno Puga

Bootcamp en Data Science & Machine Learning

ID Bootcamps

Curso 2022/2023

Deep Learning

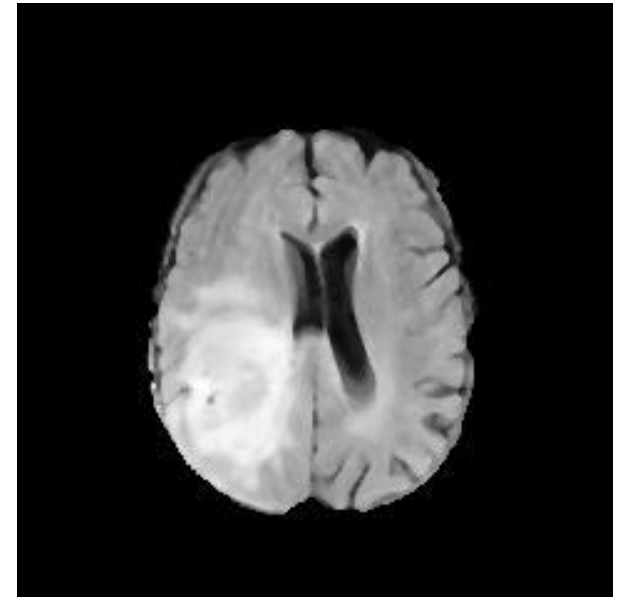
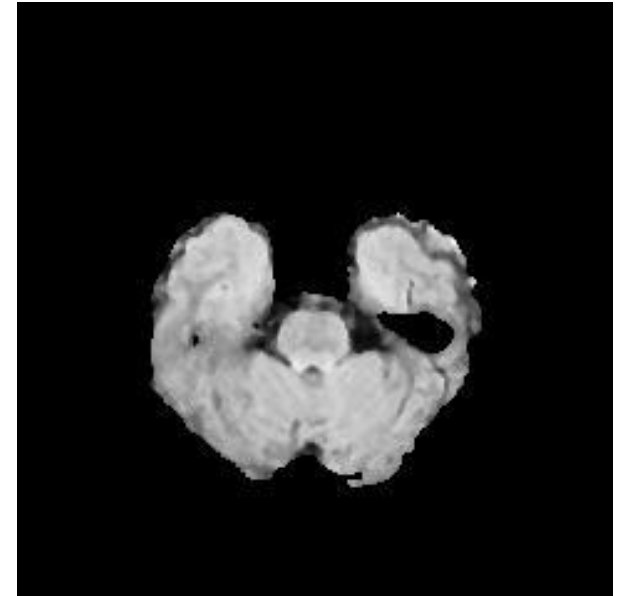


Machine Learning

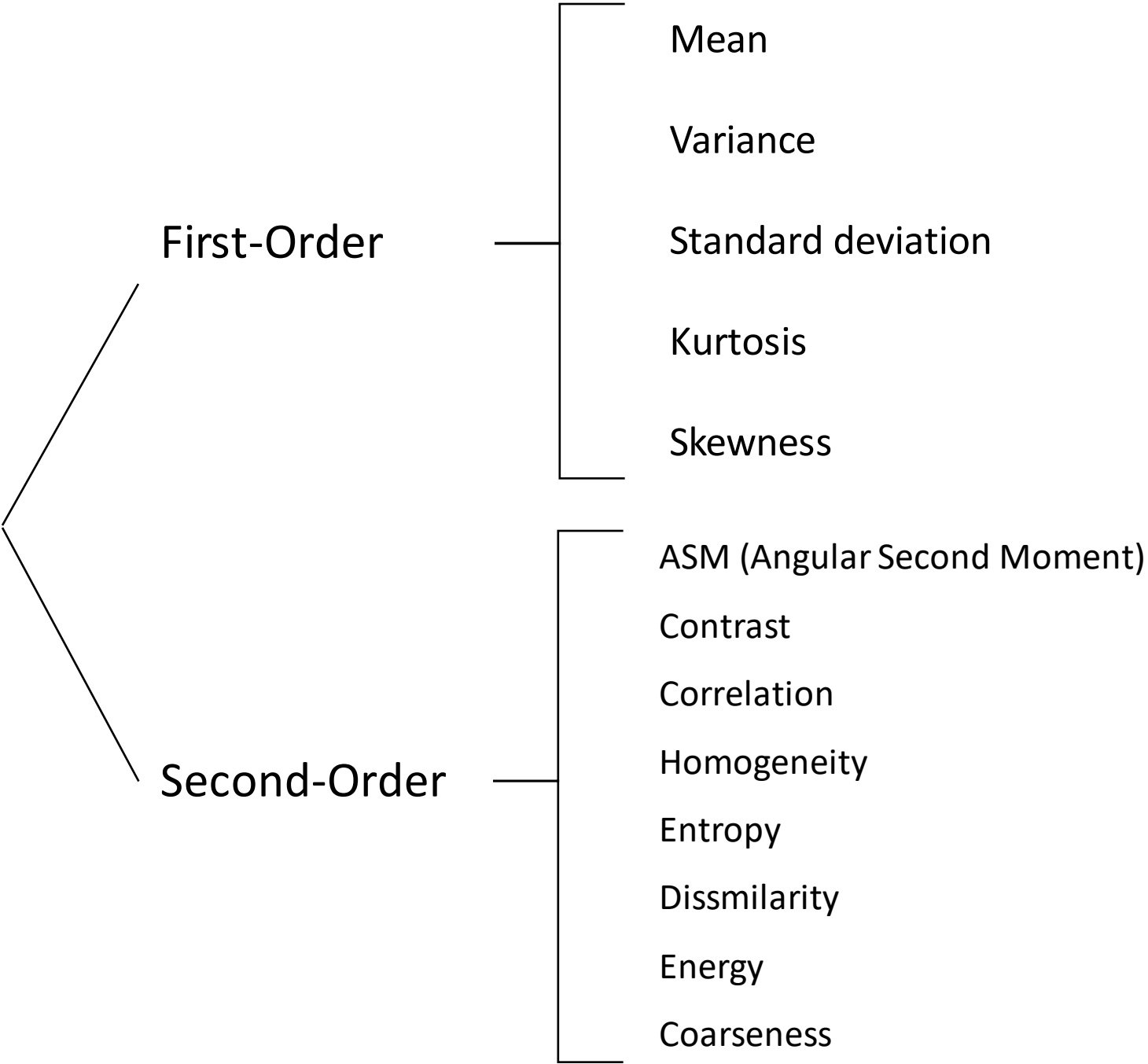
[illegible]

Dataset features

- Estadísticos que informan sobre la distribución de niveles de gris en la imagen.
- Se dividen en dos grupos: features de Primer Orden y features de Segundo Orden.
- Variable target categórica binaria que indica presencia o ausencia de tumor.



Features



Features de primer orden

Sea I cierta variable aleatoria que representa los niveles de gris en cierta región de la imagen. Se define el histograma de primer orden como

$$P(I) = \frac{\text{nº de píxeles con nivel de gris } I}{\text{nº total de píxeles en la región}}$$

- **Mean:** Valor medio del histograma.
- **Variance y Standard Deviation:** Medida de la anchura del histograma. Mide la desviación de los niveles de gris con respecto de la media.
- **Skewness:** Medida del grado de asimetría del histograma con respecto de la media.
- **Kurtosis:** Medida de la pronunciación de pico que muestra el histograma.

Features de segundo orden

Se mide la configuración de niveles de gris con una matriz de frecuencias relativas $P_{d,\theta}(I_1, I_2)$ que describe con qué frecuencia dos píxeles con niveles de gris I_1, I_2 aparecen separados por una distancia d en la dirección θ .

$$\text{ASM} = \sum_{i,j} P(I_1, I_2)^2$$

$$\text{Contrast} = \sum_{I_1, I_2} |I_1 - I_2|^2 \log P(I_1, I_2)$$

$$\text{Correlation} = \sum_{I_1, I_2} \frac{(I_1 - \mu_1)(I_2 - \mu_2)P(I_1, I_2)}{\sigma_1 \sigma_2}$$

$$\text{Homogeneity} = \sum_{I_1, I_2} \frac{P(I_1, I_2)}{1 + |I_1 - I_2|^2}$$

$$\text{Entropy} = - \sum_{I_1, I_2} P(I_1, I_2) \log P(I_1, I_2)$$

$$ASM = \sum_{i,j} P(I_1, I_2)^2$$

$$Contrast = \sum_{I_1, I_2} |I_1 - I_2|^2 \log P(I_1, I_2)$$

$$Correlation = \sum_{I_1, I_2} \frac{(I_1 - \mu_1)(I_2 - \mu_2)P(I_1, I_2)}{\sigma_1 \sigma_2}$$

$$Homogeneity = \sum_{I_1, I_2} \frac{P(I_1, I_2)}{1 + |I_1 - I_2|^2}$$

$$Entropy = - \sum_{I_1, I_2} P(I_1, I_2) \log P(I_1, I_2)$$

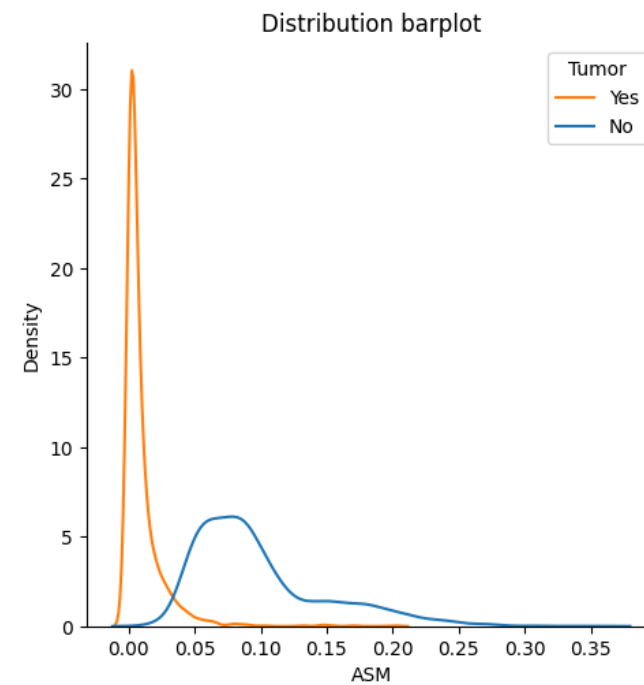
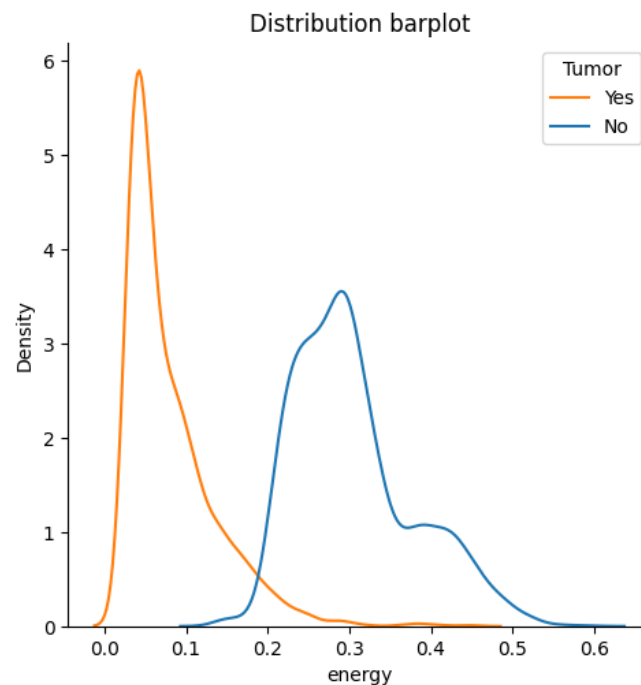
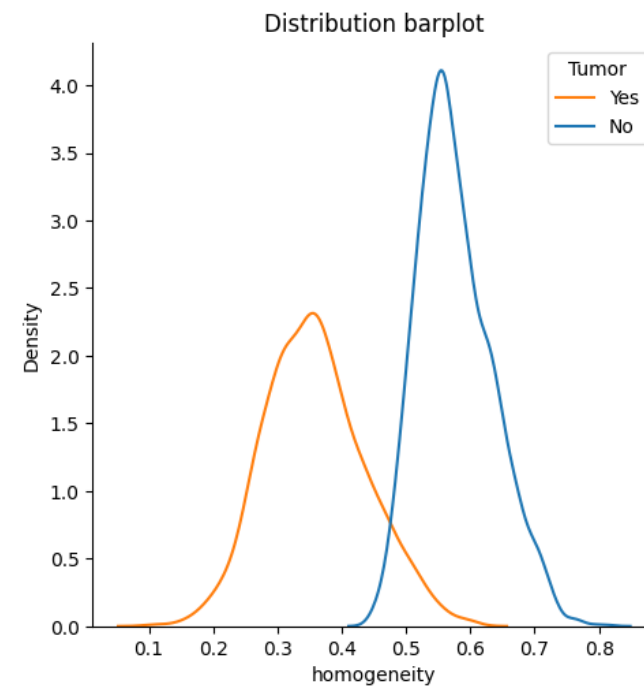
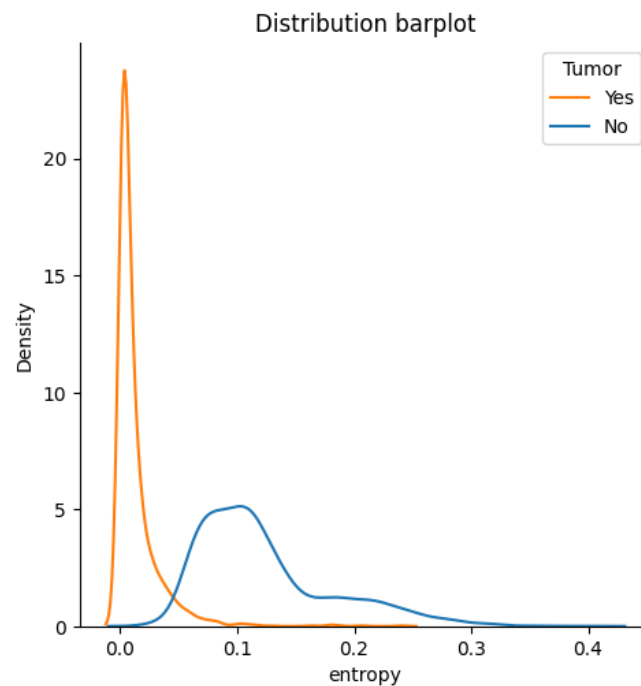
[\[1\]](#)

- **ASM:** Suavidad de la imagen. Valores más bajos indican menor suavidad.
- **Contrast:** Variaciones de niveles locales. Valores más altos indican mayor contraste.
- **Correlation:** Correlación entre píxeles en dos direcciones distintas.
- **Homogeneity:** Mayor homogeneidad indica imágenes con bajo contraste.
- **Entropy:** Aleatoriedad. Valores bajos indican imágenes con mayor suavidad.

-
- **Dissmilarity:** Falta de parecido entre píxeles. Valores altos indican menos semejanza.
 - **Energy:** No se indica en la bibliografía usada.
 - **Coarseness:** Valor constante => No se tiene en cuenta.

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

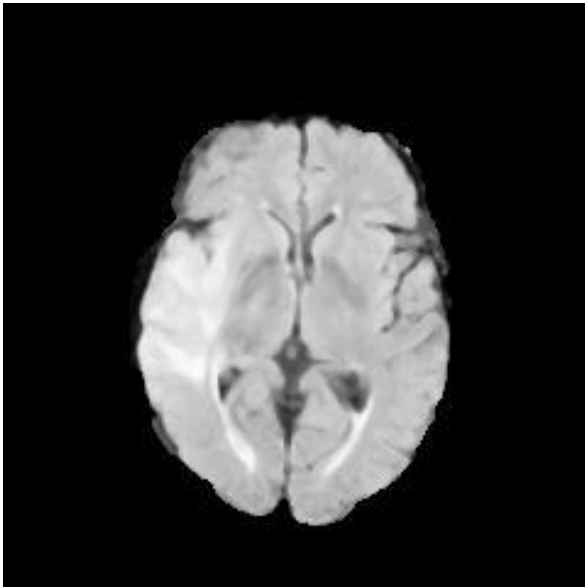
- **Épocas:** Cantidad de veces que se pasa por todo el conjunto de entrenamiento.
- **Filters:** Cantidad de filtros en cada capa. Cada filtro es una matriz de pesos que extrae características.

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

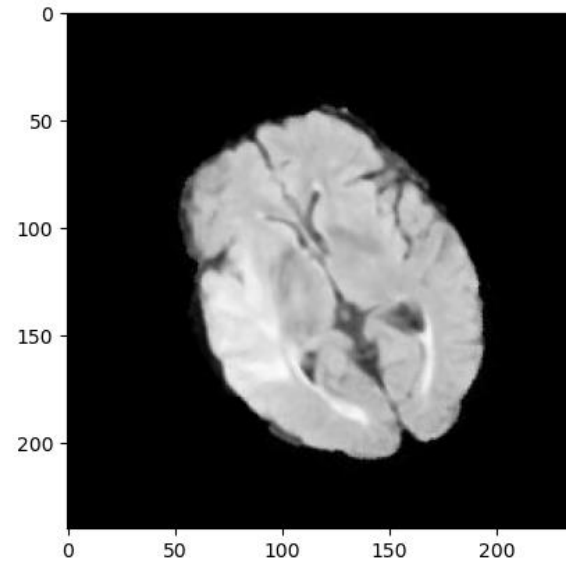
- En cada época los pesos se ajustan iterativamente.

Deep Learning

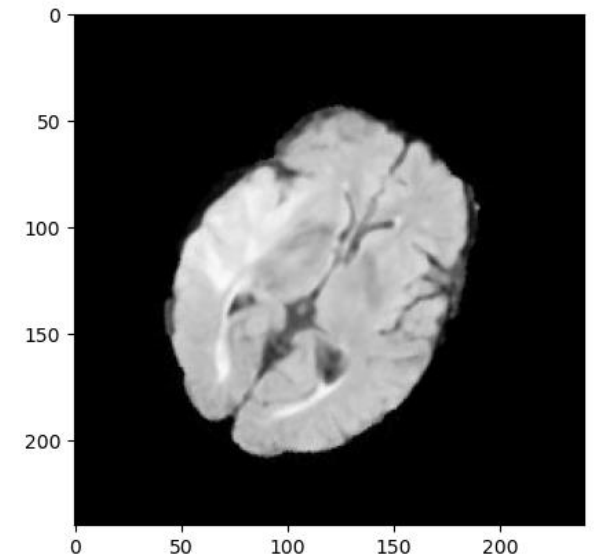
Rotaciones de 30° y -30° \longrightarrow Aumenta conjunto de entrenamiento



0°



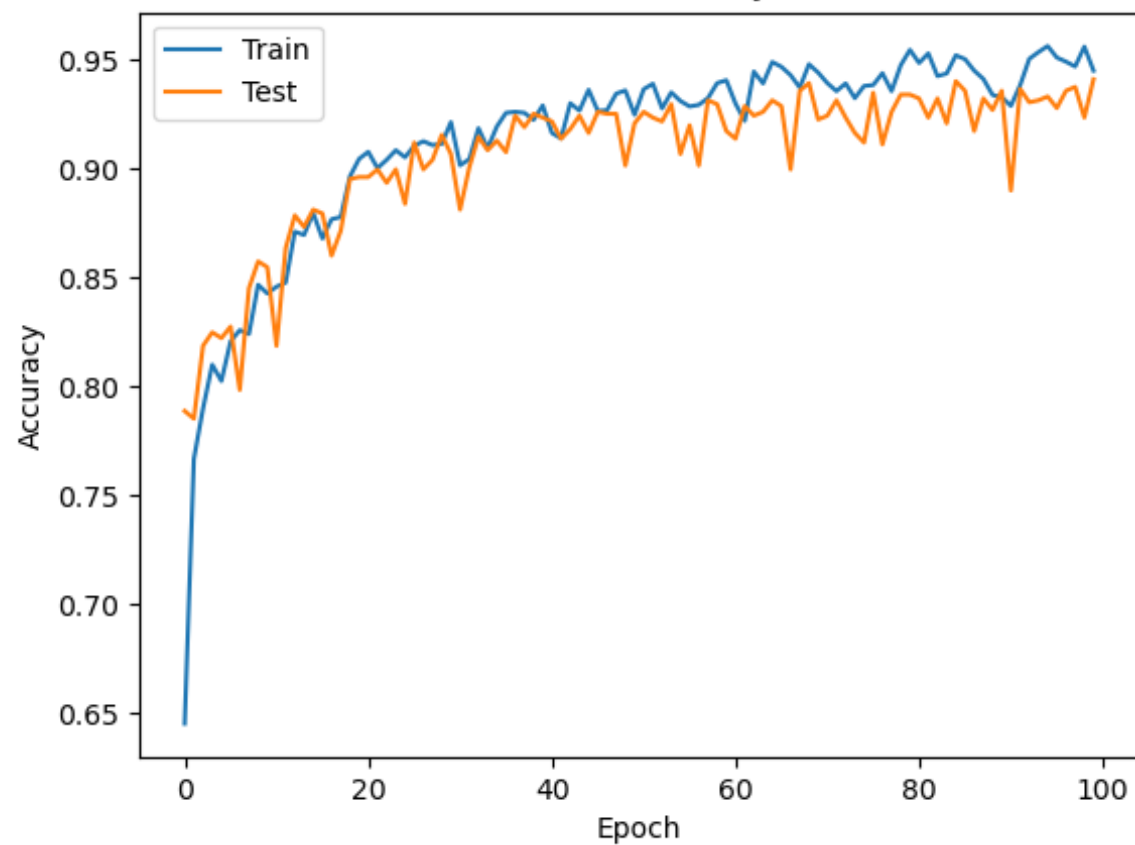
30°



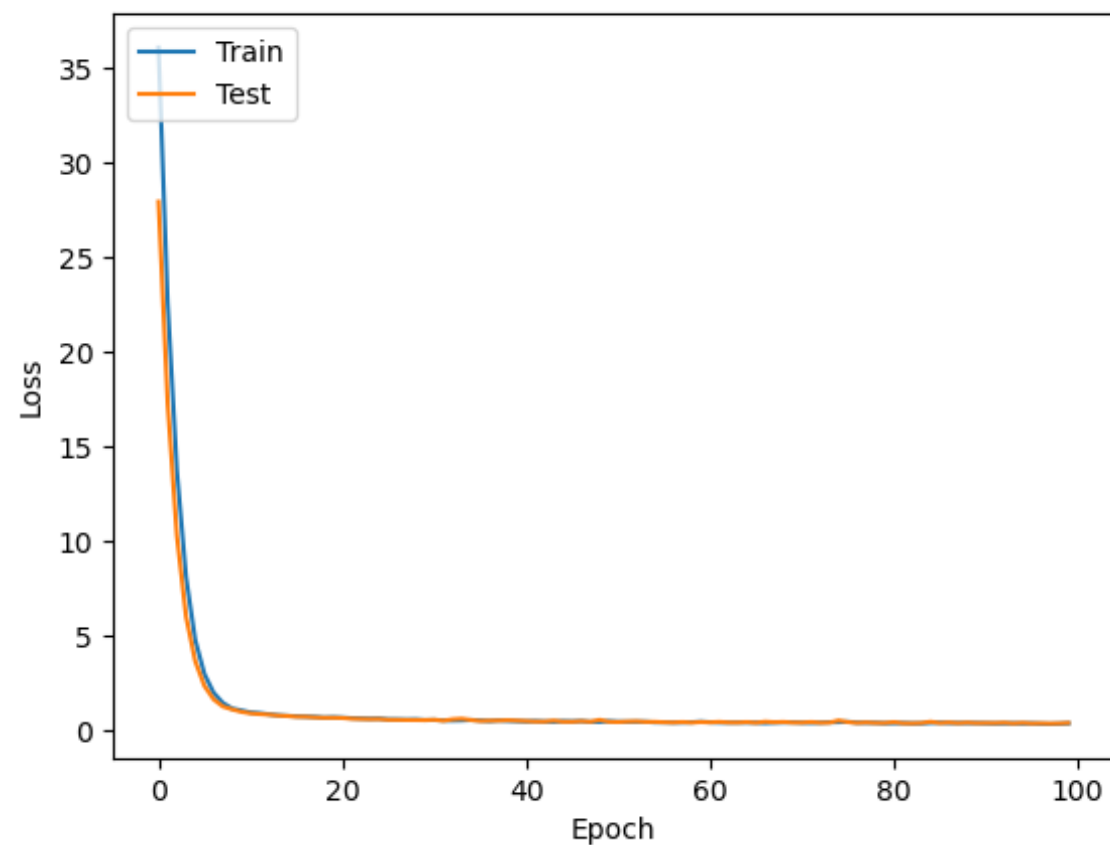
-30°

Deep Learning

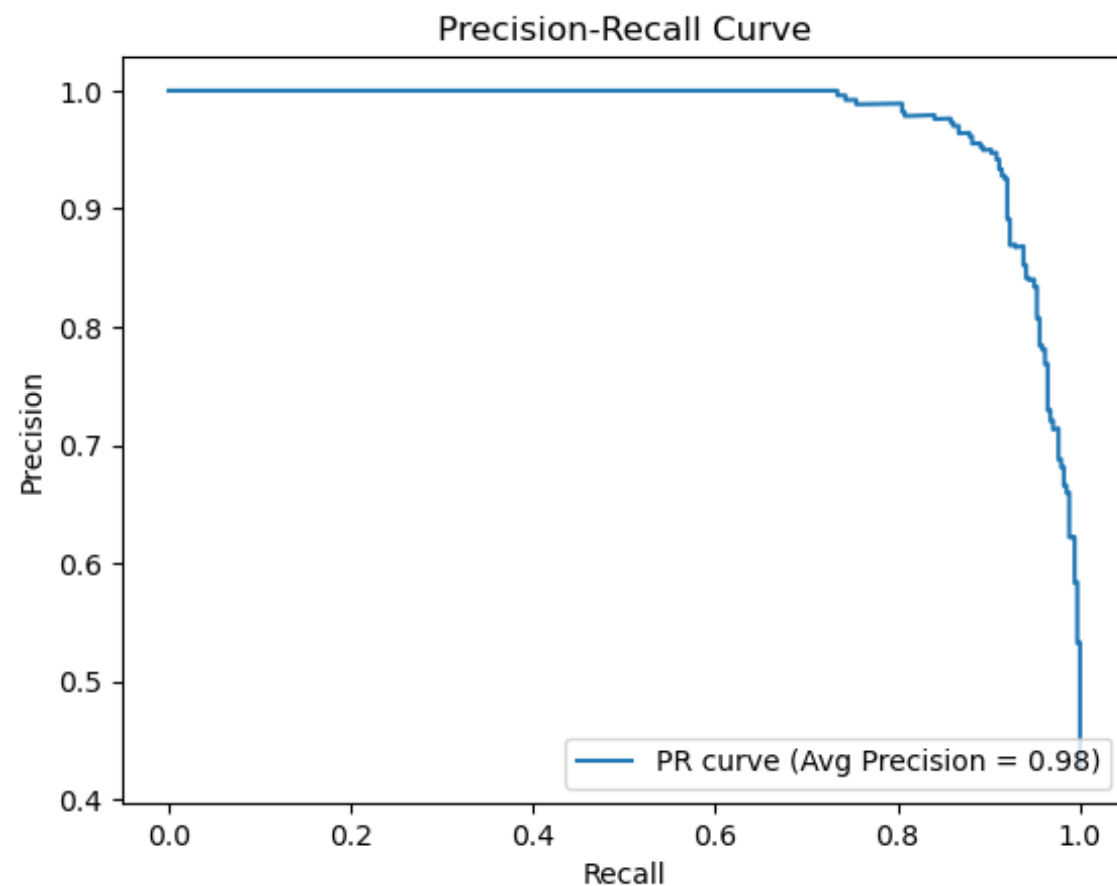
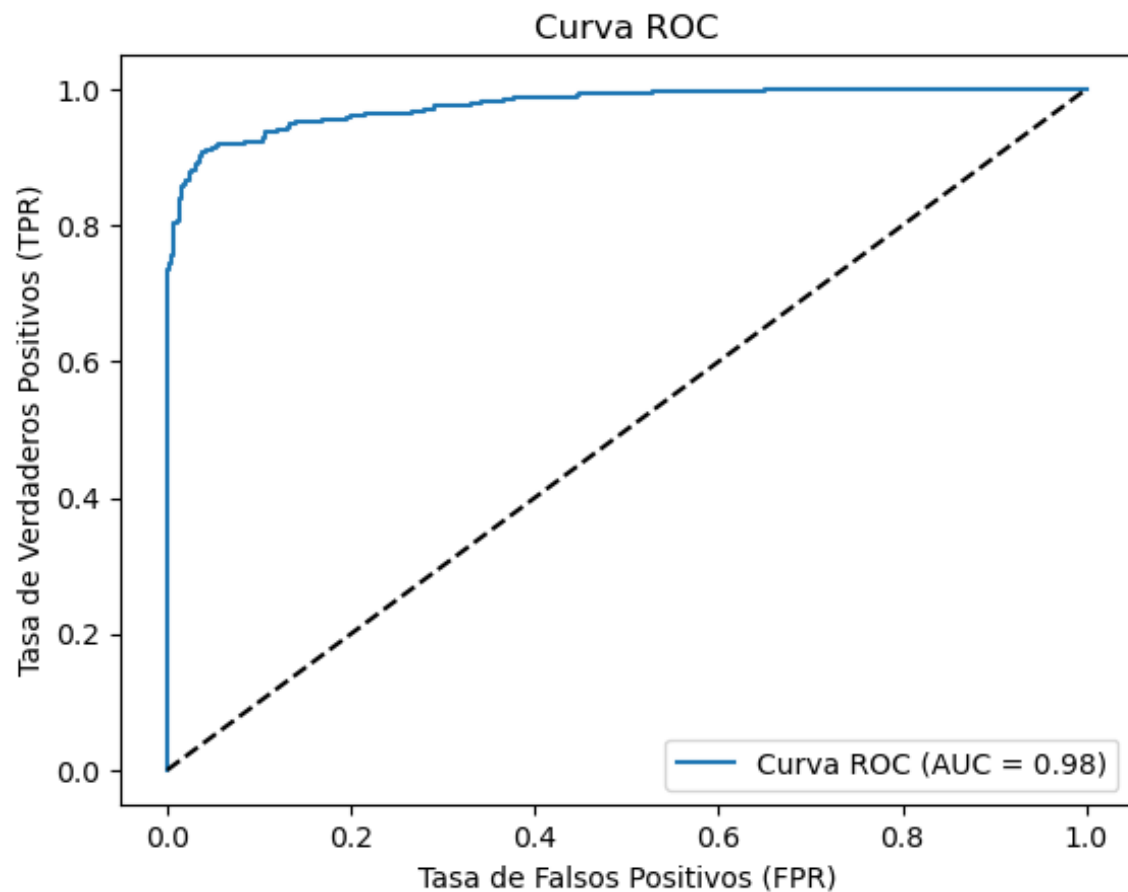
Model accuracy



Model loss



Deep Learning



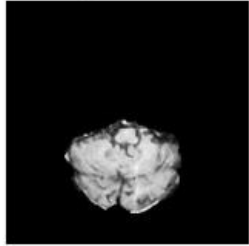
Resultados finales

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

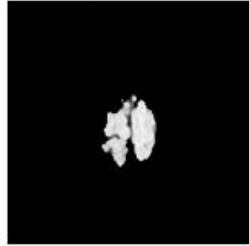
- Métricas para KNN mayores que para CNN.
- KNN requiere extracción de features previa, CNN la realiza internamente.
- Para ambos modelos, métricas mayores a 0.9.

Predictions Examples

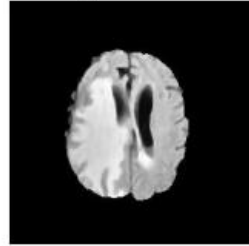
Tumor: No
Tumor prediction: No



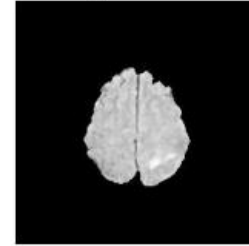
Tumor: No
Tumor prediction: No



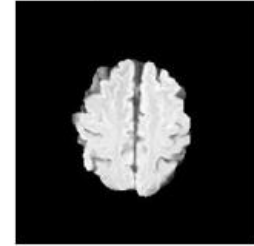
Tumor: Yes
Tumor prediction: Yes



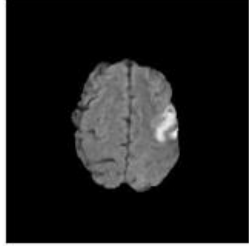
Tumor: Yes
Tumor prediction: No



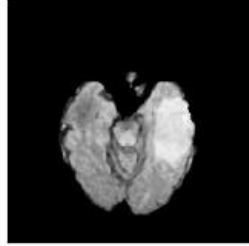
Tumor: No
Tumor prediction: No



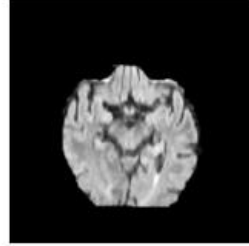
Tumor: Yes
Tumor prediction: Yes



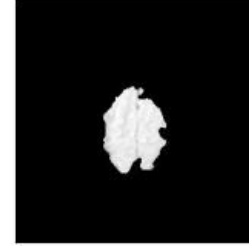
Tumor: Yes
Tumor prediction: Yes



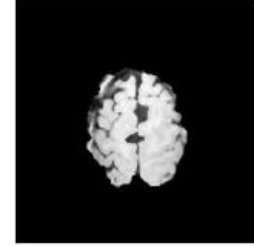
Tumor: No
Tumor prediction: Yes



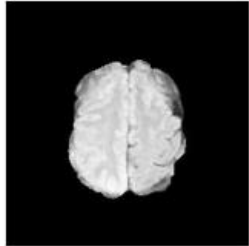
Tumor: No
Tumor prediction: No



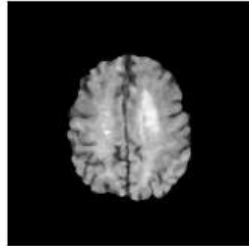
Tumor: No
Tumor prediction: No



Tumor: Yes
Tumor prediction: No



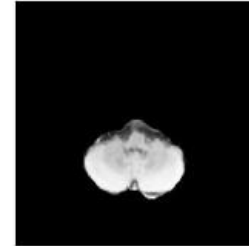
Tumor: Yes
Tumor prediction: Yes



Tumor: No
Tumor prediction: No



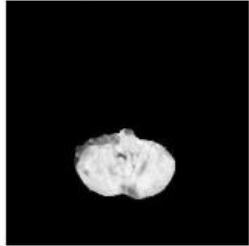
Tumor: No
Tumor prediction: No



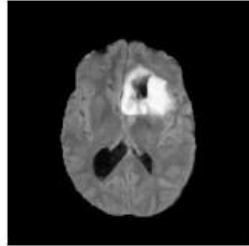
Tumor: No
Tumor prediction: No



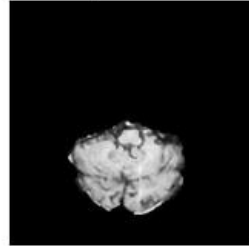
Tumor: No
Tumor prediction: No



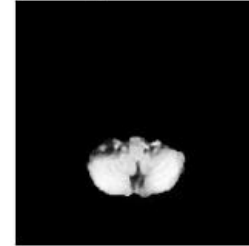
Tumor: Yes
Tumor prediction: Yes



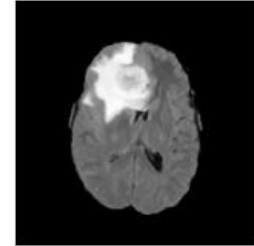
Tumor: No
Tumor prediction: No



Tumor: No
Tumor prediction: No



Tumor: Yes
Tumor prediction: Yes



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.

