BOOTCAMP EN DATA SCIENCE
Y MACHINE LEARNING

PROYECTO FINAL

Modelos de predicción para diagnóstico de tumores cerebrales

Presentado por:
Arturo Moreno Puga

Curso Académico 2022/2023

Índice

1 Introducción.

Este estudio consiste en la aplicación de modelos de Deep Learning y Machine Learning a un problema de predicción de tumores cerebrales en pacientes. Los datos disponibles se separan en dos bloques: un conjunto de 3762 imágenes de resonancias magnéticas cerebrales con distinción entre presencia o ausencia de tumor, y un dataset constituido por features correspondientes a distintas propiedades cuantitativas de las propias imágenes, con una variable target categórica binaria que asocia 0 a la ausencia de tumor y 1 a la presencia de tumor. En el estudio se aplican, de una parte, un modelo de red neuronal convolucional (CNN) al conjunto de imágenes (sección de Deep Learning), y de otra parte dos modelos de clasificación (*k-Nearest Neighbors* (KNN) y árbol de decisión) sobre el dataset numérico asociado (sección de Machine Learning).

El objetivo es, dada una imagen o sus features correspondientes, ser capaces de predecir con un margen de probabilidad aceptable si existe o no tumor cerebral. Se exponen los resultados más óptimos para cada modelo y se realiza una discusión comparativa con el fin de saber cuáles arrojan mejores resultados dependiendo de las necesidades de los estudios clínicos posteriores. Dichos resultados se basan en un objetivo claro a seguir durante la definición de los modelos: la optimización del *recall*, métrica definida como

$$recall = \frac{TP}{TP + FN} \tag{1}$$

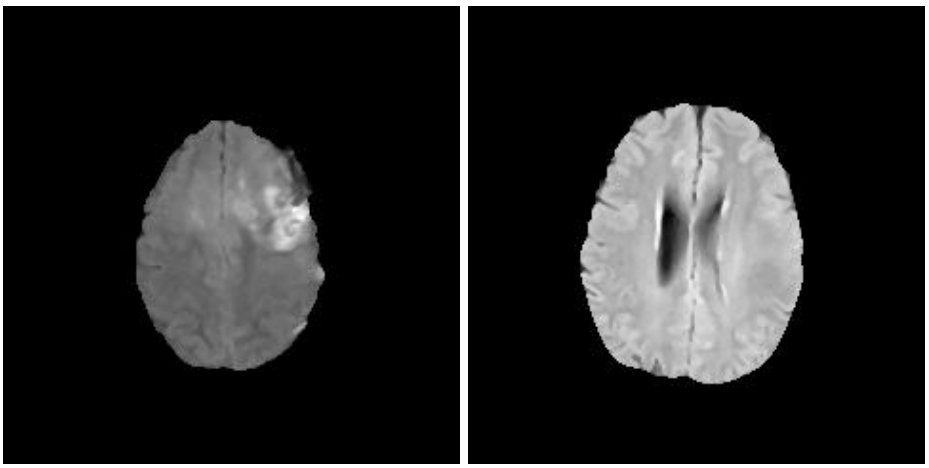
siendo TP $\equiv$ *True Positives* (Verdaderos Positivos, casos en los que la predicción de clase positiva coincide con la realidad de clase positiva) y FN $\equiv$ *False Negatives* (Falsos Negativos, casos en los que la predicción de clase negativa no coincide con la realidad de clase positiva). El *recall* informa de la proporción de Falsos Negativos, denotado por FN en la matriz de confusión (Tabla 1). Esta Tabla organiza los tipos de error que se pueden cometer al aplicar un modelo de predicción. En el caso de nuestro estudio, un Falso Negativo correspondería a un paciente que no ha sido diagnosticado con tumor cuando realmente sí lo tiene, por tanto es prioritario obtener el mínimo valor posible para FN ya que se trata de pacientes que, en el caso hipotético de que el modelo fuese un primer filtro para detectar resonancias con tumor, se le tendría como paciente sano cuando realmente no lo es. En el caso de un Falso Positivo, se obtendría una predicción positiva siendo la realidad que no existe presencia de tumor. Esto podría conducir a pruebas más exhaustivas que detectasen que no hay tumor, por tanto se solventaría. Por consiguiente, a la hora de definir los modelos hay que buscar obtener un valor aceptable para el *recall* sin afectar considerablemente al resto de métricas.

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

Tabla 1: Matriz de Confusión. TP $\equiv$ *True Positives* (Verdaderos Positivos), FN $\equiv$ *False Negatives* (Falsos Negativos), FP $\equiv$ *False Positives* (Falsos Positivos), TN $\equiv$ *True Negatives* (Verdaderos Negativos).

1.1 Imágenes y features del dataset.

Las imágenes disponibles corresponden a resonancias magnéticas cerebrales (MRI, *magnetic resonance imaging*) de pacientes que, en un 55.3 % de los casos presentan tumor y en un 44.7 % de los casos no lo presentan. En la Figura 1 se muestran dos ejemplos del conjunto de imágenes, una con tumor y otra sin tumor. Se trata de un conjunto de imágenes balanceado con objeto de aplicación puramente académica, teniéndose en un caso real la proporción entre presencia y ausencia de tumor probablemente más desbalanceada. Aun así, estas imágenes constituyen un buen conjunto ejemplo para la práctica en entrenamiento de modelos.



(a) (b)

Figura 1: Ejemplos de resonancias magnéticas con tumor (a) y sin tumor (b). Se observan las distintas tonalidades de gris a partir de las cuales se obtienen los features.

Asociados a cada imagen se dispone de distintos features que informan sobre la distribución de niveles de gris entre píxeles, organizados en un dataset. Se trata de features puramente numéricos y de dividen en dos grupos: features de Primer Orden (*mean, variance, standar deviation, kurtosis, skewness*) y features de Segundo Orden (*ASM (Angular Second Moment), contrast, correlation, homogeneity, entropy, dissmilarity, energy, coarseness*). Todos los mencionados son estadísticos asociados a las distintas distribuciones de intensidad de gris en las imágenes y se definen a continuación según el artículo de N. Aggarwal y R. K. Agrawal (2012) [1].

1.1.1 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

$$P(I) \equiv \frac{n^\circ \text{ de píxeles con nivel de gris } I}{n^\circ \text{ total de píxeles en la región}} \tag{2}$$

A partir de dicho histograma se definen los feautres de Primer Orden:

- **Mean:** Valor medio del histograma.

- **Variance y Standard Deviation:** Mide la desviación de los niveles de gris con respecto de la media. Da información sobre la anchura del histograma: a mayor varianza (que implica mayor desviación), mayor anchura.
- **Skewness:** Medida del grado de asimetría del histograma con respecto de la media.
- **Kurtosis:** Medida de la pronunciación de pico que muestra el histograma.

1.1.2 Features de Segundo Orden.

La configuración de niveles de gris de cada imagen se mide con cierta 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. A partir de esta matriz se definen los siguientes features:

$$ASM = \sum_{i,j} P(I_1\, I_2)^2 \tag{3}$$

$$Contrast = \sum_{I_1, I_2} |I_1 - I_2|^2 \log P(I_1\, I_2) \tag{4}$$

$$Correlation = \sum_{I_1, I_2} \frac{(I_1 - \mu_1)\,(I_2 - \mu_2)\,P(I_1\, I_2)}{\sigma_1 \sigma_2} \tag{5}$$

$$Homogeneity = \sum_{I_1, I_2} \frac{P(I_1\, I_2)}{1 + |I_1 - I_2|^2} \tag{6}$$

$$Entropy = -\sum_{I_1, I_2} P(I_1\, I_2) \log P(I_1\, I_2) \tag{7}$$

siendo μ_k y σ_k ciertos estadísticos [1]. Cada uno de estos features representa lo siguiente:

- **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.

Como se puede observar, varios de estos estadísticos están relacionados entre sí. Otros como *dissmilarity* y *energy* no obtienen explicación en la bibliografía usada. Aunque es ventajoso conocer de dónde proviene la definición de los features, el desconocimiento de estos dos últimos no supone diferencias a la hora de aplicar los modelos. Aun así, se puede intuir que la variable *dissmilarity* representa la falta de parecido entre píxeles, indicando valores más altos menor semejanza. La variable *coarseness* se eliminó del estudio puesto que muestra un valor constante de $7.458341 \cdot 10^{-155}$. Su consideración es innecesaria ya que un feautre de valor constante no proporciona diferenciación entre presencia y ausencia de tumor.

1.1.3 Features más relevantes.

Realizando una simple exploración gráfica se puede observar cómo las variables *ASM, energy, entropy* y *homogeneity* marcan una notable distinción entre ausencia o presencia de tumor (Figura 2).

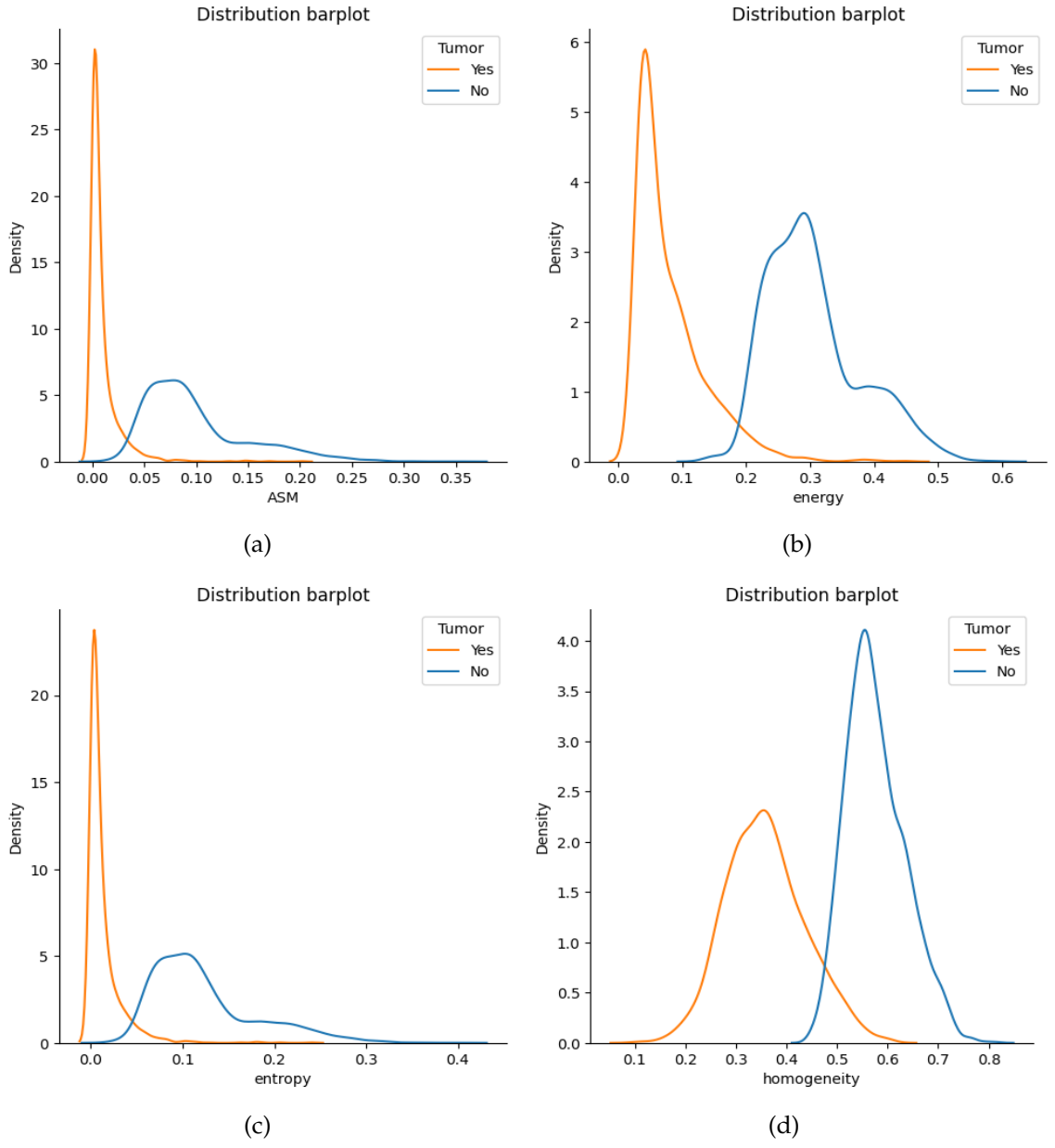


Figura 2: Gráficas de distribución para los features más relevantes, con distinción entre ausencia y presencia de tumor. Se grafican (a) *ASM*, (b) *energy*, (c) *entropy*, (d) *homogeneity*. Se puede observar la dependecia entre las variables *ASM* y *entropy*.

2 Modelos de clasificación. Resultados.

2.1 Deep Learning.

Con el fin de crear un modelo de predicción a partir de las propias imágenes se utiliza un modelo de red neuronal convolucional (CNN, del inglés *Convolutional Neural Network*) con las siguientes especificaciones:

- Arquitectura:
  - Capas convolucionales de 16, 32 y 64 filtros, todas con tamaño de kernel 3×3, strides de (2, 2), y función de activación ReLU (Rectified Linear Unit).
  - Capas de agrupamiento (MaxPooling2D) con tamaño de ventana 2×2.
  - Capa de aplanamiento (Flatten).
  - Capas completamente conectadas (Dense) de 256 neuronas y función de activación ReLU, 128 neuronas y función de activación ReLU y 1 neurona y función de activación sigmoide.
- Regularización L1 con factor de regularización 0.01, que evita el *overfitting*, aplicada en todas las capas convolucionales.
- Número de épocas: 100.
- Optimizador Adam con argumentos learning_rate = 0.001, beta_1 = 0.95, beta_2 = 0.9999, epsilon = 1e-10.
- Función de pérdida binary_crossentropy.
- Métrica accuracy utilizada para evaluar el rendimiento del modelo durante el entrenamiento.
- Aumento del conjunto de entrenamiento aplicando rotaciones de 30° y -30° a cada imagen.

Este modelo se ha implementando haciendo uso de la librería Keras.

Para calcular las métricas *recall, precision, accuracy* y *F1-score* es necesario escoger un umbral de probabilidad para realizar la clasificación. A la hora de decidir si una imagen tiene clase 0 o 1, se debe atender a cierto valor que indica con qué probabilidad la medida pertenecerá a cada clase. El umbral mencionado establece el valor de la probabilidad de predicción a partir del cual ésta se considerará de una clase u otra. La Figura 3 muestra las matrices de confusión obtenidas para distintos valores del umbral T. En cada matriz, el eje horizontal indica predicción y el eje vertical realidad. La elección del umbral adecuado se debe realizar tratando de buscar un equilibrio entre las distintas métricas y prestando especial atención a un *recall* aceptable. Por ejemplo, un valor de $T = 0.2$ presenta un valor de Falsos Negativos menor al de Falsos Positivos, con respectivas proporciones de 0.0340 % y 0.0378 % frente a los casos totales. Esto arroja los valores para las métricas dispuestos en la Tabla 2. Aun así, la elección del umbral más óptimo sería un proceso de discusión posterior en función del criterio y necesidades del usuario final del modelo.

Recall	Precision	Accuracy	F1-score
0.9201	0.9120	0.9281	0.9160

Tabla 2: Métricas para modelo CNN con un valor umbral de $T = 0.2$. Se define cada métrica en el Apéndice A.

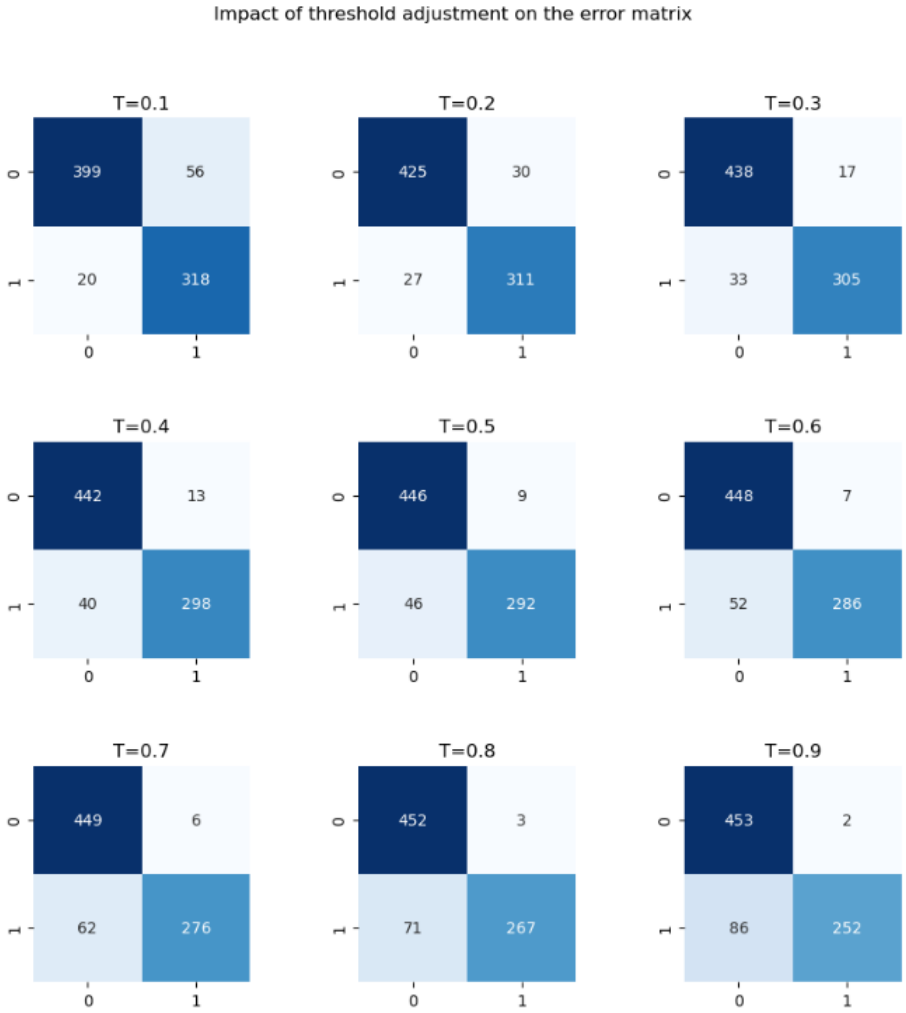


Figura 3: Matriz de confusión según distintos valores del umbral T.

En las Figuras 4 y 5 se pueden apreciar respectivamente la evolución del *accuracy* y de las pérdidas del modelo a medida que se ejecutan las épocas. En ambos casos se observa un buen rendimiento y es notable cómo la regularización L1 y los argumentos del optimizador Adam reducen enormemente el ruido en la gráfica de pérdidas, además de no producirse *overfitting*.

En la Figura 6 se muestra la curva ROC del modelo, indicando un valor de área bajo la curva de $AUC = 0.98$, muy cercana a la unidad, teniéndose por tanto un muy buen rendimiento. Esto también se puede notar en la separación que presenta la curva respecto a la diagonal. Así mismo, la Figura 7 indica también un buen rendimiento del modelo debido a la proximidad que adquiere la curva *precision-recall* al punto (1, 1), con un área de $AUC\text{-}PR = 0.98$.

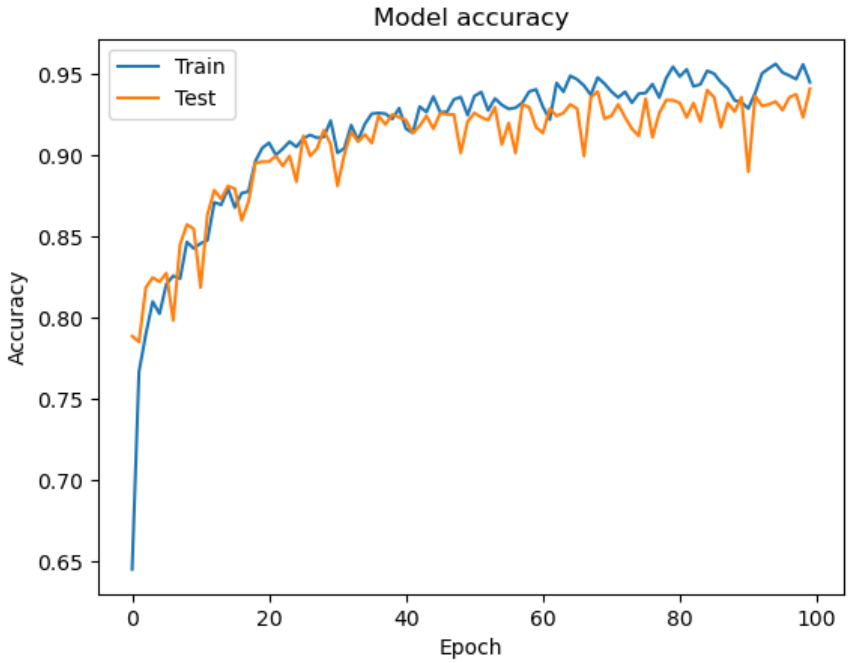


Figura 4: Evolución del *accuracy* en función de las épocas.

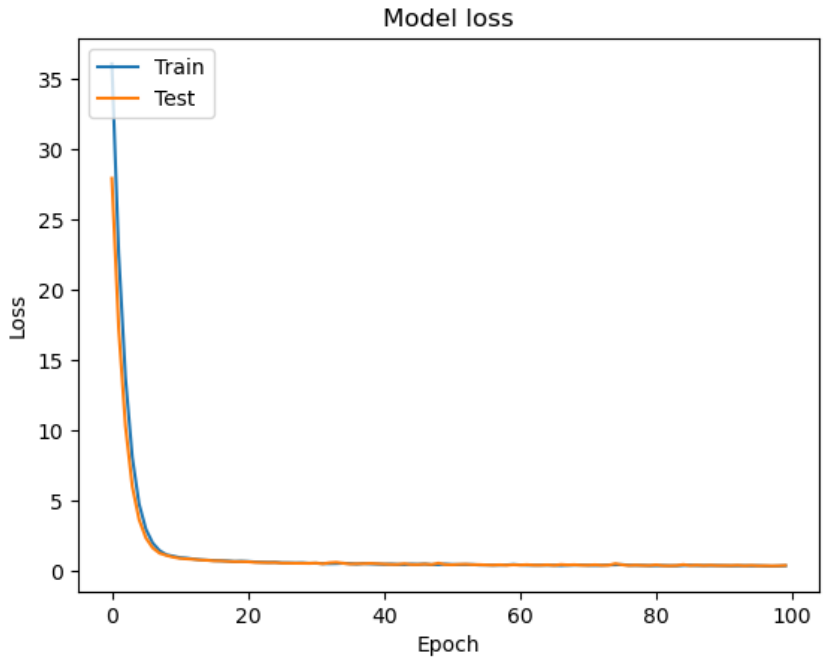


Figura 5: Evolución de las pérdidas en función de las épocas. Se puede apreciar cómo afecta la regularización L1.

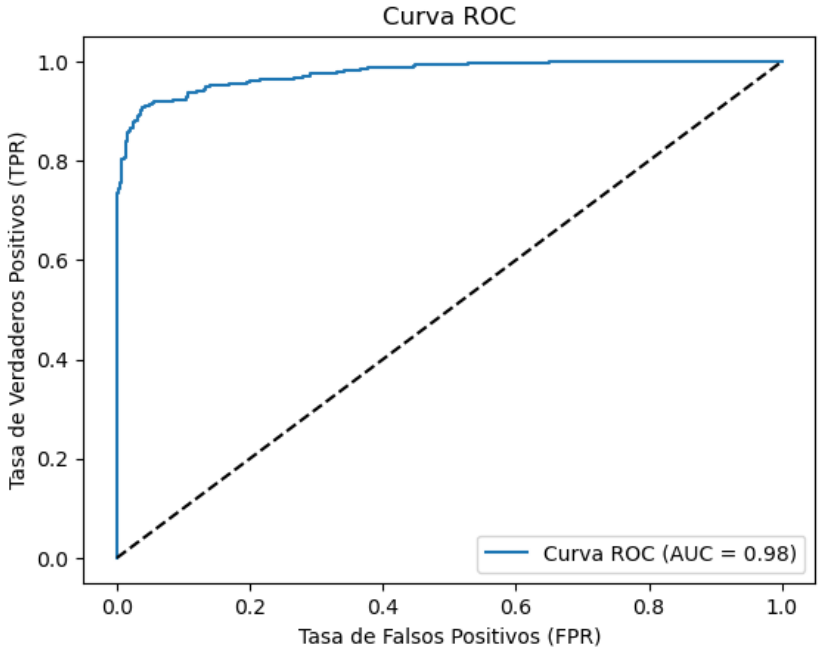


Figura 6: Curva ROC del modelo que representa la Tasa de Verdaderos Positivos (TPR) frente a la Tasa de Falsos Positivos (FPR), con un valor de $AUC = 0.98$. Se puede observar la clara separación de la curva respecto a la diagonal.

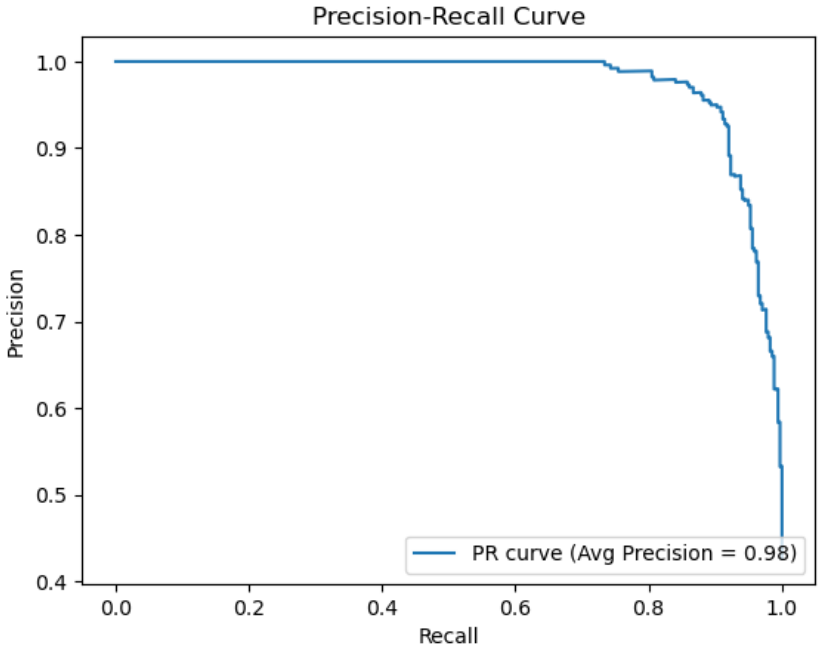


Figura 7: Curva *precision-recall* del modelo, con un área bajo la curva de $AUC\text{-}PR = 0.98$. Se puede observar cómo la gráfica se aproxima al punto (1, 1).

Cabe mencionar que el código utilizado considera en cada ejecución conjuntos de entrenamiento y testeo completamente aleatorios, pudiéndose obtener por tanto distintos resultados para distintas aplicaciones, mostrándose aquí los mejores obtenidos. Aun así,

no hay gran diferencia entre distintas pruebas, obteniéndose valores de áreas entre 0.96 y 0.98. El tiempo aproximado de entrenamiento del modelo con 100 épocas y demás especificaciones mencionadas es de 66 minutos.

2.2 Machine Learning.

A la hora de implementar modelos de Machine Learning con el dataset de features se optó por considerar KNN (*k-Nearest Neighbors*) y Árbol de Decisión (del inglés *Decision Tree*), con las siguientes especificaciones:

- KNN
  - n_neighbors = 1
  - weights = uniform
  - leaf_size = 10
- Decision Tree
  - criterion = gini
  - max_depth = 150

Dichos valores se obtuvieron aplicando una optimización de hiperparámetros con GridSearch, enfocado en optimizar el *recall*.

La evaluación del resultado obtenido en cada modelo se realiza atendiendo a las Tablas 3 y 4, prestando especial atención al *recall*. Se observa claramente cómo las métricas son todas mayores para el modelo KNN, por lo que se escoge como modelo más óptimo, con un *recall* de 0.9760, *precision* de 0.9939, *accuracy* de 0.9867 y *F1-score* de 0.9849.

Recall	Precision	Accuracy	F1-score
0.9760	0.9939	0.9867	0.9849

Tabla 3: Métricas para modelo KNN. Se define cada métrica en el Apéndice A.

Recall	Precision	Accuracy	F1-score
0.9671	0.9788	0.9761	0.9729

Tabla 4: Métricas para modelo Decision Tree. Se define cada métrica en el Apéndice A.

3 Discusión de los resultados.

Los resultados para el modelo KNN de Machine Learning presenta un mejor rendimiento frente al modelo CNN de Deep Learning atendiendo a las métricas *recall*, *precision*, *accuracy* y *F1-score*. Aun así, las demás métricas obtenidas para el modelo CNN indican muy buen rendimiento. Además hay que tener en cuenta que el modelo KNN requiere una extracción previa de features que no se ha realizado en este trabajo puesto que el dataset ya estaba formado. Esto supondría un añadido de gran importancia a la elaboración del modelo ya que los resultados dependen enormemente de dicha extracción. Es por esto que el modelo CNN supone una aplicación mucho más sencilla debido a que dicho proceso se realiza internamente.

Por otra parte, estos resultados poseen tan buen rendimiento y presentan métricas tan aparentemente satisfactorias debido a que se está trabajando sobre un dataset balanceado. En un caso real, se trataría de un dataset desbalanceado puesto que la presencia de tumor cerebral no es un diagnóstico muy común, y se debería comprobar en dicho caso cómo rinden los modelos para estudiar si sería preciso considerar distintas optimizaciones.

A modo ilustrativo, en la Figura 8 se muestra una selección aleatoria de 20 predicciones, mostrando la imagen asociada y clases real y predicha.

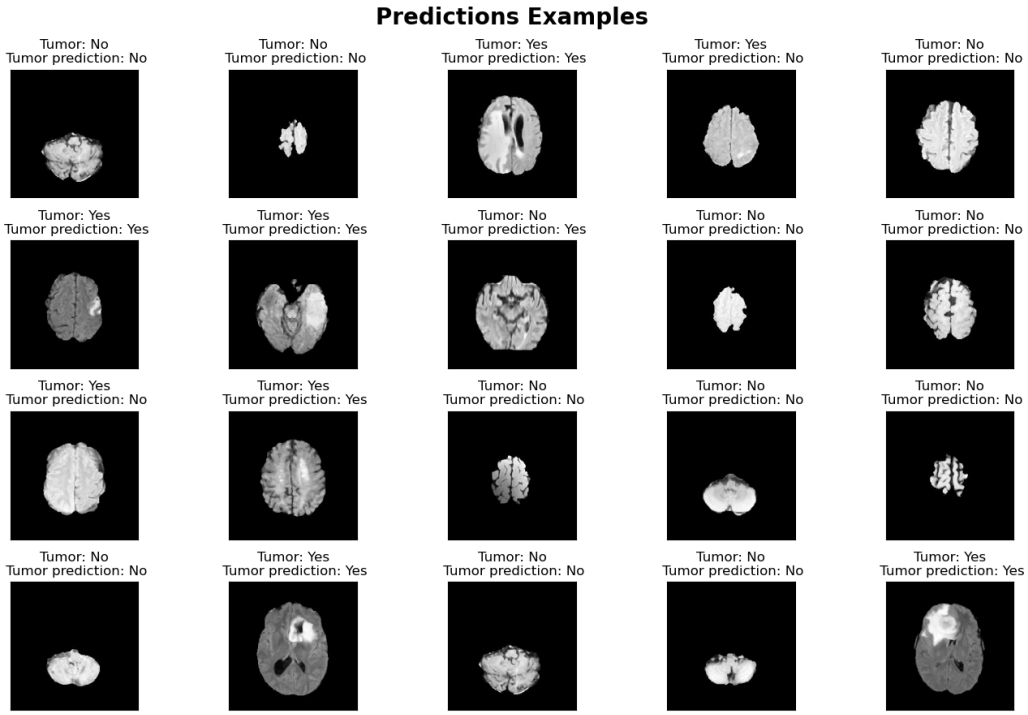


Figura 8: Ejemplos de predicciones aleatorias. Se pueden observar los dos tipos de errores: clase predicha como positiva cuando en realidad es negativa (Falso Positivo, segunda fila tercera columna) y clase predicha como negativa cuando en realidad es positiva (Falso Negativo, tercera fila primera columna).

4 Conclusiones.

Este trabajo supone un breve estudio sobre la aplicación de distintos modelos de aprendizaje (KNN, *Decision Tree* y CNN) sobre un problema de determinación de tumores, centrándonos en KNN y CNN. En definitiva, ambos modelos arrojan resultados satisfactorios y si se continuase el estudio se podría requerir, por ejemplo, un estudio de la extracción de features para el modelo KNN, así como un análisis donde se pudiese apreciar el efecto de un dataset menos balanceado, lo cual mostraría el rendimiento en un caso más próximo a la realidad, sobre todo en la aplicación del modelo CNN.

En última instancia, el modelo CNN supone un modelo mucho más utilizado en el ámbito de la Medicina debido a la simplicidad de su aplicación, aun siendo un modelo conceptualmente más complejo.

Referencias

[1] Aggarwal, N., Agrawal, R. K. (2012). First and Second Order Statistics Features for Classification of Magnetic Resonance Brain Images. *Journal of Signal and Information Processing*, Vol. 3 No. 2, pp. 146-153.
doi:10.4236/jsip.2012.32019

A Métricas de un modelo de predicción.

Sea la matriz de confusión dispuesta en la Tabla 1, se definen las métricas *recall*, *precision*, *accuracy* y *F1-score* como

$$recall = \frac{TP}{TP + FN} \tag{8}$$

$$precision = \frac{TP}{TP + FP} \tag{9}$$

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{10}$$

$$F1\text{-}score = 2 \cdot \frac{precision \cdot recall}{precision + recall} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{11}$$