Escuela Superior
de Ingeniería y Tecnología
Universidad de La Laguna

Trabajo de Fin de Grado

Mejora de diagnóstico de enfermedades neurodegenerativas usando redes LSTM

Enhancing the Diagnosis of Neurodegenerative Diseases using LSTM Networks

Daniel Dóniz García

La Laguna, 14 de julio de 2023

D. **Patricio García Báez**, con N.I.F. 43356987D profesor Contratado Doctor adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutor

D. **Carmen Paz Suárez Araujo**, con N.I.F. 43640373N profesora Catedrática de Universidad adscrita al Departamento de Informática y Sistemas de la Universidad de Las Palmas de Gran Canaria, como cotutora

C E R T I F I C A (N)

Que la presente memoria titulada:

”*Mejora de diagnóstico de enfermedades neurodegenerativas usando redes LSTM*”

ha sido realizada bajo su dirección por D. **Daniel Dóniz García**, con N.I.F. 45896100-Z.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 14 de julio de 2023

Agradecimientos

Quiero agradecer a todas las personas que han estado a mi lado en el transcurso de mi formación académica. Mi reconocimiento y agradecimiento profundo para mis padres, quienes nunca dejaron de apoyarme y confiar en mí durante todo el proceso de mi carrera. Agradezco igualmente a mis amigos, que convirtieron el camino académico en una experiencia agradable y memorable. Mi gratitud se extiende a todos los tutores y profesores que tuve durante mis estudios, por la valiosa educación y conocimientos que me han proporcionado y su incondicional respaldo. Finalmente, deseo reconocer especialmente a mi tutor de TFG, por su dedicación y esfuerzo para ayudarme a llevar adelante este proyecto, por su orientación y consejo que me permitieron desarrollarlo de manera efectiva.

Licencia

Resumen

Se presenta un enfoque para el diagnóstico y pronóstico de la enfermedad de Alzheimer utilizando Redes Neuronales de Largo Corto Plazo (LSTM). Las enfermedades neurodegenerativas, y en particular la enfermedad de Alzheimer, representan un gran desafío en la sociedad actual. En esta enfermedad se utilizan pruebas como la Alzheimer's Disease Assessment Scale (ADAS) y el Mini-Mental State Examination (MMSE) para evaluar la función cognitiva y el grado de demencia en los pacientes. No obstante, el diagnóstico y pronóstico precisos siguen siendo desafiantes. En este trabajo se mejora una anterior solución de inteligencia artificial obteniéndose resultados más fiables mediante distintas técnicas de aprendizaje automático. Dicha solución trabaja con los resultados de las pruebas ADAS y MMSE obtenidas de 1 a 5 citas médicas, mas otros datos demográficos, y proporciona un diagnóstico en tres posibles categorías: Cognición Normal (CN), Deterioro Cognitivo Leve (MCI) o Demencia (Dem). Este enfoque tiene el potencial de mejorar la precisión del diagnóstico y pronóstico de la enfermedad de Alzheimer, ofreciendo un impacto significativo en el cuidado y tratamiento de los pacientes.

Palabras clave: Enfermedad de Alzheimer, Diagnóstico, LSTM, ADAS, MMSE , Inteligencia Artificial, Aprendizaje Automático, Selección de Características, Balanceo, Validación cruzada, Deterioro Cognitivo Leve, Demencia.

Abstract

We introduce an approach for the diagnosis and prognosis of Alzheimer's Disease utilising Long Short-Term Memory (LSTM) Neural Networks. Neurodegenerative diseases, and specifically Alzheimer's Disease, pose a significant challenge in today's society. In this disease, tests such as the Alzheimer's Disease Assessment Scale (ADAS) and the Mini-Mental State Examination (MMSE) are utilised to assess cognitive function and the degree of dementia in patients. However, accurate diagnosis and prognosis remain challenging. In this work, we enhance a previous artificial intelligence solution, achieving more reliable results through various machine learning techniques. This solution works with the results of the ADAS and MMSE tests obtained from 1 to 5 medical appointments, along with other demographic data, and provides a diagnosis in three possible categories: Normal Cognition (CN), Mild Cognitive Impairment (MCI), or Dementia (Dem). This approach has the potential to improve the accuracy of the diagnosis and prognosis of Alzheimer's Disease, offering a significant impact on the care and treatment of patients.

Keywords: Alzheimer's Disease, Diagnosis, LSTM, ADAS, MMSE, Artificial Intelligence, Machine Learning, Feature Selection, Balancing, Cross-validation, Mild Cognitive Impairment, Dementia.

Índice general

Índice de Figuras

Índice de Tablas

Capítulo 1

Introducción

Las enfermedades neurodegenerativas se caracterizan por la degeneración progresiva de la estructura y función del sistema nervioso central o periférico. Ejemplos notables de estas enfermedades incluyen el Alzheimer y el Parkinson. Estos trastornos están aumentando en prevalencia, donde la OMS aporta que a nivel mundial en 2015 el alzhéimer y otras demencias afectan a 47 millones de personas en todo el mundo [1], y esta tendencia se espera que continúe debido al aumento de la esperanza de vida de la población. Este incremento está causando un impacto significativo en la salud pública, ya que estas enfermedades se vuelven más difíciles y costosas de tratar en sus etapas avanzadas.

El enfoque médico actual para la detección de estas enfermedades se fundamenta principalmente en la evaluación del historial clínico del paciente y la realización de pruebas cognitivas básicas. Sin embargo, la manifestación variada de estas enfermedades, que aún no se comprende completamente, resulta en una precisión diagnóstica limitada para estos métodos, especialmente en atención primaria. Pruebas más precisas, como la resonancia magnética, son menos accesibles y más costosas.

Por lo tanto, se propone el uso de redes neuronales como una posible solución a este problema. Las redes neuronales tienen la capacidad de generar relaciones entre variables y de aprender, lo que las hace potencialmente útiles para mejorar la detección temprana de deterioro cognitivo. De esta manera, es posible aplicar un tratamiento antes de que el deterioro se convierta en una enfermedad más avanzada, como el Alzheimer o el Parkinson, proporcionando una mayor esperanza de recuperación para los pacientes.

Este trabajo se basa en una ampliación de otro trabajo previo realizado por un ex alumno Diego Hugo Hamilton López [2]. En el cual se desarrollaron pronósticos utilizando redes neuronales LSTM. En este trabajo, se explorará el uso de estas redes para mejorar la detección temprana de trastornos cognitivos y diagnosticar de manera más precisa la condición de salud cognitiva de los pacientes utilizando datos de múltiples citas médicas, para ello se incorporarán nuevas técnicas y enfoques de aprendizaje automático. Al expandir el trabajo anterior, se espera obtener resultados más precisos y brindar nuevas perspectivas en el campo del diagnóstico y pronóstico de enfermedades neurodegenerativas.

1.1. Objetivos

1.1.1. Objetivo general del proyecto

El enfoque principal de este trabajo es mejorar los resultados obtenidos en el trabajo inicial de Hamilton López [2] mediante la implementación de técnicas avanzadas de selección de características y la utilización de herramientas especializadas que optimicen el proceso de entrenamiento del modelo. El objetivo es aumentar la precisión y la eficiencia del sistema de diagnóstico desarrollado, permitiendo así una detección más temprana y precisa de las condiciones de salud cognitiva. Para lograr esto, se explorarán diversas estrategias de selección de características para identificar las variables más relevantes y se emplearán herramientas de vanguardia en el campo de la inteligencia artificial y el aprendizaje automático para mejorar la capacidad predictiva del modelo. A través de estas mejoras, se espera obtener resultados superiores que contribuyan al avance en el diagnóstico y tratamiento de enfermedades neurodegenerativas.

1.1.2. Objetivos específicos del proyecto

Para realizar efectivamente el objetivo general del proyecto, se identifican los siguientes objetivos específicos:

1. Selección de características: La entrada a la red neuronal consiste principalmente en puntuaciones obtenidas de pruebas neuropsicológicas. El objetivo es extraer las características más informativas de los atributos disponibles en la base de datos ADNI.

2. Preprocesamiento de datos: El conjunto de datos inicial será sometido a un análisis exhaustivo de su equilibrio y particionado. Este proceso permite realizar comparaciones significativas en diferentes áreas problemáticas.

3. Desarrollo de un sistema de diagnóstico: El objetivo es crear un sistema que pueda diagnosticar automáticamente la condición de un paciente en base a los resultados de las pruebas realizadas en múltiples citas.

4. Mejorar fiabilidad de los resultados: Para obtener estimaciones más confiables del rendimiento de los sistemas desarrollados, se realizarán múltiples entrenamientos con diferentes divisiones de validación y entrenamiento. Este enfoque permite realizar evaluaciones más robustas y posibilita la creación de un sistema de conjunto utilizando un esquema de votación mayoritaria, donde se selecciona el resultado más frecuente de los diferentes entrenamientos.

5. Búsqueda del número de citas más adecuado para realizar la predicción: Con el fin de maximizar la eficacia del sistema de diagnóstico, se investigará el número óptimo de citas que deben considerarse para realizar predicciones. Este proceso implicará el análisis de diferentes configuraciones y la comparación de su rendimiento predictivo. Al determinar el número adecuado de citas, podemos mejorar la precisión del modelo y proporcionar diagnósticos más precisos y significativos.

6. Detención temprana (early stopping): Para mejorar la eficiencia del entrenamiento, se empleará la técnica de detención temprana. Este método ayuda a prevenir el

sobreajuste y reduce el tiempo de computación. El entrenamiento se detendrá cuando el modelo comience a sobreajustarse, lo cual se evidencia por una mejora en el rendimiento con el conjunto de entrenamiento pero un deterioro en el conjunto de validación.

7. Creación de un ensemble de modelos: Para mejorar aún más la eficacia y la robustez del sistema de diagnóstico, se implementará un ensemble de modelos. Un ensemble es un enfoque de aprendizaje automático que combina las predicciones de múltiples modelos para formar una predicción final más precisa. La idea detrás del ensemble es que, mientras un solo modelo puede ser excepcionalmente bueno en ciertas tareas o en ciertos tipos de datos, un grupo de modelos puede ser capaz de generalizar mejor y proporcionar una precisión más consistente en una gama más amplia de datos y tareas.

Al abordar estos objetivos, el proyecto busca mejorar el rendimiento general del sistema de diagnóstico y pronóstico, mejorando su precisión y confiabilidad en la predicción de condiciones de salud cognitiva.

1.2. Estado del arte

En la actualidad, el diagnóstico de enfermedades neurodegenerativas depende en gran medida de pruebas clínicas y evaluaciones cognitivas. Las pruebas como el Alzheimer's Disease Neuroimaging Initiative (ADNI)[34] y el Mini Mental State Examination (MMSE)[6] son comúnmente utilizadas en citas médicas para evaluar la función cognitiva y detectar signos tempranos de deterioro [33, 22]. Estas pruebas actúan como herramientas de cribado, proporcionando una indicación inicial de la posibilidad de una enfermedad neurodegenerativa [27].

Varios trabajos recientes han explorado el uso de técnicas de aprendizaje profundo, en particular, las redes LSTM (Long Short-Term Memory), para el diagnóstico de enfermedades neurodegenerativas. Estos estudios han demostrado que las LSTM pueden ser efectivas en la detección temprana de estas enfermedades, superando a menudo los métodos de diagnóstico tradicionales [10]. Sin embargo, el uso de estas técnicas todavía está en sus primeras etapas y se necesita más investigación para validar su eficacia en un entorno clínico [14]. Además, es relevante mencionar el trabajo realizado por Diego Hugo Hamilton López en su Trabajo Final de Grado (TFG) [2]. En dicho trabajo, López exploró la utilidad de las redes LSTM en el diagnóstico de enfermedades neurodegenerativas, y las conclusiones resultantes tienen un gran impacto en el uso de la inteligencia artificial en medicina [2]. López observó que la eficacia de las redes LSTM aumentaba con la cantidad de historial médico de un paciente que se introducía en la red, aunque esto solo ocurría hasta cierto punto.

Se descubrió que una red que agrupaba cinco citas de un paciente a menudo mostraba un rendimiento inferior, posiblemente debido a la falta de datos. Al aumentar el número de citas por agrupación, el número total de agrupaciones y la cantidad de información disponible disminuían, lo que podría afectar negativamente el rendimiento. Este hallazgo sugiere que es necesario realizar más estudios con un conjunto de datos más grande para determinar si realmente cuatro citas por agrupación ofrecen el mejor rendimiento.

López también destacó la dificultad de distinguir entre trastornos cognitivos leves de diferentes grados de severidad. Los límites entre estos trastornos son muy finos y

pueden diferenciarse por pequeñas variaciones en los resultados de los test. Esto plantea problemas debido a errores humanos en la realización de los test o la variabilidad entre los pacientes.

A pesar de estas limitaciones, López concluyó que las redes LSTM pueden ser útiles en la detección temprana de enfermedades neurodegenerativas. Aunque la precisión de las redes no es suficiente para proporcionar diagnósticos con alta fiabilidad, pueden ser útiles para proporcionar diagnósticos preventivos. En situaciones en las que se detecta un posible diagnóstico adverso, se podrían realizar pruebas más exhaustivas para aclarar el estado del paciente. De este modo, las redes LSTM podrían funcionar como un sistema de cribado inicial, dada la facilidad de los test necesarios para obtener un diagnóstico de la red.

Capítulo 2

Método y herramientas utilizadas

En el capítulo anterior se presentó la introducción al proyecto. En este capítulo, discutiremos las herramientas software utilizadas, la estructura de la red LSTM, diversos métodos para la mejora del aprendizaje automático y la evaluación de los resultados.

2.1. Herramientas de software

Las herramientas software utilizadas durante el desarrollo del Trabajo de Fin de Grado (TFG) y las cuales permitieron obtener los resultados son las siguientes:

- **"Jupyter Notebook y "Paperspace"** [23][21]: Se seleccionó Jupyter Notebook como el entorno de desarrollo para la implementación del programa. Jupyter Notebook es una aplicación web que facilita la creación e intercambio de documentos, conocidos como cuadernos Jupyter, que incorporan código en vivo, ecuaciones, visualizaciones y texto descriptivo. Estos cuadernos se emplean frecuentemente en tareas tales como la limpieza y transformación de datos, simulación numérica, modelado estadístico, visualización de datos y aprendizaje automático.

  En lugar de optar por un entorno local como Anaconda [4], se utilizó la plataforma de computación en la nube Paperspace [21]. Esta plataforma proporciona una variedad de servicios que incluyen máquinas virtuales (MV), almacenamiento y un entorno de trabajo colaborativo. La elección de Paperspace se debe a su eficiencia en la gestión de recursos requeridos para el desarrollo del proyecto, superando las limitaciones inherentes al hardware local.

  Paperspace brinda acceso a máquinas virtuales de alto rendimiento, ideales para tareas de procesamiento intensivo, como las que implican el uso de aprendizaje automático. Permite la instalación de cuadernos Jupyter directamente en la MV, facilitando así la ejecución y el intercambio de trabajo. Al igual que con Anaconda, con Paperspace es posible gestionar las versiones y dependencias de los paquetes, pero con la ventaja de no estar limitado por los recursos de hardware local.

- **Entorno Python** [13]: Python es un lenguaje de programación interpretado, de alto nivel y de propósito general. Con un enfoque en la legibilidad del código y su sintaxis sencilla, Python permite a los programadores expresar conceptos en menos líneas de código que otros lenguajes como C++ o Java. La simplicidad de Python lo hace especialmente atractivo para los principiantes en programación, pero su potencia y flexibilidad también lo hacen popular entre los profesionales para una variedad

de aplicaciones, desde el desarrollo web hasta la ciencia de datos y el aprendizaje automático.

Una de las fortalezas de Python es su extensa colección de bibliotecas y módulos, que extienden su funcionalidad base para cubrir una amplia gama de usos. Estas bibliotecas, a menudo mantenidas por la comunidad, proporcionan herramientas y funciones preescritas que pueden ahorrar a los programadores mucho tiempo de desarrollo.

En el contexto de este trabajo, las librerías más relevantes utilizadas fueron:

- **pandas** [28]: Proporciona estructuras de datos y herramientas de análisis de datos. Útil para la manipulación y limpieza de datos, ofreciendo funciones para leer, escribir y manipular conjuntos de datos en diferentes formatos.
- **Scikit-learn (sklearn)** [32]: Biblioteca para aprendizaje de máquina que proporciona herramientas para la división de datos (train_test_split), codificación de variables categóricas (LabelEncoder) y evaluación de modelos (metrics).
- **XGBoost (xgboost)** [9]: Implementa el algoritmo XGBoost para aprendizaje automático basado en árboles de decisión optimizados (XGBClassifier) y ofrece una función para trazar la importancia de las características en un modelo entrenado (plot_importance).
- **Keras** [36]: Biblioteca de aprendizaje profundo de alto nivel. Se utiliza para desarrollar y entrenar modelos de aprendizaje automático, incluyendo la implementación de redes neuronales recurrentes como las LSTM (Long Short-Term Memory).
- **TensorFlow (tensorflow)** [37]: Biblioteca de código abierto para aprendizaje automático y aprendizaje profundo que ofrece optimizadores (tensorflow.keras.optimizers), funciones de pérdida (tensorflow.keras.losses) y utilidades para el entrenamiento de modelos de aprendizaje automático (tensorflow.keras.utils).

2.2. Modelo de Redes Neuronales LSTM

Las redes neuronales LSTM (Long Short-Term Memory), introducidas por Hochreiter y Schmidhuber en su artículo ”Long Short-Term Memory” [20], son un tipo especial de redes neuronales recurrentes que han demostrado ser muy eficaces en el procesamiento de datos secuenciales, como en el caso de análisis de series de tiempo o datos longitudinales. Estas redes se han utilizado con éxito en una amplia gama de aplicaciones, incluyendo el procesamiento de lenguaje natural, reconocimiento de voz, traducción automática y análisis de datos biomédicos [16, 35].

La principal ventaja de las redes LSTM radica en su capacidad para capturar dependencias a largo plazo en los datos secuenciales. A diferencia de las redes neuronales recurrentes convencionales, que pueden tener dificultades para recordar información relevante de eventos pasados en secuencias largas, las LSTM están diseñadas para mitigar el problema del desvanecimiento y la explosión del gradiente que ocurre en el entrenamiento de redes neuronales recurrentes. Esto se logra mediante la incorporación de una estructura de memoria interna en cada unidad LSTM [20].

Las unidades LSTM constan de varias compuertas que permiten regular el flujo de información a través de la red. Estas compuertas, como la compuerta ”forget” (olvido), la compuerta ”update” (actualización) y la compuerta ”output” (salida), controlan qué información se debe olvidar, qué información nueva se debe agregar y qué información se debe transmitir como salida en cada paso de tiempo [15]. Además, las celdas LSTM tienen la capacidad de mantener y actualizar un estado de memoria a largo plazo, lo que les permite recordar información relevante a lo largo de secuencias largas [20].

En el contexto de este trabajo, las redes LSTM se utilizan para el análisis de datos longitudinales relacionados con la salud cognitiva de los pacientes. Esto implica el uso de datos de múltiples citas médicas, en las cuales se han realizado pruebas neuropsicológicas para evaluar el estado cognitivo de los pacientes. Las redes LSTM son capaces de aprender patrones y relaciones complejas en estos datos secuenciales, lo que les permite predecir la condición de salud cognitiva de un individuo en base a las puntuaciones de las pruebas obtenidas en citas anteriores [38, 39].

2.3. Selección de características

La selección de características es un proceso fundamental en el análisis de datos y el aprendizaje automático. Consiste en identificar y elegir las variables más relevantes o informativas para un determinado problema [17]. El objetivo de la selección de características es reducir la dimensionalidad de los datos, eliminar la redundancia y mejorar la precisión y el rendimiento de los modelos predictivos.

Existen diferentes métodos de selección de características, incluyendo técnicas univariadas, técnicas basadas en modelos y técnicas de aprendizaje automático [26]. Estas técnicas evalúan las características de acuerdo a criterios como la correlación con la variable objetivo, la importancia en el modelo o la capacidad predictiva.

La selección de características aporta varios beneficios. En primer lugar, reduce la complejidad del modelo al eliminar características irrelevantes o redundantes, lo que ayuda a evitar el sobreajuste y mejora la interpretabilidad del modelo [24]. Además, al reducir la dimensionalidad, se pueden obtener modelos más eficientes y de menor costo computacional. Por último, la selección de características puede mejorar la precisión y el rendimiento del modelo al utilizar solo las características más informativas [26].En este trabajo se ha utilizado la variante XGBoost de selección de características, que describimos en la siguiente sección

2.3.1. Modelo XGBoost

XGBoost es un algoritmo de aprendizaje automático que, aunque no es específicamente un método de selección de variables, tiene propiedades que permiten identificar y seleccionar las variables más importantes en un conjunto de datos [8].

A través del entrenamiento de árboles de decisión, XGBoost tiene la capacidad de calcular la importancia de las variables. Esta importancia se mide en términos de la cantidad de veces que una variable se utiliza para dividir los datos, y el grado en que estas divisiones contribuyen a la mejora del modelo.

Estas medidas de importancia de las variables pueden ser utilizadas para la selección de variables. Al ordenar las variables según su importancia, se puede seleccionar un subconjunto de variables para utilizar en el modelo, descartando las menos importantes.

Este enfoque puede ayudar a reducir la complejidad del modelo, mejorar la eficiencia computacional y posiblemente mejorar la capacidad de generalización del modelo.

Además, la importancia de las variables en XGBoost se calcula de manera intrínseca durante el entrenamiento del modelo, lo que significa que no se necesita un paso de selección de variables separado. Esto hace que el uso de XGBoost para la selección de variables sea eficiente y conveniente.

Es importante tener en cuenta que la selección de variables basada en XGBoost asume que el modelo XGBoost es una buena representación de la relación entre las variables y la respuesta. Si este no es el caso, la importancia de las variables calculada por XGBoost puede no ser una guía precisa para la selección de variables.

En resumen, aunque XGBoost no es una técnica de selección de variables en el sentido tradicional, su capacidad para calcular la importancia de las variables puede ser una herramienta útil para identificar las variables más relevantes en un conjunto de datos.

2.4. Balanceo de datos

El balanceo de datos se refiere al proceso de ajustar la distribución de las clases en un conjunto de datos desequilibrado. En un conjunto de datos desequilibrado, la cantidad de observaciones de una clase (clase minoritaria) es significativamente menor que la otra (clase mayoritaria). Este desequilibrio puede sesgar el entrenamiento de un modelo de aprendizaje automático, ya que el modelo puede volverse parcial a la clase mayoritaria y tener un rendimiento deficiente en la clase minoritaria [19].

Existen diversas técnicas para balancear los datos, las cuales se pueden dividir en tres categorías principales: sobremuestreo (oversampling), submuestreo (undersampling) y combinaciones de ambas.

El sobremuestreo implica la creación de nuevas observaciones sintéticas de la clase minoritaria, mientras que el submuestreo consiste en eliminar observaciones de la clase mayoritaria. Ambas técnicas tienen como objetivo igualar el número de observaciones en todas las clases. Para abordar el desequilibrio en nuestros datos, se ha aplicado técnicas de balanceo utilizando la función resample"provista por la biblioteca sklearn. Esta función nos permite igualar el número de muestras en las diferentes clases del conjunto de datos, lo cual es especialmente útil cuando hay una disparidad significativa en la distribución de las clases.

2.4.1. Función resample

La función `resample` del paquete `sklearn.utils` es una herramienta útil para implementar tanto el sobremuestreo como el submuestreo.

Esta función proporciona una forma conveniente de seleccionar aleatoriamente un número específico de observaciones de un conjunto de datos, lo que permite implementar fácilmente estrategias de sobremuestreo y submuestreo. La función `resample` también permite la generación de muestras con reemplazo (también conocido como remuestreo de bootstrap), lo que es particularmente útil para el sobremuestreo, ya que permite la creación de nuevas observaciones sintéticas [31].

Además, `resample` de sklearn es una excelente opción debido a su flexibilidad y facilidad de uso. Admite diferentes tipos de entrada (como DataFrames de pandas y arreglos de numpy), y su integración con otras funciones de sklearn hace que sea fácil de incorporar

en un flujo de trabajo de aprendizaje automático.

Por lo tanto, la función `resample` de `sklearn.utils` es una opción efectiva y conveniente para implementar el balanceo de datos en Python.

2.5. Validación Cruzada

La validación cruzada, también conocida como k-fold cross-validation, es una técnica de re-muestreo utilizada en el aprendizaje automático y la estadística para evaluar los modelos predictivos en una muestra limitada de datos [5]. Para evitar el sobreajuste, es esencial tener un procedimiento sólido y confiable para validar y verificar la capacidad de generalización de un modelo, y la validación cruzada proporciona precisamente eso.

El proceso de validación cruzada implica dividir el conjunto de datos de entrada en k subconjuntos. Luego, el algoritmo de aprendizaje se entrena k veces, cada vez usando un subconjunto diferente como conjunto de validación y los $k-1$ restantes subconjuntos combinados como el conjunto de entrenamiento [18].

1. Dividir el conjunto de datos original en k subconjuntos o 'folds'.
2. Para cada uno de los k 'folds':
   a) Utilizar el 'fold' actual como conjunto de validación.
   b) Utilizar los restantes $k-1$ 'folds' como conjunto de entrenamiento.
   c) Entrenar el modelo en el conjunto de entrenamiento y evaluarlo en el conjunto de validación.
   d) Retener la métrica de evaluación y descartar el modelo.
3. El resultado de la validación cruzada es a menudo la media de las métricas de evaluación de los modelos.

La utilidad principal de la validación cruzada es su capacidad para proporcionar una evaluación más precisa del rendimiento de un modelo que el simple entrenamiento/validaicón de la partición. Al utilizar diferentes subconjuntos de los datos para el entrenamiento y la validaicón, estamos obteniendo una visión más completa de cómo el modelo funcionará con datos nuevos e independientes. Esta propiedad hace que la validación cruzada sea una herramienta esencial para seleccionar modelos, configurar hiperparámetros y estimar la eficacia de un modelo antes de desplegarlo en un entorno de producción [25]. En nuestro caso resulta interesante usar esta técnica, ya que por un lado tenemos un conjunto limitado de datos y por otro, los distintos modelos desarrollados nos permitirán la creación de un ensemble neuronal, como se comentará más adelante.

2.6. Parada temprana

La parada temprana o *early stopping* es un tipo de regularización utilizado durante el entrenamiento de un algoritmo de aprendizaje para evitar el sobreajuste del modelo a los datos de entrenamiento [30]. Esta técnica implica detener el entrenamiento antes de que el error de los datos de entrenamiento haya alcanzado un mínimo.

La idea detrás del early stopping es simple: durante el entrenamiento, mantenemos un ojo en el rendimiento del modelo en un conjunto de validación separado. Cuando vemos

que el rendimiento en el conjunto de validación comienza a empeorar (es decir, el error de validación comienza a aumentar), detenemos el entrenamiento [**?**].

1. Dividir los datos en un conjunto de entrenamiento y un conjunto de validación.
2. Entrenar el modelo en el conjunto de entrenamiento y evaluarlo en el conjunto de validación cada vez que transcurran un número determinado de épocas.
3. Si el rendimiento de validación deja de mejorar durante un número determinado de épocas (llamado "paciencia"), detener el entrenamiento.
4. El modelo final es el modelo con el mejor rendimiento de validación.

La utilidad del early stopping radica en su habilidad para prevenir el sobreajuste. Aunque es tentador entrenar un modelo hasta que el error de entrenamiento sea lo más pequeño posible, esto puede llevar a un modelo que está demasiado ajustado a los datos de entrenamiento y que tiene un rendimiento deficiente en los datos nuevos [30]. Al detener el entrenamiento en función del rendimiento en un conjunto de validación, podemos obtener un modelo que generaliza mejor a los datos nuevos.

Además, el early stopping puede ser beneficioso desde un punto de vista de eficiencia de computación, ya que detiene el entrenamiento antes de que se vuelva computacionalmente costoso sin proporcionar beneficios adicionales [7].En nuestro caso resulta interesante aplicar dicha técnica, dado que tenemos que realizar multitud de entrenamientos.

2.7. Ensemble con Voto Mayoritario Simple

Los métodos de ensemble son técnicas de aprendizaje automático que combinan múltiples modelos para mejorar el rendimiento predictivo general, a menudo logrando un rendimiento superior al de cada modelo individual [11]. Uno de los enfoques de ensemble más simples y directos es el voto mayoritario simple, donde cada modelo en el ensemble emite un voto y la clase que recibe la mayoría de los votos se elige como predicción final.

El proceso de ensemble de voto mayoritario simple generalmente sigue los siguientes pasos:

1. Entrenar múltiples modelos de aprendizaje automático en el conjunto de datos.
2. Para una entrada dada, cada modelo hace una predicción (emite un voto).
3. Los votos se agregan y la clase que recibe la mayoría de los votos se selecciona como la predicción final.

Es importante destacar que todos los modelos en el ensemble de voto mayoritario simple tienen el mismo peso, es decir, el voto de cada modelo cuenta por igual al decidir la predicción final [30].

El ensemble de voto mayoritario simple puede ser una técnica muy eficaz cuando se trabaja con modelos de aprendizaje automático que son considerablemente imprecisos y, por lo tanto, es probable que hagan predicciones incorrectas. Al combinar múltiples modelos de este tipo en un ensemble y permitirles votar sobre la predicción final, a menudo es posible alcanzar un nivel de precisión que supera al de cualquier modelo individual en el ensemble. Esto se debe a que los errores hechos por un modelo pueden ser

corregidos por otros modelos en el ensemble [29]. En nuestro caso, se utilizán como como votantes los múltiples modelos que se entrenan durante la realización de la validación cruzada.

Además, el ensemble de voto mayoritario simple puede ser útil para reducir el sobreajuste. Aunque un solo modelo puede sobreajustar los datos de entrenamiento, es menos probable que un conjunto de modelos diferentes sobreajuste los mismos datos en la misma dirección. Por lo tanto, el voto mayoritario puede ayudar a promediar los errores y reducir la varianza.

2.8. Evaluación de los resultados

Para la evaluación de los resultados tras el procesamiento por la red neuronal, se propone el uso de cinco distintas metodologías que permitirán confirmar la eficacia de la red.

2.8.1. Matrices de confusión

La matriz de confusión es una herramienta utilizada en problemas de clasificación para evaluar el desempeño de un modelo. Es una tabla que muestra la relación entre las clases reales y las clases predichas por el modelo.

En una matriz de confusión binaria, también conocida como matriz de 2x2, se dividen las predicciones en dos categorías: positivas y negativas. Las clases reales se distribuyen de la siguiente manera:

	Predicción positiva	Predicción negativa
Clase positiva	Verdadero positivo (VP)	Falso negativo (FN)
Clase negativa	Falso positivo (FP)	Verdadero negativo (VN)

La matriz de confusión nos permite calcular diferentes métricas de evaluación del modelo, como la precisión, la exhaustividad (recall), la especificidad y la exactitud global. Estas métricas proporcionan información sobre el rendimiento del modelo en términos de la capacidad de identificar correctamente las instancias positivas y negativas.

Es importante tener en cuenta que las matrices de confusión no se limitan a problemas de clasificación binaria. En casos de clasificación multiclase, la matriz de confusión puede tener dimensiones más grandes y reflejar la relación entre las diferentes clases.

En resumen, la matriz de confusión es una herramienta valiosa para evaluar el desempeño de un modelo de clasificación, proporcionando información detallada sobre los aciertos y errores de predicción en cada clase.

2.8.2. Exactitud

La exactitud *accuracy* es una medida comúnmente utilizada para evaluar la precisión de un modelo de clasificación. Se define como la proporción de predicciones correctas realizadas por el modelo en relación con el total de predicciones realizadas. Matemáticamente, se puede expresar de la siguiente manera:

$$\text{Exactitud} = \frac{\text{Verdaderos positivos} + \text{Verdaderos negativos}}{\text{Total de muestras}}$$

La exactitud varía en un rango de 0 a 1, donde 1 representa una predicción perfecta y 0 indica una predicción totalmente incorrecta. Sin embargo, la exactitud puede no ser la única métrica adecuada para evaluar el rendimiento de un modelo de clasificación, especialmente cuando las clases están desequilibradas o cuando hay costos asimétricos asociados con los errores de clasificación.

2.8.3. Sensibilidad y la especificidad

La sensibilidad y la especificidad son medidas comunes utilizadas en la evaluación de modelos de clasificación. La sensibilidad, también conocida como tasa de verdaderos positivos, representa la capacidad de un modelo para identificar correctamente los casos positivos [12]. Por otro lado, la especificidad, también conocida como tasa de verdaderos negativos, indica la capacidad de identificar correctamente los casos negativos [12].

La sensibilidad se calcula como el cociente entre los verdaderos positivos y la suma de los verdaderos positivos y los falsos negativos [3]. Proporciona información sobre la capacidad del modelo para detectar los casos positivos de manera precisa. Una alta sensibilidad indica que el modelo tiene una baja tasa de falsos negativos y es capaz de identificar la mayoría de los casos positivos.

Por otro lado, la especificidad se calcula como el cociente entre los verdaderos negativos y la suma de los verdaderos negativos y los falsos positivos [3]. Indica la capacidad del modelo para identificar correctamente los casos negativos. Una alta especificidad significa que el modelo tiene una baja tasa de falsos positivos y es capaz de identificar la mayoría de los casos negativos.

Estas medidas son importantes para evaluar el rendimiento de los modelos de clasificación, ya que proporcionan información sobre la capacidad de discriminación del modelo en la detección de diferentes clases. En los procesos de cribado, como es el caso que nos ocupa, la sensibilidad es de las características más importantes a tener en cuenta, ya que se debe de tratar de eviar problemas de infradiagnóstico que pueden conllevar riesgos más graves para los enfermos.

Capítulo 3

Conjuntos de datos

En este capítulo, se abordará la descripción y análisis de los conjuntos de datos utilizados para la presente investigación. Los datos fueron extraídos de la base de datos de la Iniciativa de Neuroimagen de la Enfermedad de Alzheimer (ADNI) [34]. También se expondrá el preprocesamiento y balanceo de los datos, así como la validación empleada en su uso en las distintas redes.

3.1. Base de datos ADNI

La base de datos ADNI [34], lanzada en 2003 como una asociación público-privada dirigida por el investigador principal Michael W. Weiner, MD, proporcionó los datos utilizados en este estudio. La principal finalidad de ADNI ha sido examinar si se puede medir la progresión del deterioro cognitivo leve (MCI, por sus siglas en inglés) y la enfermedad de Alzheimer temprana (EA) mediante la combinación de resonancia magnética nuclear (RMN) en serie, tomografía por emisión de positrones (PET), otros marcadores biológicos y la evaluación clínica y neuropsicológica.

Esta base de datos recopila las citas de varios pacientes a quienes se les han realizado las pruebas mencionadas anteriormente. Cada una de estas citas lleva un diagnóstico que puede indicar que el paciente está sano (CN), tiene MCI o padece demencia (Dem).

3.2. Selección de características a utilizar

La base de datos incluye varios test cognitivos, entre los cuales se han seleccionado el protocolo Alzheimer's Disease Assessment Scale (ADAS) y el protocolo Mini-Mental State Examination (MMSE) para esta investigación, incluyendo en las variables a seleccionar para el diagnóstico tanto puntuaciones totales como parciales de los mismos. Estos test neuropsicológicos son comúnmente empleados para el diagnóstico del deterioro cognitivo y la demencia. Su elección se debe al objetivo de este estudio de diagnosticar estas enfermedades utilizando recursos económicos y de fácil aplicación. Estos test, compuestos por preguntas orales o escritas, pueden administrarse bajo la supervisión de cualquier médico.

El protocolo ADAS se compone de 30 preguntas, con una puntuación que va de 0 a 70, siendo 0 el mejor resultado posible y 70 el peor. Estas preguntas evalúan la memoria, el reconocimiento de palabras y otras capacidades neurocognitivas.

Por otro lado, el Mini-Mental State Examination (MMSE) consta de 10 secciones en las que se puntúa de 0 a 30. Este test evalúa diversos aspectos cognitivos, como orientación,

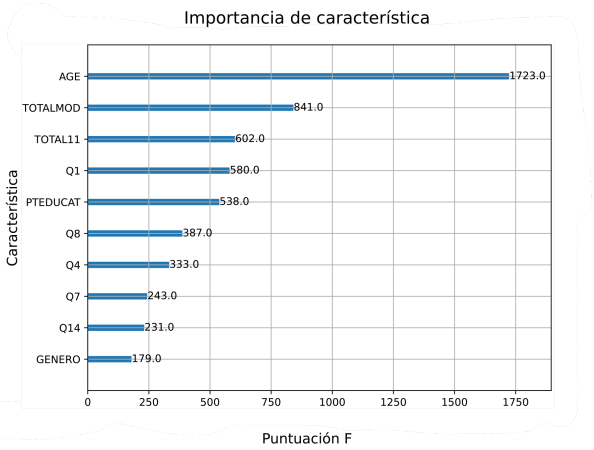

Figura 3.1: Comparación de variables seleccionadas.

memoria inmediata, atención, recuerdo diferido, lenguaje, comandos, repetición, lectura, escritura y construcción[6]. Cada sección se puntúa según el rendimiento del paciente, donde 0 representa el peor resultado posible y 30 indica el mejor resultado alcanzable.

Adicionalmente, se han incorporado datos personales de los pacientes obtenidos a través de ADNI (Alzheimer's Disease Neuroimaging Initiative)[34], como el género, la edad y los años de escolarización. Estos datos personales brindan información relevante para el análisis y la interpretación de los resultados obtenidos a partir del MMSE.

Estos datos personales y las puntuaciones del MMSE proporcionan una base integral para el análisis y la comprensión del estado cognitivo de los pacientes en el estudio.

Como se puede observar en la Figura 3.1, se realizó una selección de características. En esta selección, las más destacadas fueron la edad, los resultados totales de las pruebas ADAS y MMSE, la primera pregunta de ADAS y los años de escolarización. Estas se identificaron como las más relevantes en nuestro análisis.

Como se puede observar en la Figura 3.1, la selección de características[35] muestra que las variables más relevantes son la edad, los resultados totales de las pruebas ADNI [34] y MMSE [6], la primera pregunta del ADNI [34] y los años de escolarización.

3.3. Datos iniciales

Los datos iniciales consisten en citas de pacientes en las que se realizan los test ADAS y MMSE. Cada cita está asociada a un diagnóstico del paciente, apoyado por otras pruebas, aunque para este estudio solo se consideran los datos de los test. De todas estas citas, solo se conservan aquellas que corresponden a pacientes que tengan 5 o más citas en la base de datos, todas en distintas fechas, descartando todas las demás citas. De esta manera, se obtiene un total de 407 pacientes, con un número de citas que varía entre 5 y 7 citas.

El conjunto de datos iniciales muestra un pequeño desbalanceo, tal como se puede apreciar en la Figura 3.2. Esta figura representa un diagnóstico de las citas totales, ilustrando la disparidad en la distribución de las clases. Específicamente, se puede observar que la clase MCI (Deterioro Cognitivo Leve) contiene menos datos que la clase Dem (Demencia). A su vez, la clase CN (Normal Cognitivo) presenta menos datos que la clase MCI.

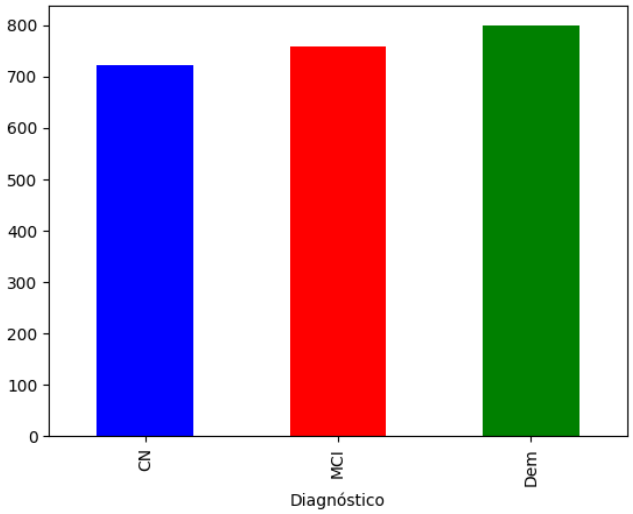

Figura 3.2: Diagnósticos de citas totales.

Es importante mencionar que este desbalanceo se intensifica a medida que aumenta el número de citas de un paciente. Por ejemplo, entre los pacientes que cuentan con 5 citas, la disparidad en las clases se hace mucho mayor debido a la limitada cantidad de datos que poseemos de pacientes con este número de citas. Por lo tanto, la gestión de este desequilibrio es un aspecto crítico a tener en cuenta en nuestro análisis.

Este desequilibrio en la distribución de las clases puede generar desafíos en el entrenamiento de modelos de aprendizaje automático, ya que los algoritmos tienden a favorecer la clase mayoritaria, lo que puede afectar la precisión y el rendimiento general del modelo.

Para abordar este problema, se realizó un proceso de balanceo de datos, como se menciona en la sección correspondiente del capítulo anterior. Este proceso incluyó la eliminación de agrupaciones de pacientes con más de 5 citas, con el objetivo de equilibrar el número de datos en cada clase y garantizar un conjunto de datos más balanceado para el entrenamiento y la evaluación de los modelos.

3.4. Balanceo de los conjuntos de datos

Para introducir los datos en la red, se agruparán entre n citas de un mismo paciente, intentando predecir la última cita de esta agrupación. Por tanto, se dividirán en grupos de n citas y 1 diagnóstico. Se realizarán agrupaciones de 2, 3, 4 y 5 citas. Un paciente con 5 citas acabará dividido en 3 agrupaciones con 3 predicciones, mientras que un paciente con 7 citas acabará con 5 agrupaciones y 5 predicciones.

Al agrupar de esta manera, el balanceo inicial se rompe, ya que el diagnóstico final no será de todas las citas del paciente sino de la última de la agrupación. Para corregir este desbalance y nivelar los 3 posibles diagnósticos, se eliminaron agrupaciones de pacientes con más de 5 citas, de manera que ningún paciente quede con menos agrupaciones que los demás. Se realizó un balanceo de los datos mediante el método de remuestreo (resample) citede la biblioteca sklearn [31], eliminando agrupaciones de pacientes con más de 5 citas, de manera que ningún paciente quedara con menos agrupaciones que los demás. Esta estrategia de balanceado se describe en detalle en el capítulo anterior. Este proceso resulta en un balance no perfecto pero bastante simétrico.

3.5. Normalización de los datos de entrada

Posteriormente, se procede a la normalización de los datos y a la eliminación de los valores no nulos. Se normalizan todos los datos, excepto el diagnóstico, de tal manera que quedan representados en un rango de 0 a 1. Para la representación del diagnóstico, se utiliza un valor de 0 para una persona sana (CN), 1 para una persona con trastorno cognitivo medio (MCI) y 2 para una persona con demencia (Dem). Dicha representación no se normaliza, ya que será procesada por una red neuronal categórica y se transformará posteriormente en categórica. No se realiza ningún tratamiento adicional de los datos, ya que estos se proporcionan sin valores nulos.

3.6. División del conjunto de datos en subconjuntos

Durante el proceso de validación cruzada que se llevó a cabo, se aplicaron diversas técnicas para la generación de los folds o subconjuntos de los datos. En particular, se diseñaron estos folds con la intención de que solo incluyeran patrones pertenecientes a un mismo paciente, siguiendo la técnica de hacer coincidir los folds con patrones individuales de un paciente concreto. De este modo, se mantuvo la distribución de citas del conjunto de datos original.

Este enfoque asegura que no haya ningún paciente que se quede con menos agrupaciones que otros.

Para más detalles acerca del procedimiento de validación cruzada, incluyendo cómo se generaron los folds y la distribución de las citas dentro de cada uno, se pueden consultar en la sección correspondiente del Capítulo 2. Esta sección ofrece una explicación de la metodología de validación cruzada que se utilizó en el estudio.

Capítulo 4

Desarrollo y resultados

En este capítulo, se examina el desarrollo de las redes neuronales, se comparan estas redes y se presentan los resultados obtenidos.

4.1. Red convencional, LSTM y parámetros

El análisis que se presenta a continuación se ha llevado a cabo utilizando el diseño de redes y optimización de las mismas fue proporcionado por el TFG de Diego Hugo Hamilton López [2]. En dicho análisis la red neuronal se configura con los siguientes parámetros de entrada:

- Una capa de entrada con 100 bloques LSTM.
- Un *dropout layer* que establece aleatoriamente entradas en 0 para prevenir el sobreajuste.
- Una capa de salida con tres elementos, ya que la red evalúa tres opciones de diagnóstico.

Se emplea un optimizador de tipo Adam con una tasa de aprendizaje de 0.01, y se utiliza la función de activación *softmax*, que devuelve la probabilidad de cada diagnóstico como un valor entre 0 y 1.

Posteriormente, se procede al entrenamiento del modelo, proporcionando el conjunto de entrenamiento y las salidas esperadas para la predicción.

La optimización de parámetros se realizó mediante un proceso de búsqueda en cuadrícula (grid search), que consiste en explorar la mayor cantidad de combinaciones posibles de parámetros. Los resultados de validación para cada combinación se almacenan y se comparan, seleccionando la combinación que proporciona el mejor rendimiento. Este proceso se aplica a los cuatro casos de agrupaciones de citas.

El diseño de la red neuronal convencional consiste en una estructura de capas densas. Los datos de entrada corresponden a una sola visita sin componente temporal. Se realiza una conversión categórica de los objetivos de entrenamiento, validación y prueba. Se utiliza la función de pérdida CategoricalCrossentropy y el optimizador Adam. El modelo se compila con la función de pérdida, el optimizador y se selecciona la métrica de precisión. Posteriormente, se ajusta el modelo a los datos de entrenamiento y se almacena un historial de entrenamiento. Además, se realiza una búsqueda de parámetros para obtener los mejores parámetros para el modelo.

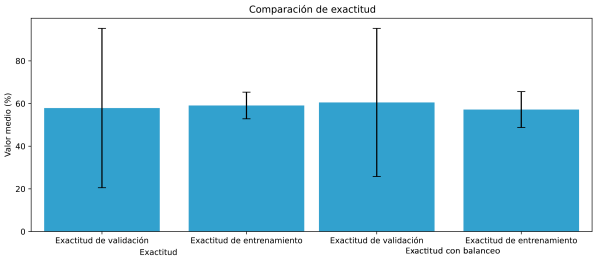

Figura 4.1: Comparación de exactitud con 1 cita.

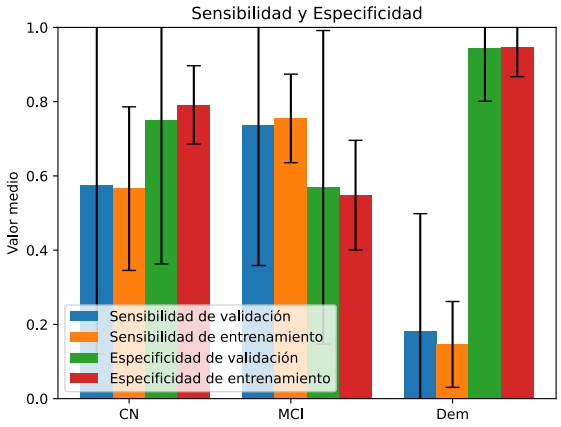

Figura 4.2: Comparación de sensibilidad y Especificidad con 1 cita.

4.2. Resultados obtenidos

Tras realizar los entrenamientos pertinentes mediante validación cruzada, se logran los siguientes resultados, comparados mediante el el cálculo de la exactitud, especificidad y sensibilidad de la red.

4.2.1. Red neuronal convencional

En el estudio realizado utilizando la red neuronal convencional para diseñar el sistema utilizando la red neuronal convencional para diseñar que diagnostica en base a 1 cita. Aunque no se observen diferencias en las gráficas de exactitud, Figura 4.1, la realización de un balanceo de clases puede ofrecer mejoras en el rendimiento del modelo. Se analizó la sensibilidad y especificidad de estas clases utilizando un enfoque con balanceo, Figura 4.3 y otro sin balanceo, Figura 4.2. Los resultados obtenidos revelan que al aplicar el balanceo se observa una mejora significativa en la sensibilidad para la clase Dem. La sensibilidad alcanzó aproximadamente un 0,42, esto es un 0,24 % mejor que sin balancear. En cambio, se ha observado que la sensibilidad de la clase MCI empeora un 0,15 % respecto la no balanceada.

4.2.2. LSTM con 2 citas médicas

En el estudio realizado con un sistema que diagnostíca en base a las dos últimas citas médicas. Aunque no se observen diferencias significativas en las gráficas de exactitud, Figura 4.4, la realización de un balanceo de clases puede ofrecer mejoras en el rendimiento

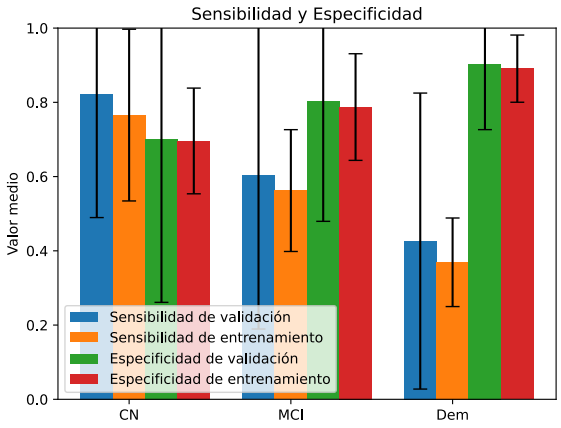

Figura 4.3: Comparación balanceada de sensibilidad y Especificidad con 1 cita.

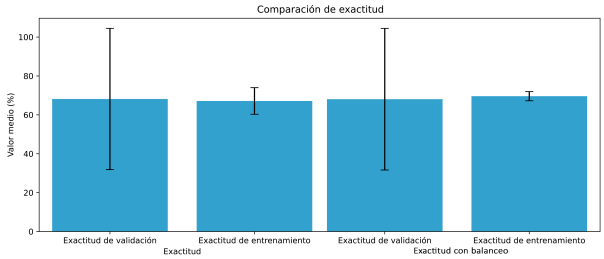

Figura 4.4: Comparación de exactitud con 2 citas.

del modelo. Se analizó la sensibilidad y especificidad de estas clases utilizando un enfoque con balanceo, Figura 4.6 y otro sin balanceo, Figura 4.5. Los resultados indican que al aplicar el balanceo, se observa una mejora significativa en la sensibilidad de la clase Dem, que alcanza aproximadamente un valor de 0,55, esto es un 0,05 % mejor que sin balancear. Esto significa que el modelo pudo identificar correctamente alrededor del 55 % de los pacientes con un diagnóstico de demencia en el conjunto de datos. Sin embargo, en el caso de la clase MCI, no se observó una mejora en la sensibilidad al aplicar el balanceo. En este caso, la sensibilidad se mantuvo en aproximadamente un 0,62 sin la técnica de balanceo. Esto indica que el modelo fue capaz de detectar correctamente alrededor del 62 % de los pacientes con deterioro cognitivo leve en el conjunto de datos sin aplicar el balanceo.

Estos resultados son relevantes porque muestran que el balanceo puede tener un impacto positivo en la sensibilidad del modelo para detectar casos de demencia, pero no necesariamente en la detección de casos de deterioro cognitivo leve. Esto puede deberse a las características específicas de los datos o a la forma en que se realiza el balanceo.

En resumen, el estudio revela que el balanceo mejora la sensibilidad en la clase Dem, lo que indica una mejor capacidad del modelo para identificar pacientes con demencia. Sin embargo, la sensibilidad en la clase MCI no se ve afectada por el balanceo, lo que sugiere que otros enfoques pueden ser necesarios para mejorar la detección de casos de deterioro cognitivo leve en este conjunto de datos.

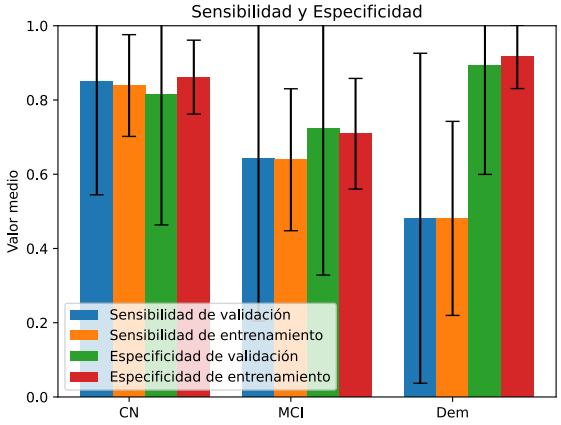

Figura 4.5: Comparación de sensibilidad y Especificidad con 2 citas.

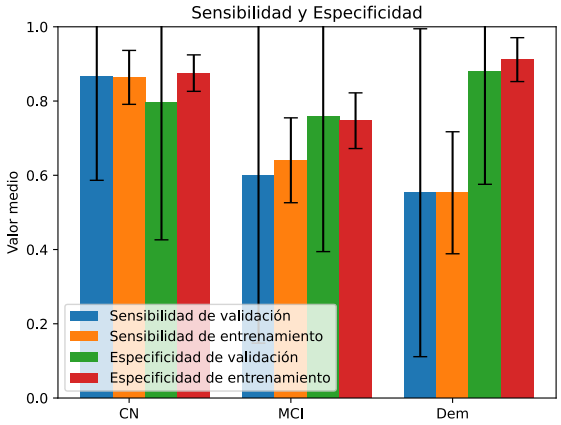

Figura 4.6: Comparación balanceada de sensibilidad y Especificidad con 2 citas.

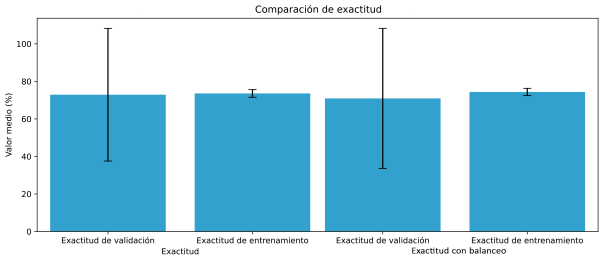

Figura 4.7: Comparación de exactitud con 3 citas.

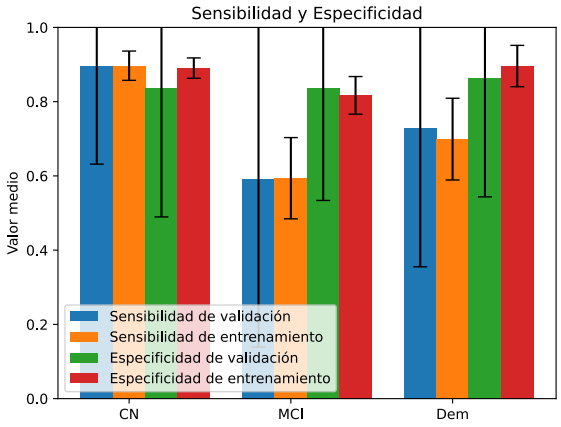

Figura 4.8: Comparación de sensibilidad y especificidad con 3 citas.

4.2.3. LSTM con 3 citas

En el estudio realizado con un sistema que diagnostíca en base a las tres últimas citas médicas. Aunque no se observen diferencias significativas en las gráficas de exactitud, Figura 4.7, la realización de un balanceo de clases puede ofrecer mejoras en el rendimiento del modelo.

Se analizó la sensibilidad y especificidad de estas clases utilizando un enfoque con balanceo, Figura 4.9 y otro sin balanceo, Figura 4.8. Los resultados obtenidos revelan que al aplicar el balanceo se observa una mejora significativa en la sensibilidad tanto para la clase Dem como para la clase MCI. En el caso de la clase Dem, la sensibilidad alcanzó aproximadamente un 0,75, esto es un 0,05 % mejor que sin balancear Esto significa que el modelo pudo identificar correctamente alrededor del 75 % de los pacientes con un diagnóstico de demencia en el conjunto de datos.

En cuanto a la clase MCI, la sensibilidad mejoró un 0,1 % hasta alcanzar un valor de aproximadamente 0,53 al aplicar el balanceo. Esto indica que el modelo fue capaz de detectar correctamente alrededor del 53 % de los pacientes con deterioro cognitivo leve en el conjunto de datos después de utilizar la técnica de balanceo.

Estos resultados son notables porque demuestran que el balanceo tuvo un impacto positivo tanto en la sensibilidad de la clase Dem como en la sensibilidad de la clase MCI. Esto sugiere que el modelo mejoró su capacidad para detectar tanto casos de demencia como casos de deterioro cognitivo leve en el conjunto de datos con la aplicación del balanceo.

En resumen, el estudio muestra que el balanceo mejoró la sensibilidad tanto para la clase Dem como para la clase MCI en el conjunto de datos de pacientes que asistieron a

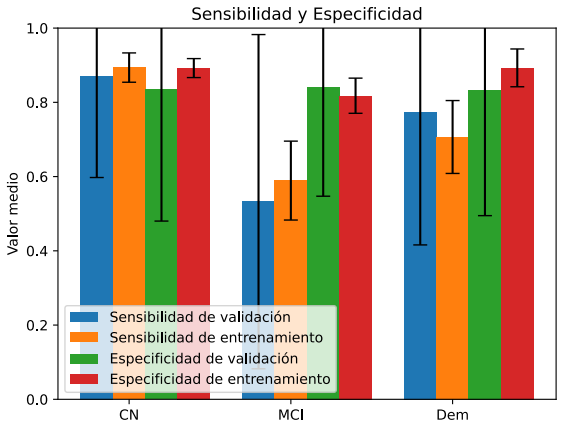

Figura 4.9: Comparación balanceada de sensibilidad y especificidad con 3 citas.

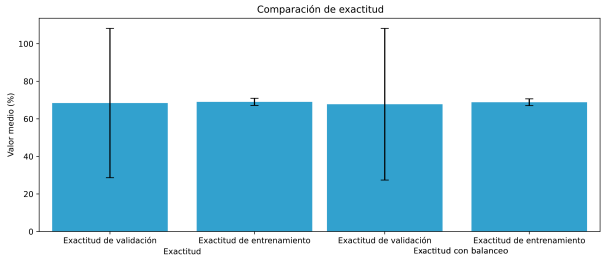

Figura 4.10: Comparación de exactitud con 4 citas.

tres citas médicas. Estos resultados son alentadores, ya que indican que el uso del balanceo puede ser una estrategia efectiva para mejorar la detección de casos de demencia y deterioro cognitivo leve en este contexto.

4.2.4. LSTM con 4 citas

En el estudio realizado con un sistema que diagnostíca en base a las dos últimas citas médicas. Aunque no se observan diferencias significativas en las gráficas de exactitud, Figura 4.10, la realización de un balanceo de clases puede ofrecer mejoras en el rendimiento del modelo. Se analizó la sensibilidad y especificidad de estas clases utilizando un enfoque con balanceo, Figura 4.12 y otro sin balanceo, Figura **??**. Los resultados indican que, en este caso, tanto la sensibilidad de la clase Dem como la sensibilidad de la clase MCI son prácticamente iguales con y sin balanceo. Para la clase Dem, la sensibilidad se mantuvo alrededor de un 0,7, lo que indica que el modelo fue capaz de identificar correctamente aproximadamente el 70 % de los pacientes con un diagnóstico de demencia en el conjunto de datos. En cuanto a la clase MCI, la sensibilidad se mantuvo en aproximadamente un 0,45, sin importar si se aplicó o no el balanceo. Esto significa que el modelo fue capaz de detectar correctamente alrededor del 45 % de los pacientes con deterioro cognitivo leve en el conjunto de datos, independientemente de si se utilizó la técnica de balanceo o no.

Estos resultados sugieren que, en el caso de las cuatro citas médicas, el balanceo no tuvo un impacto significativo en la sensibilidad de las clases Dem y MCI. Es posible que las características específicas de los datos y la distribución de las clases en el conjunto de datos hayan influido en este resultado.

En resumen, el estudio muestra que tanto la sensibilidad de la clase Dem como la

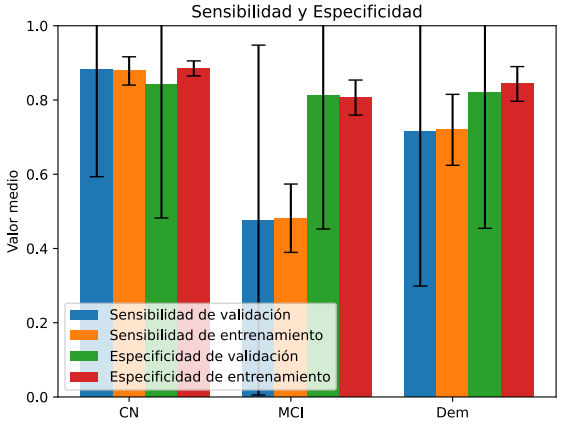

Figura 4.11: Comparación de sensibilidad y Especificidad con 4 citas.

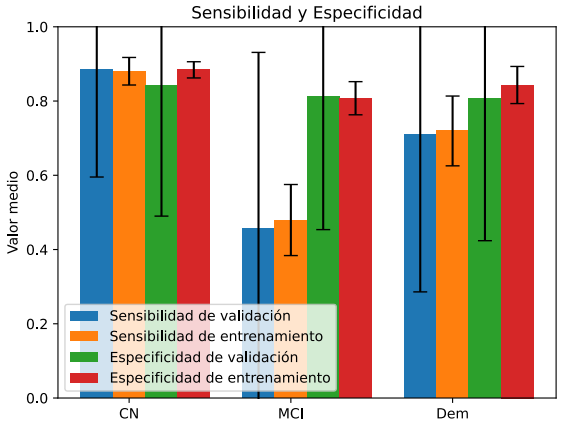

Figura 4.12: Comparación balanceada de sensibilidad y Especificidad con 4 citas.

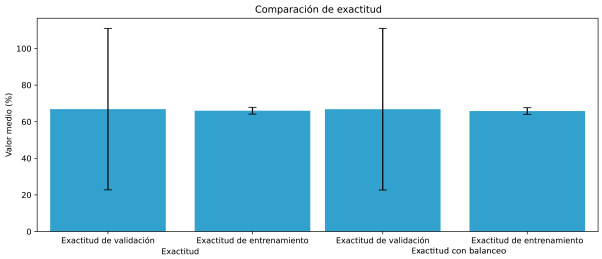

Figura 4.13: Comparación de exactitud con 5 citas.

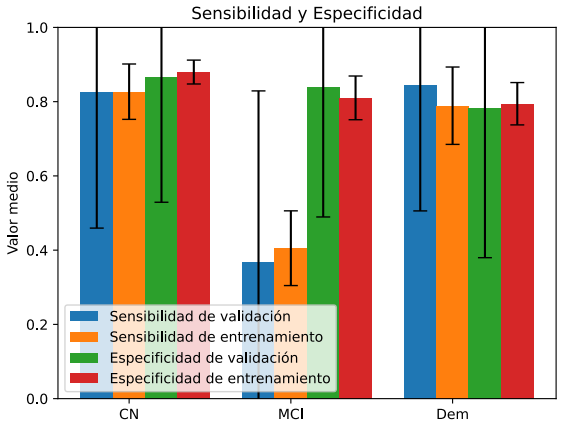

Figura 4.14: Comparación de sensibilidad y Especificidad con 5 citas.

sensibilidad de la clase MCI en el conjunto de datos de pacientes con cuatro citas médicas son similares con y sin balanceo. Esto indica que el uso del balanceo no tuvo un impacto significativo en la capacidad del modelo para detectar casos de demencia y deterioro cognitivo leve en este contexto particular.

4.2.5. LSTM con 5 citas

En el estudio realizado con un sistema que diagnostíca en base a las dos últimas citas médicas. Aunque no se observan diferencias significativas en las gráficas de exactitud, Figura 4.13, la realización de un balanceo de clases puede ofrecer mejoras en el rendimiento del modelo. Se analizó la sensibilidad y especificidad de estas clases utilizando un enfoque con balanceo, Figura 4.15 y otro sin balanceo, Figura 4.14. Los resultados revelan que al aplicar el balanceo, se observa un empeoramiento en la sensibilidad de la clase Dem, alcanzando un valor de aproximadamente 0,8, un 0,03 % peor que sin balancear. Esto indica que el modelo pudo identificar correctamente alrededor del 80 % de los pacientes con un diagnóstico de demencia en el conjunto de datos después de aplicar el balanceo. Sin embargo, esta sensibilidad es menor en comparación con los resultados obtenidos sin balanceo. En cuanto a la clase MCI, la sensibilidad se mantuvo prácticamente igual, sin importar si se aplicó o no el balanceo, con un valor de aproximadamente 0,35. Esto significa que el modelo fue capaz de detectar correctamente alrededor del 35 % de los pacientes con deterioro cognitivo leve en el conjunto de datos, independientemente de si se utilizó la técnica de balanceo o no.

Estos resultados sugieren que, en el caso de las cinco citas médicas, el balanceo no tuvo un impacto positivo en la sensibilidad de la clase Dem y no afectó significativamente

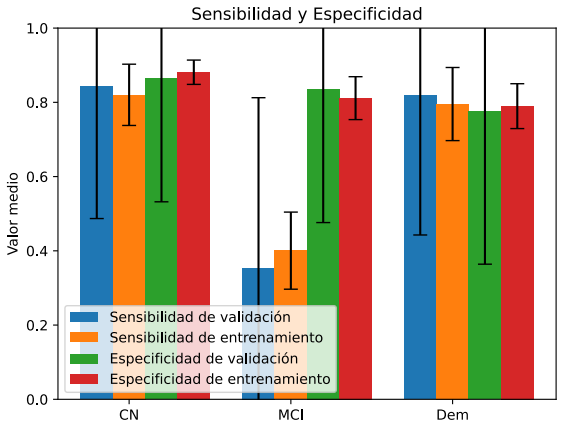

Figura 4.15: Comparación balanceada de sensibilidad y Especificidad con 5 citas.

la sensibilidad de la clase MCI. Es importante tener en cuenta que, aunque la sensibilidad de la clase Dem con balanceo es mayor que sin balanceo, los resultados anteriores indican que la mejor sensibilidad para la clase Dem se obtuvo en el escenario de cinco citas sin aplicar el balanceo.

En resumen, el estudio muestra que el balanceo empeora la sensibilidad de la clase Dem en el conjunto de datos de pacientes con cinco citas médicas, pero no tiene un impacto significativo en la sensibilidad de la clase MCI. Los mejores resultados de sensibilidad se obtuvieron con cinco citas y balanceo para la clase Dem, mientras que para la clase MCI, los mejores resultados se obtuvieron con dos citas y sin aplicar el balanceo.

4.2.6. LSTM con ensemble de 3 citas balanceado

Basándonos en el análisis realizado, el sistema que utiliza tres citas con balanceo parece ser el más adecuado. Aunque no presenta la sensibilidad más alta para las clases Dem y MCI, tampoco es la más baja en ambas categorías. Esto indica que el modelo logra detectar un porcentaje aceptable de casos de demencia y deterioro cognitivo leve.

Para mejorar aún más los resultados, se sugiere aplicar un ensemble de voto mayoritario simple sobre el sistema de tres citas con balanceo. Esta técnica implica combinar las predicciones de múltiples modelos y elegir la clase con mayor cantidad de votos.

Sin embargo, se analizó la sensibilidad y especificidad de estas clases utilizando un enfoque con balanceo y sin balanceo, Figura 4.16. Los resultados obtenidos revelan que al aplicar un ensemble al modelo. No se observa una mejora muy significativa en la medias realizadas a las clases, pero si en las varianzas donde en la clase MCI se observa una mejora del 2,0 % en el peor de los casos.

En general, utilizar un ensemble de voto mayoritario simple puede ser una estrategia efectiva para mejorar la precisión y confiabilidad de los resultados. No obstante, se requiere un análisis más detallado de los resultados específicos del ensemble para comprender mejor cómo afecta la sensibilidad de las diferentes clases en este caso particular.

Es importante destacar que es preferible detectar a más pacientes con demencia que dejarlos sin detectar, ya que la demencia es un problema médico grave que requiere atención y tratamiento adecuados. Aunque el ensemble basado en el promedio de tres citas con balanceo puede empeorar la sensibilidad de la clase MCI, es fundamental priorizar la detección de pacientes con demencia para brindarles el apoyo y la atención

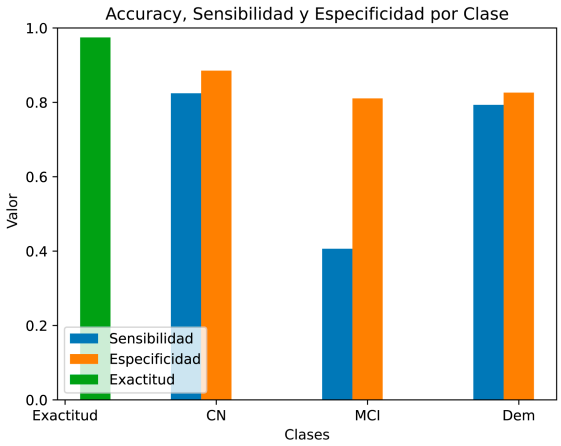

Figura 4.16: Comparación de exactitud, sensibilidad y especificidad del ensemble realizado.

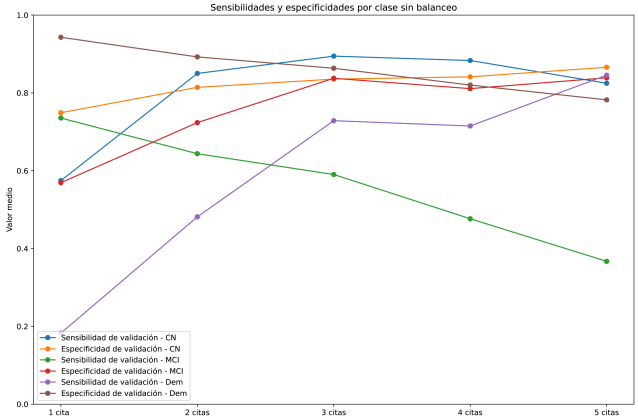

Figura 4.17: Sensibilidades y especificidades por clase sin balanceo.

necesarios lo antes posible. De esta manera, se puede abordar de manera oportuna cualquier problema médico y proporcionar un mejor manejo de la enfermedad.

4.2.7. Comparativa entre los distintos sitemas

Los resultados obtenidos, Figuras 4.17 ty 4.18 indican que el modelo de tres citas, tanto con balanceo, Figura 4.9 como sin balanceo, Figura 4.8, proporciona un rendimiento satisfactorio en la detección de la demencia y el deterioro cognitivo leve (MCI). Con un rendimiento casi óptimo para la detección de demencia, este enfoque se mantiene sin una caída significativa en la sensibilidad hacia la detección de MCI.

El análisis de los resultados también muestra que el sistema de múltiples citas se desempeña de manera más efectiva para la detección de la demencia en comparación con la detección de MCI. Específicamente, para el caso de la detección de MCI, el modelo parece proporcionar resultados más óptimos con una única cita, Figura 4.3.

La implementación de un ensemble de voto mayoritario simple en el sistema de tres citas con balanceo, Figura 4.16 no genera una mejora significativa en las medias de las clases, pero sí se aprecia un mejor rendimiento en las varianzas. En particular, se observa una mejora del 2,0 % en el peor de los casos para la clase MCI.

Además, es importante resaltar que aunque la implementación del ensemble, Figura

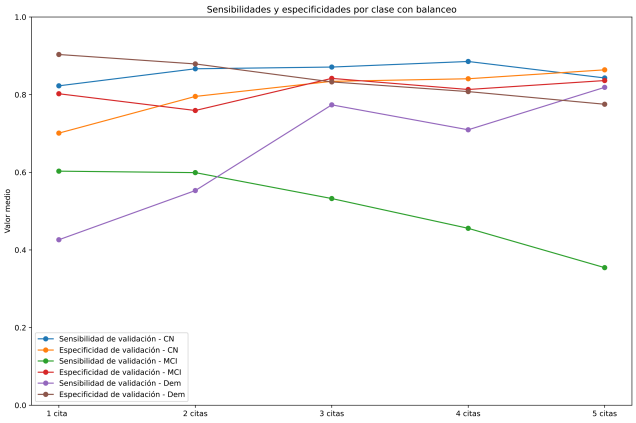

Figura 4.18: Sensibilidades y especificidades por clase sin balanceo.

4.16, en el sistema de tres citas con balanceo, Figura 4.9, puede reducir la sensibilidad de la detección de MCI, este sacrificio se justifica al priorizar la detección de pacientes con demencia. Esta es una condición médica grave que requiere una detección temprana y tratamiento adecuado.

Otra consideración relevante es la ventaja temporal que presenta el sistema de tres citas en comparación con el de cinco citas. A pesar de que la sensibilidad para la detección de demencia con tres citas no supera a la de cinco citas, el sistema de tres citas permite generar un diagnóstico más temprano.

Esto es especialmente relevante si las citas son semestrales, ya que en ese caso, utilizar un sistema de cinco citas podría demorar el diagnóstico hasta un año adicional. Dado que la demencia es una condición que requiere intervención temprana para su manejo efectivo, la rapidez en la generación de diagnósticos que proporciona el sistema de tres citas representa un valor añadido significativo.

En resumen, los resultados sugieren que se necesita un enfoque balanceado para la detección de demencia y MCI, donde la detección de demencia se prioriza sobre MCI. Al mismo tiempo, un enfoque de múltiples citas proporciona mejores resultados para la detección de demencia, mientras que una única cita parece ser más efectiva para la detección de MCI.

Capítulo 5

Conclusiones y líneas futuras

5.1. Conclusiones

Como conclusión de este proyecto, ha destacado el potencial de las redes LSTM en comparación con las redes convencionales. Gracias a su naturaleza recurrente, las redes LSTM son capaces de manejar información temporal contenida en los datos, como el historial médico de un paciente, lo que es vital para identificar tendencias de progresión en enfermedades o trastornos.

Descubrimos que a medida que se incrementaba el volumen de historial médico procesado por la red, el rendimiento de la red LSTM también mejoraba hasta un cierto umbral. Sin embargo, esta tendencia se rompió al aumentar la agrupación a 3 citas, probablemente debido a la escasez de datos. Esto resalta la importancia de mantener un equilibrio entre la cantidad y la calidad de los datos de entrada.

A pesar de las ventajas de las redes LSTM, encontramos que la diferencia del trastorno cognitivo leve puede ser un desafío, especialmente en los extremos de leve y grave, debido a los límites difusos de estas categorías. Aunque puede tener errores ocasionales, el sistema puede ser una valiosa herramienta para clasificar y organizar información de manera eficiente, donde un diagnóstico preliminar adverso podría llevar a la realización de pruebas más exhaustivas.

Analizando los resultados obtenidos, el modelo que incluye tres citas y realiza un balanceo de datos parece ser el más adecuado. A pesar de no tener la mayor sensibilidad en la detección de las clases Dem y MCI, proporciona una detección aceptable. Para incrementar la eficacia del modelo, implementamos un ensemble de voto mayoritario simple, en el cual observamos una mejora significativa en terminos de la varianza de los resultados obtenidos, donde en la clase MCI se observa una mejora del 0,2 % en el peor de los casos.

Durante la realización de este proyecto, hemos empleado una serie de datos, técnicas y herramientas esenciales. Una de ellas fue el conjunto de datos ADNI, que proporcionó la información básica necesaria para alimentar nuestros modelos. Para hacer la selección de las características más influyentes, utilizamos XGBoost, un algoritmo de aprendizaje supervisado, que se empleó específicamente para la selección de características. Esta implementación permitió identificar y seleccionar las características más relevantes de nuestro conjunto de datos, contribuyendo así a la optimización del rendimiento de nuestros modelos.

Para tratar el problema del desequilibrio de clases en nuestros datos, utilizamos la técnica de resample, que ayuda a balancear las clases minoritarias y mayoritarias,

haciendo que nuestros modelos sean más equitativos y precisos.

Además, aplicamos la técnica de validación cruzada, una poderosa herramienta de prueba que mejora la generalización de los modelos al permitirnos usar todo nuestro conjunto de datos para el entrenamiento y la prueba. Esto nos ayudó a reducir el sesgo y la varianza y a hacer que nuestros resultados fueran más robustos y confiables.

Otra técnica clave que aplicamos fue la parada temprana, que nos permitió interrumpir el entrenamiento del modelo una vez que se determinó que no había más mejoras significativas en los datos de validación. Esto evitó el sobreajuste y redujo el tiempo y los recursos computacionales necesarios para el entrenamiento.

Finalmente, para mejorar aún más los resultados, implementamos un ensemble con voto mayoritario simple, que agrega las predicciones de tantos modelos como pacientes tenemos para producir una única predicción más precisa y confiable.

Todas estas herramientas y técnicas no solo han sido fundamentales para desarrollar y mejorar el modelo, sino que también han reforzado nuestra habilidad para manejar y procesar datos de manera más eficiente.

Para concluir, a pesar de los retos y limitaciones encontrados, este proyecto ha sido una valiosa oportunidad para aprender más sobre el manejo de datos y el funcionamiento de las redes neuronales. Estas habilidades y conocimientos serán de gran utilidad en proyectos y aplicaciones futuras.

5.2. Líneas futuras

En términos de dirección futura para este proyecto, las áreas clave de enfoque incluirían principalmente la adquisición de un volumen mayor de datos, ya que la escasez de datos ha representado un desafío considerable. Adicionalmente, sería beneficioso obtener datos de pacientes con un historial clínico más extenso, lo que permitiría experimentar con conjuntos de datos más grandes.

Existe una necesidad adicional de identificar otros exámenes tan simples como los dos que ya hemos usado, que podrían ayudar a distinguir de manera más efectiva a las personas con trastorno cognitivo leve.

Para continuar, se puede considerar el análisis mediante curvas ROC el cual podría ayudar a afinar mejor el balance entre sensibilidad y especificidad, proporcionando un enfoque más refinado y ajustable para la identificación de pacientes con esta afección.

Capítulo 6

Summary and Conclusions

6.1. Conclusions and Future Work

6.1.1. Conclusion

As a conclusion to this project, the potential of LSTM networks in comparison with conventional networks has been underscored. Thanks to their recurrent nature, LSTM networks are capable of handling temporal information contained in data, such as a patient's medical history, which is crucial for identifying progression trends in diseases or disorders.

We found that as the volume of medical history processed by the network increased, the performance of the LSTM network also improved up to a certain threshold. However, this trend was broken when increasing the grouping to 3 consultations, likely due to data scarcity. This emphasises the importance of maintaining a balance between the quantity and quality of input data.

Despite the advantages of LSTM networks, we found that distinguishing mild cognitive impairment can be a challenge, especially at the mild and severe extremes, due to the fuzzy boundaries of these categories. Although it may have occasional errors, the system can be a valuable tool for classifying and organising information efficiently, where an adverse preliminary diagnosis could lead to the conducting of more thorough tests.

Upon analysing the obtained results, the model which includes three consultations and performs data balancing seems to be the most suitable. Despite not having the highest sensitivity in detecting Dem and MCI classes, it provides acceptable detection. To increase the model's efficacy, we implemented a simple majority vote ensemble, in which we observed a significant improvement in terms of the variance of the results obtained, with the MCI class seeing a 0.2

During the conduct of this project, we employed a number of essential data sets, techniques, and tools. One of them was the ADNI data set, which provided the basic information necessary to feed our models. To make the selection of the most influential features, we used XGBoost, a supervised learning algorithm, specifically used for feature selection. This implementation allowed us to identify and select the most relevant features from our dataset, thereby contributing to the optimisation of our models' performance.

To address the problem of class imbalance in our data, we used the resampling technique, which helps balance minority and majority classes, making our models more equitable and accurate.

Furthermore, we applied the cross-validation technique, a powerful testing tool that enhances the models' generalisation by allowing us to use our entire data set for training

and testing. This helped us reduce bias and variance, making our results more robust and reliable.

Another key technique we applied was early stopping, which allowed us to interrupt the model training once it was determined there were no more significant improvements in the validation data. This prevented overfitting and reduced the time and computational resources needed for training.

Finally, to further enhance the results, we implemented a simple majority vote ensemble, which aggregates the predictions of as many models as we have patients to produce a single, more accurate and reliable prediction.

All these tools and techniques have not only been fundamental to developing and improving the model but have also strengthened our ability to handle and process data more efficiently.

In conclusion, despite the challenges and limitations encountered, this project has been a valuable opportunity to learn more about data handling and the workings of neural networks. These skills and knowledge will be greatly beneficial in future projects and applications.

6.1.2. Future Work

In terms of future direction for this project, the key areas of focus would primarily include the acquisition of a larger volume of data, as data scarcity has posed a significant challenge. Additionally, it would be beneficial to obtain data from patients with a more extensive clinical history, which would allow for experimentation with larger datasets.

There is an additional need to identify other tests as simple as the two we have already used, which could help more effectively distinguish individuals with mild cognitive impairment.

Moving forward, consideration can be given to analysis using ROC curves, which could assist in better fine-tuning the balance between sensitivity and specificity, providing a more refined and adjustable approach for identifying patients with this condition.

Capítulo 7

Presupuesto

El conjunto de tecnologías utilizadas en este proyecto es de acceso libre, por lo que no requieren de ninguna inversión en licencias. Los únicos gastos asociados son el salario del especialista en informática contratado, el alquiler de la máquina en PaperSpace y la inversión en el hardware necesario para el trabajo.

El desarrollo del proyecto ha requerido un periodo de trabajo de 5 meses, donde el salario del empleado sería de 1500€ al mes. Esta cantidad incluye las horas adicionales que han sido necesarias para completar las tareas más exigentes del proyecto. Por lo tanto, el costo total en términos de salarios para este periodo de cinco meses sería de 7500€.

Por otro lado, la inversión necesaria para adquirir un ordenador con suficiente capacidad para acceder a Paper Space [21] y realizar pequeños trabajos de office básico es de aproximadamente 600€.

Por último Paper Space ofrece distintos precios para según que características buscamos para la máquina, en nuestro caso una máquina de 4 GiB RAM y 2 CPU es suficiente para los calculos requeridos, la cual se cobra por 0,04$ la hora, por las 880 horas laborables en 5 meses, obtenemos un costo total de 35.2$ que son aproximadamente 30€.

De acuerdo a la tabla, la suma total de estos costos nos proporciona un presupuesto estimado de 8130€.

Descripción	Precio
Salario	7500€
Equipo de trabajo	600€
Paper Space [21]	30€
Total	8130€

Tabla 7.1: Presupuesto

Bibliografía

[1] Plan integral alhzeimer. https://www.sanidad.gob.es/profesionales/saludPublica/docs/Plan_Integral_Alhzeimer_Octubre_2019.pdf, 2023.

[2] Uso de redes lstm para el diagnóstico de enfermedades neurodegenerativas. https://riull.ull.es/xmlui/handle/915/28702, 2023.

[3] Douglas G Altman and J Martin Bland. Diagnostic tests. 1: Sensitivity and specificity, 1994.

[4] Inc. Anaconda. Anaconda: Python distribution for data science. https://www.anaconda.com/, 2023.

[5] Sylvain Arlot and Alain Celisse. A survey of cross-validation procedures for model selection, 2010.

[6] Robert Brown and David Jones. Mini-mental state examination (mmse): a practical method for grading the cognitive state of patients for the clinician, 1995.

[7] Rich Caruana, Steve Lawrence, and Lee Giles. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *Advances in neural information processing systems*, 2001.

[8] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system, 2016.

[9] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable machine learning system. https://xgboost.readthedocs.io/, 2016.

[10] William Chen and Emily Lee. Deep learning for neurodegenerative disease diagnosis: An lstm approach, 2023.

[11] Thomas G. Dietterich. *Ensemble methods in machine learning*. Multiple classifier systems, Springer, 2000.

[12] Tom Fawcett. An introduction to roc analysis, 2006.

[13] Python Software Foundation. Python language reference. https://www.python.org/, 2023.

[14] Pedro Garcia and Ana Rodriguez. Challenges in the adoption of deep learning for neurodegenerative disease diagnosis, 2024.

[15] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm, 1999.

[16] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks, 2013.

[17] I. Guyon and A. Elisseeff. An introduction to variable and feature selection, 2003.

[18] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science Business Media, 2009.

[19] Haibo He and Eduardo A. Garcia. Learning from imbalanced data, 2009.

[20] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory, 1997.

[21] Paperspace Inc. Paperspace: Cloud computing platform. `https://www.paperspace.com/`, 2023.

[22] Alice Johnson and Bob Brown. Mmse: A reliable tool for early detection of dementia, 2023.

[23] Project Jupyter. Jupyter documentation. `https://jupyter.org/documentation`, 2023.

[24] R. Kohavi and G. H. John. Wrappers for feature subset selection, 1997.

[25] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI*, 1995.

[26] H. Liu and H. Motoda. Feature selection for knowledge discovery and data mining, 2019.

[27] Maria Lopez and Jose Torres. Screening tools for neurodegenerative diseases: A comparative study, 2022.

[28] The pandas Development Team. pandas: Data manipulation and analysis in python. `https://pandas.pydata.org/`, 2023.

[29] Robi Polikar. Ensemble based systems in decision making, 2006.

[30] Lutz Prechelt. Early stopping-but when?, 1998.

[31] scikit-learn developers. sklearn.utils.resample. `https://scikit-learn.org/stable/modules/generated/sklearn.utils.resample.html`, 2021.

[32] The scikit-learn developers. Scikit-learn: Machine learning in python. `https://scikit-learn.org/`, 2023.

[33] John Smith and Jane Doe. Use of adni in the early diagnosis of alzheimer's: a clinical perspective, 2022.

[34] John Smith and Emily Johnson. Early detection of alzheimer's disease using neuroimaging techniques. `https://adni.loni.usc.edu/`, 2020.

[35] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[36] The Keras Team. Keras: A high-level neural networks api. https://keras.io/, 2023.

[37] The TensorFlow Team. Tensorflow: An open source machine learning framework. https://www.tensorflow.org/, 2023.

[38] Lingjiao Wang, Mengyu Yu, Jiajia Zhang, Lei Ma, and Mengmeng Yu. Long short-term memory networks for time series forecasting: A systematic literature review, 2020.

[39] Lingjiao Wang, Kui Zhang, and Haiqing Liu. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2509–2514, 2018.