eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

INFORMATIKA FAKULTATEA
FACULTAD DE INFORMÁTICA

Trabajo de Fin de Grado

Grado en Ingeniería Informática

Computación

Detección de la enfermedad de Alzheimer en base a volúmenes de regiones cerebrales

Jon Vadillo Del Ser

Dirección

Javier Muguerza Rivero

Ibai Gurrutxaga Goikoetxea

23 de junio de 2023

Resumen

El Alzheimer es a día de hoy una de las principales causas de muerte entre la población de avanzada edad. Además, conlleva una pérdida en la calidad de vida de las personas que la padecen y sus allegados.

El diagnóstico de la enfermedad requiere numerosas pruebas, incluyendo alguna dolorosa e invasiva como es la punción lumbar y, teniendo en cuenta que no existe cura alguna para la enfermedad, un gran número de los afectados no se realizan las pruebas.

Con el objetivo de reducir el proceso necesario para la detección de la enfermedad, este proyecto intenta desarrollar una herramienta que a través del aprendizaje automático y utilizando la información volumétrica extraída de resonancias magnéticas, permita un diagnóstico fiable del Alzheimer. Para llevar a cabo el proyecto, se ha contado con la colaboración de la fundación CITA Alzheimer, la cual ha aportado los datos y el conocimiento médico sobre la enfermedad.

Tras llevar a cabo el proyecto y analizar los resultados se ha llegado a la conclusión de que, debido a la dificultad del problema y a la limitada cantidad de datos de los que se dispone, no ha sido posible lograr un diagnóstico suficientemente preciso de la enfermedad. Sin embargo, este trabajo puede servir para el desarrollo de futuros proyectos que permitan una mejora en el diagnóstico de la enfermedad.

Por otro lado, y como objetivo alternativo, se ha propuesto un sistema capaz de minimizar el impacto que supone la punción lumbar, a través de un sistema predictivo que limita la necesidad de la punción lumbar a los casos más probables.

Índice de contenidos

Índice de figuras

Índice de tablas

CAPÍTULO 1

Introducción

1.1. Contexto

El Alzheimer es una enfermedad neurodegenerativa progresiva que afecta principalmente la memoria y otras funciones cognitivas, causando dificultades en el pensamiento, el comportamiento y la capacidad de llevar a cabo las actividades diarias. En la actualidad, esta es la principal causa de la demencia por encima de otras enfermedades como el Parkinson o las enfermedades cerebrovasculares [1]. De hecho, dos tercios de la población de más de 65 años que padece demencia es a causa del Alzheimer [2]. Se estima que en Estados Unidos entre un 5 % y un 10 % de la población de más de 65 años padece Alzheimer [3].

Estos datos representan una amenaza creciente en la sociedad actual, cada vez más envejecida, siendo su tratamiento y cura un objetivo primordial hoy en día. Por desgracia actualmente no existe ningún fármaco capaz de detener la progresión del Alzheimer o curarlo. Por lo tanto, a pesar de los avances realizados en los últimos años en diversos campos como por ejemplo el uso de biomarcadores o el empleo de terapias genéticas, aún se está lejos de conseguir una solución definitiva a dicha enfermedad.

El diagnóstico del Alzheimer suele venir tras la realización de numerosas pruebas cognitivas y de una punción lumbar. Dicha prueba resulta invasiva y dolorosa lo que provoca que un porcentaje alto de los afectados por la enfermedad rehúsen hacerse las pruebas. Además, otro de los problemas a la hora de diagnosticar el Alzheimer radica en el carácter neurodegenerativo de la enfermedad. Esto provoca que una vez la atrofia neuronal comienza, no es posible revertir los daños causados. Según el Plan Integral de Alzheimer y otras Demencias de 2019 del Ministerio de Sanidad,

un 70 % de los casos diagnosticados en España son en la etapa moderada o avanzada siendo imposible una recuperación total. Estos datos potencian la necesidad de tener métodos para la detección de la enfermedad de una manera menos invasiva que facilite la detección de la misma en estadios más tempranos de la enfermedad.

En este contexto, la fundación CITA y el grupo de investigación ALDAPA de la Facultad de informática de la UPV/EHU colaboran para hacer frente a esta problemática. La fundación CITA-alzhéimer es una fundación guipuzcoana privada sin ánimo de lucro, la cual tiene como objetivo el desarrollo y financiación de la investigación de la enfermedad del alzhéimer. Dentro de los proyectos de la fundación se encuentran los estudios PGA y DEBA, los cuales tienen como objetivo la investigación sobre el alzhéimer preclínico y prodrómico.

Los resultados obtenidos en estos proyectos servirán como punto de partida para el desarrollo de herramientas de aprendizaje automático por parte del grupo de investigación ALDAPA con el objetivo de desarrollar una herramienta que permita sustituir la punción lumbar por una resonancia magnética como medio de diagnóstico, la cual resulta menos invasiva y dolorosa. De esta manera, este nuevo método permitirá aumentar el número de gente dispuesta a realizarse la prueba, reduciendo así el porcentaje de enfermos no detectados.

1.2. Objetivo

El objetivo de este trabajo es el de desarrollar un proceso que permita mediante el uso de aprendizaje automático, realizar predicciones de diagnósticos de Alzheimer a partir de datos cognitivos y volúmenes cerebrales obtenidos mediante resonancias magnéticas.

Además, también se espera identificar qué características o atributos son más relevantes para el diagnóstico del Alzheimer.

CAPÍTULO 2

Planificación

En este capítulo se ha determinado la planificación del proyecto indicando los distintos paquetes de trabajo que forman el mismo, así como la duración estimada para cada uno de ellos. También se muestra tanto el análisis de riesgo como el seguimiento y control.

El proyecto está dividido en paquetes de trabajo que a su vez se agrupan en diferentes conjuntos dependiendo de su naturaleza. En la Figura 2.1 se puede ver el esquema de los paquetes de trabajo.

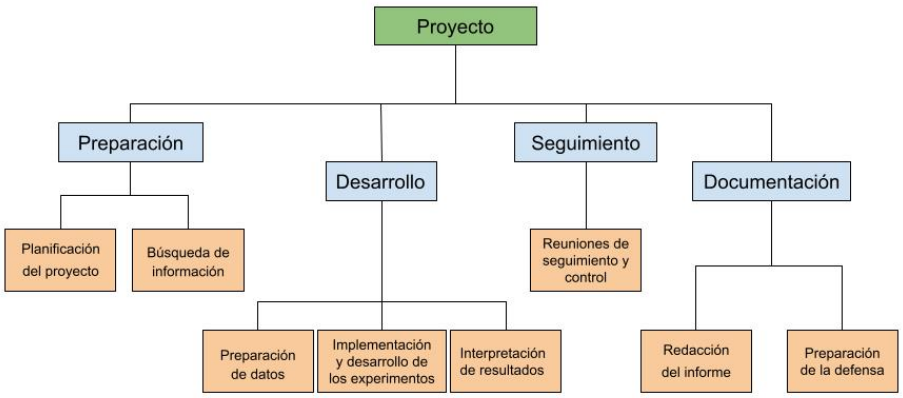

Figura 2.1: Esquema distribución de los paquetes de trabajo del proyecto

A pesar de la naturaleza diferente de los paquetes de cada conjunto, estos se encuentran, muy relacionados entre sí por lo que se han desarrollado de forma simultánea. Por otro lado, debido al carácter de cada paquete, es normal que en ciertos momentos del desarrollo del proyecto, ciertos paquetes hayan presentado una carga de

trabajo más elevada. Al inicio del proyecto, por ejemplo, el trabajo se ha centrado en el conjunto de trabajo de preparación. En la última etapa del proyecto, en cambio, el volumen de trabajo se centró en la documentación y defensa del proyecto.

Para comprender mejor cada grupo de paquetes de trabajo, a continuación se muestra un listado de los mismo incluyendo una pequeña descripción del trabajo que se ha realizado en cada uno de ellos.

- Preparación

  Paquetes relacionados con la preparación del proyecto. Estos paquetes son; el paquete de planificación del proyecto y el paquete de recopilación de información y conocimientos necesarios para poder realizar el proyecto.

- Desarrollo

  Paquetes que pertenecen al desarrollo del proyecto. Estos son: el trabajo previo necesario con los datos, el paquete de implementación y desarrollo de los experimentos y la interpretación de los resultados obtenidos.

- Documentación

  Paquetes relacionados con la defensa del proyecto, tanto la redacción del documento, como la preparación de la propia defensa.

- Seguimiento y control

  Paquete relacionado con el seguimiento y control del proyecto. Únicamente está formado por un paquete, el de reuniones de seguimiento y control. Este incluye el tiempo utilizado en las reuniones, así como el tiempo utilizado para la redacción de los informes necesarios para dichas reuniones.

En la Tabla 2.1 se muestran las horas adjudicadas a cada uno de los paquetes

Conjunto	Tarea	Duración Estimada
Preparación	Planificación del proyecto	15
	Búsqueda de información	25
Desarrollo	Preparación de datos	30
	Implementación y desarrollo de los experimentos	100
	Interpretación de resultados	25
Seguimiento	Reuniones de seguimiento y control	20
Documentación	Redacción del informe	70
	Preparación de la defensa	15
TOTAL		300

Tabla 2.1: Distribución de horas de trabajo por paquete

En la Figura 2.2 se puede observar más claramente la distribución de las horas. Analizando el gráfico, se puede observar como la mayor parte del trabajo se ha centrado en el conjunto de paquetes de desarrollo, los cuales han abarcado cerca de la mitad del tiempo total. Por otro lado, la segunda mitad del proyecto ha estado dividida entre la redacción y preparación de la defensa, el tiempo dedicado a la realización del seguimiento y control del proyecto y el tiempo dedicado a la planificación del mismo.

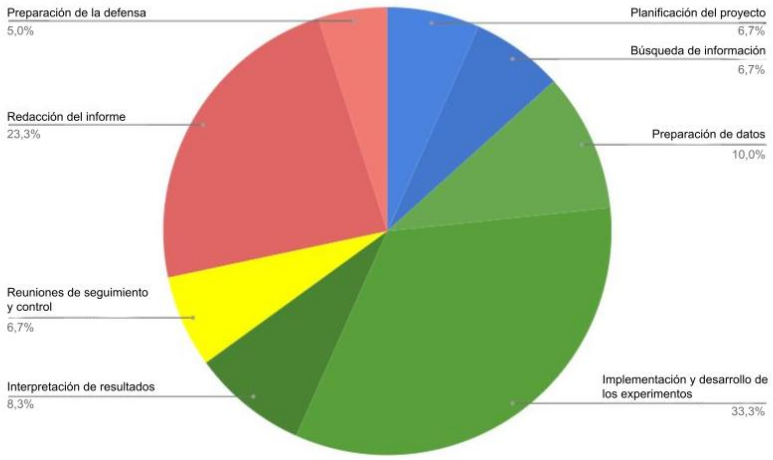

Figura 2.2: Porcentaje de horas de trabajo por paquete

2.1. Análisis de riesgos

- Extensión del trabajo:

  Este riesgo resulta destacable debido a la existencia de estudios similares que no han conseguido lograr el objetivo deseado. Si se diera este caso, se podría llegar a alargar la duración del proyecto enormemente. Ya que, en el transcurso del proyecto, a menos que se encuentre una solución óptima, siempre se va a intentar buscar otra solución hasta lograr el objetivo. En este caso el principal riesgo es el de no saber identificar correctamente cuando finalizar la parte experimental del proyecto.

- Falta de datos:

  La base de datos a utilizar en el proyecto es de un tamaño muy pequeño. Esto puede crear dificultades a la hora de encontrar una estrategia de validación eficaz. Esta falta de datos también puede generar problemas en el rendimiento de los clasificadores. Sin embargo es complicado conseguir más datos debido a la naturaleza de los mismos. Por lo tanto la estrategia que se plantea para paliar este riesgo es utilizar una estrategia de validación robusta.

- Modificaciones del proyecto durante su transcurso:

  Uno de los mayores riesgos en cualquier proyecto son los cambios de objetivos durante su transcurso. Dado que el proyecto también depende de la fundación CITA, es posible que durante el proyecto, debido a los resultados obtenidos o a las necesidades de la fundación, los objetivos de este proyecto varíen. Por otro lado, la posibilidad de que vayan llegando nuevos datos una vez ya se está trabajando, puede ocasionar la necesidad de replantear parte del proyecto.

2.2. Seguimiento y control

Para el seguimiento y control del trabajo realizado, se han realizado reuniones con los tutores del proyecto. Durante los primeros meses del proyecto las reuniones se realizaron semanalmente, hasta que la planificación del proyecto quedó clara y a partir de ese momento las reuniones se pasaron a realizar cada 2 o 3 semanas. Durante estas reuniones se fueron comentando los resultados del proyecto y los futuros pasos a realizar.

Tarea	Estimado	Real	Desviación
Planificación del proyecto	20	20	0 %
Búsqueda de información	20	23	+15 %
Preparación de datos	30	35	+16 %
Implementación y desarrollo de los experimentos	100	100	0 %
Interpretación de resultados	25	24	-4 %
Reuniones de seguimiento y control	20	23	+15 %
Redacción del informe	70	80	+14 %
Preparación de la defensa	15	15	0 %
TOTAL	300	320	+7 %

Tabla 2.2: Comparación duración planeada y real del trabajo

En la Tabla 2.2 se muestra la desviación de la duración de los paquetes de trabajo respecto a lo planificado. En esa misma tabla se puede observar un aumento de hasta el 10 % de horas de trabajo. Este aumento en horas de trabajo ha sido causado por el primer riesgo anteriormente comentado. Los resultados que se han ido obteniendo durante el transcurso del proyecto no han sido los deseados, por lo que se han tenido que buscar alternativas. Esto ha provocado un incremento sobre el planteamiento inicial de horas de trabajo programadas.

CAPÍTULO 3

Alzheimer

3.1. Introducción

En este capítulo se va a aportar información sobre la enfermedad del Alzheimer, destacando los factores de riesgo, síntomas, tratamiento, así como su detección. Finalmente se ha incluido un breve estado del arte sobre la detección de la enfermedad del Alzheimer mediante el uso de aprendizaje automático.

La enfermedad del Alzheimer es una enfermedad neurodegenerativa progresiva que se caracteriza por una formación de depósitos de proteínas que van creciendo en el cerebro, además de pérdidas de conexiones neuronales. Es una enfermedad degenerativa que a día de hoy no tiene cura.

3.2. Factores de Riesgo

Aunque no se conocen las causas exactas que llevan a desarrollar la enfermedad, sí que se conocen ciertos factores que pueden aumentar o disminuir el riesgo de padecer la enfermedad [4]. A continuación se enumeran los principales:

- La edad:

  Se sabe que uno de los factores de riesgo más importantes para el Alzheimer es la edad. La enfermedad pocas veces afecta a menores de 45 años y según se va envejeciendo los casos aumentan. Entre las edades de 45 a 65 años la enfermedad afecta a un 0.05 % de la población, y va aumentando hasta que un 40 % de la población de más de 90 años la padece [5].

- El factor hereditario:

  Se ha detectado que es más probable padecer la enfermedad si algún familiar la ha padecido. El gen asociado a la enfermedad más común es apolipoproteína-E (APOE) y dependiendo de qué alelo se tenga el riesgo de padecer la enfermedad aumenta.

- El estilo de vida:

  Tener un estilo de vida saludable como hacer deporte o mantener una dieta sana disminuye la probabilidad de padecer la enfermedad.

- El uso cerebral:

  De acuerdo con algunos autores [6] [7] mantener una vida social y ejercitar la memoria puede ayudar a disminuir la probabilidad de padecer la enfermedad.

3.3. Síntomas

Para comprender los síntomas del Alzheimer, primero se ha de diferenciar las etapas de la enfermedad. Esta se puede dividir en 4 etapas, cada una de las mismas se diferencia por sus síntomas [8]:

- Preclínica:

  En esta etapa el paciente todavía no padece ningún síntoma visible y solo es posible detectar la enfermedad a través de pruebas médicas. Esta etapa puede durar años o décadas.

- Leve:

  En esta etapa comienzan a manifestarse los primeros síntomas, aunque todavía son muy leves y pasan desapercibidos. Normalmente en esta etapa se suele empezar a sufrir leves pérdidas de memoria sobre hechos recientes.

- Moderada:

  Esta es la etapa donde se suele detectar la enfermedad. En este caso los síntomas son más notables y comprenden el olvido de palabras y hechos, cambio de personalidad y desorientación. Sin embargo, el afectado sigue pudiendo realizar una vida medianamente normal.

- Severa:

  En esta etapa los síntomas ya son claros y graves tanto que el paciente no puede realizar una vida normal y necesitan asistencia total. En esta etapa los

síntomas son: la pérdida total de memoria, cambio de personalidad, imposibilidad de realizar las tareas cotidianas y la dificultad para hablar.

Cabe señalar que la mayoría de estos síntomas no son exclusivos del Alzheimer y que son comunes en la mayoría de enfermedades pertenecientes al grupo de la demencia. Por lo tanto, puede que, aun padeciendo estos síntomas, el afectado no padezca Alzheimer, si no otra enfermedad similar.

3.4. Tratamiento

Hoy en día no existe ningún medicamento capaz de ralentizar o parar la enfermedad completamente. Sin embargo, existen tratamientos para contrarrestar o minimizar los síntomas de la enfermedad a través de medicamentos o terapias cognitivas. Dichos tratamientos se vuelven cada vez menos efectivos según avanza la enfermedad.

Recientemente, han salido a la luz diversos medicamentos en fase experimental que parecen ralentizar la enfermedad en torno a un 30 % [9][10]. Sin embargo, estos fármacos están todavía fase de en estudio ya que generan efectos secundarios entre los que se encuentra la inflamación cerebral [10].

3.5. Detección

Como se ha mencionado anteriormente, los conocimientos acerca de la enfermedad no son muy amplios, por lo que no existe un método eficaz para su detección. Además, se han de realizar numerosas pruebas para su detección, siendo complicado establecer un diagnóstico fiable de si la demencia está provocada por el Alzheimer o por otra enfermedad ya que estas pruebas no permiten determinar esto último.

Entre las pruebas que se realizan hay test cognitivos u otras pruebas no invasivas como resonancias magnéticas [11]. Pero la que más precisión tiene en estos momentos es la punción lumbar. Esta prueba consiste en la extracción del líquido cefalorraquídeo y la comprobación del nivel de las proteínas Beta-amiloide, Tau y Phospho-tau que son los indicadores más importantes de la enfermedad.

Dado que este diagnóstico es invasivo y ante la ausencia de tratamiento efectivo contra la enfermedad, mucha gente opta por no realizar la punción lumbar, lo que conlleva a que ahora mismo se estime que un 50 % de los afectados por la enfermedad no estén diagnosticados.

3.6. Estado del Arte en la detección del Alzheimer mediante el uso de aprendizaje automático

Recientemente, numerosos artículos han sido publicados sobre la detección de la enfermedad del Alzheimer mediante el uso de aprendizaje automático. Algunos de ellos utilizando técnicas similares a las utilizadas en este proyecto [12], y otros utilizando técnicas de aprendizaje profundo [13].

Por otro lado, también existen trabajos en la literatura en la que se realiza una aproximación distinta, proponiendo otro tipo de pruebas médicas aparte de las resonancias magnéticas. En ese contexto, se han explorado alternativas como el uso de escáneres de retina [14] o el uso del aprendizaje con datos del genoma [13] entre otros.

Finalmente cabe destacar el trabajo desarrollado por *Inglese et al* [12], el cual, partiendo de datos similares a los utilizados en este proyecto, realiza un filtrado y preprocesado de las imágenes resultado de la resonancia magnética. Como resultado, se obtuvo hasta un 98 % de "Accuracy" en un conjunto de datos de prueba. No obstante, los resultados obtenidos en otro conjunto de datos resultan muy inferiores por lo que resulta complicado predecir su validez si se aplicase a la base de datos presentada en este trabajo.

CAPÍTULO 4

Marco Teórico

4.1. Aprendizaje automático

El aprendizaje automático es una rama de la computación que intenta crear algoritmos que repliquen el proceso de razonamiento humano. Con el fin de lograr este objetivo, se han desarrollado diversas técnicas y algoritmos. Estos algoritmos intentar realizar una tarea a través de un conocimiento previo y un proceso de aprendizaje.

Dentro del aprendizaje automático existe más de un tipo de algoritmos. Como los de aprendizaje supervisado y no supervisado. Este trabajo se va a centrar en los pertenecientes al campo del aprendizaje supervisado. Estos Algoritmos se entrenan con datos previamente etiquetados para así aprender a clasificar datos desconocidos. A continuación se enumeran los diferentes tipos de algoritmos, a los cuales se conocen también como clasificadores. Además, también se expondrán otras técnicas del campo de la computación utilizadas durante el proyecto.

4.1.1. Support Vector Machine

Las "Support Vector Machine" (SVM) o, en castellano, Máquinas de vectores de soporte son un tipo de algoritmo de aprendizaje supervisado desarrollado en 1995 por Vladimir Vapnik y Corinna Cortés [15]. El SVM es un clasificador que está diseñado para problemas binarios, pero que a través de técnicas como el "one vs one" o "one vs all" puede llegar a ser utilizado para problemas multiclase [16].

Representando cada dato como un vector de dimensión "p", el SVM intentará encontrar un hiperplano de dimensión "p-1" que divida los datos en dos regiones que corresponderán cada una de ellas a una clase. Puede que exista más de un

hiperplano capaz de realizar esta tarea, por lo que el SVM escogerá el hiperplano que maximice el margen, el cual viene definido por la distancia entre el hiperplano y el vector de cada clase más cercano [17]. Para realizar esta búsqueda de forma óptima, el SVM trabajará con los productos vectoriales de los datos.

Si no fuera posible separar los datos con un hiperplano lineal, se pueden buscar hiperplanos no lineales. Para realizar esto, el algoritmo lleva los datos a un espacio de mayor dimensión en el cual si sea posible realizar esa separación lineal. Realizar esta transformación puede ser muy costosa, por lo que se utilizará un método denominado "kernel trick" (trucos de kernel) que permite simplificar el proceso [18].

Para ello el "kernel trick" permite calcular los productos vectoriales de los datos sin necesidad de transformar los datos a la nueva dimensionalidad aplicando una función kernel adecuada sobre los datos originales que es equivalente a realizar el producto vectorial sobre los datos transformados. Existen distintos tipos de funciones de kernel que se pueden utilizar para esta transformación, las más comunes son la polinomial (4.1) y la base radial Gaussiana (4.2):

$$K_{polinomial}(u, v) = (u \cdot v + r)^d \tag{4.1}$$

$$K_{BRG}(u, v) = exp\left\{-\frac{|u - v|^2}{\sigma^2}\right\} \tag{4.2}$$

Donde:

- σ es un parámetro a decidir antes de la clasificación, esta muy relacionado con el parámetro gamma y sirve para indicar cuanto queremos que se curve nuestro hiper plano.
- r un parámetro a decidir antes de la clasificación, también llamado coste, y es el utilizado para indicar cuanto se castiga los errores de clasificación.
- d un parámetro a decidir antes de la clasificación, el cual indica el grado del polinomio.

4.1.2. K-Nearest Neightbors

KNN (K-Nearest Neightbors) o, en castellano, K- Vecinos Cercanos es un algoritmo supervisado de clasificación que, debido a su simplicidad e interpretabilidad, es un algoritmo muy utilizado. Además, no necesita de fase de entrenamiento para su uso [19].

El KNN asume que aquellas instancias que más cerca estén entre sí pertenecen a la misma clase, por lo que el clasificador calcula la distancia entre el nuevo caso a clasificar y el resto de instancias [20] y clasificará la instancia nueva haciendo una votación de mayoría entre las clases de las "k" instancias más cercanas. Dependiendo del valor de "k" escogido, el clasificador puede sufrir de subreajuste o sobreajuste. Existen distintas métricas para medir la distancia, siendo las más habituales la distancia de Manhattan:

$$D_{manhattan}\left(p,q\right) = \sum_{i=1}^{n} |q_i - p_i| \tag{4.3}$$

o la distancia euclídea:

$$D_{Euclidea}\left(p,q\right) = \sqrt{\sum_{i=1}^{n} \left(q_i - p_i\right)^2} \tag{4.4}$$

4.1.3. Naive Bayes

NB (Naive Bayes) es un algoritmo de clasificación probabilista basado en el teorema de Bayes (4.5) [21], el cual indica que:

$$\Pr(A|B) = \frac{\Pr(B|A)\Pr(A)}{Pr(B)} \tag{4.5}$$

Donde:

- $\Pr(X|Y)$ = Probabilidad del suceso X conociendo de antemano el suceso Y.
- $\Pr(X)$ = Probabilidad del suceso X.

Utilizando este teorema y, asumiendo que los atributos son independientes entre sí, "Naive Bayes" calcula la probabilidad de que una instancia pertenezca a una clase conociendo el valor de sus atributos. La asunción que realiza "Naive Bayes" no suele darse en la realidad, pero permite que el clasificador funcione de forma eficiente en entornos de alta dimensionalidad.

Dependiendo de la distribución de los datos, existen distintos tipos de "Naive Bayes". Cada tipo de clasificador asume que los datos están distribuidos de una manera distinta [22], siendo los principales: "Gaussian naive Bayes", que se utiliza para atributos continuos, "Multinominal Naive Bayes", que se utiliza con atributos discretos y "Brenouilli Naive Bayes", que es utilizado para atributos binarios.

Uno de los principales problemas de NB es que suele funcionar mal en conjuntos de datos desbalanceados, como es el caso del presente trabajo.

4.1.4. C4.5

El C4.5 es un algoritmo de árbol de decisión desarrollado Ross Quinlan [23]. Los algoritmos de árboles de decisión se caracterizan por el uso de una estructura en forma de árbol para clasificar las nuevas instancias.

En estos árboles, los nodos internos representan un atributo y umbral o una regla que se aplica a ese atributo para dividir los datos en ramas en función de su valor.

En cada uno de los niveles del árbol se elige el mejor atributo para dividir los datos, habiendo más de una forma de elegir el mejor atributo. En el caso de C4.5, se utiliza la relación de ganancia, la cual, tiene en cuenta la información mutua entre las características y la clase [23].

Otra característica de C4.5 respecto a otros algoritmos de árboles de decisión es que utiliza la poda para eliminar las ramas irrelevantes que pueden causar sobre-ajuste y añadir costes computacionales.

4.1.5. Random Forest

El algoritmo Random Forest crea múltiples árboles de decisión. Cada uno de los cuales, utiliza una submuestra aleatoria de los datos con reemplazo (Bagging) y, a continuación, emplea un sistema de votación por mayoría con los resultados de todos los árboles para clasificar los datos.

En cada uno de los niveles de los árboles de decisión, el atributo se selecciona a partir de una submuestra generada aleatoriamente de la totalidad de los atributos. El número de atributos de esta submuestra suele ser la raíz cuadrada del número

total de atributos.

El árbol de decisión generado por el "Random Forest" es de tipo "CART" y difiere del C4.5 en el método utilizado para elegir los atributos óptimos. En este caso, en lugar de utilizar la relación de ganancia como hace el C4.5, utiliza la medida de impureza "Giny" que se basa en la probabilidad de que una instancia desconocida sea mal clasificada [24].

La ventaja del "Random Forest" respecto a otro tipo de árbol como el C4.5 es que mantiene los beneficios de la decisión y al mismo tiempo, elimina el problema del sobreajuste. Sin embargo, también es computacionalmente más caro y pierde la interpretabilidad de los árboles de decisión.

4.1.6. Consolidated Tree Construction algorithm

El "Consolidated Tree Construction algorithm" (CTC) es un método de construcción de árboles de decisión que, entre otros beneficios, permite lidiar con el desequilibrio de clases.

Esta metodología, al igual que el "Random Forest", genera múltiples propuestas de árboles de decisión a partir de submuestreos de data. A diferencia del "Random Forest" que construye múltiples árboles y utiliza su predicción para clasificar, el CTC generará un único árbol final. Este árbol se genera a partir de los árboles propuestos de cada uno de los submuestreos de data. Los árboles propuesta generados además lo serán siguiendo el algoritmo C4.5.

Para generar el árbol final, en cada uno de los niveles del árbol se escoge el atributo que más veces haya sido escogido entre los múltiples árboles propuesta generados. Además, esta decisión también se aplica sobre los árboles propuesta que no hubieran elegido dicho atributo. De esta forma, cuando lleguemos al final del árbol y no se pueda seguir dividiendo más ramas, obtendremos el árbol final [25].

Una ventaja del uso de este clasificador es que, además de obtener la ventaja del "Bagging", a diferencia del "Random Forest", este si consigue un resultado que puede ser interpretado ya que el resultado sigue siendo un único árbol.

4.2. Sobremuestreo de datos

En este apartado se describirán las técnicas de sobremuestreo de datos utilizadas en el transcurso el proyecto. Estas técnicas crean de nuevos datos con la finalidad de conseguir un conjunto de datos equilibrado.

4.2.1. Synthetic Minority Oversampling Technique

"Synthetic Minority Oversampling Technique" (SMOTE) o, en castellano, Técnica de Sobremuestreo Sintético de Minorías, es una técnica de aumento de datos desarrollada en 2002 [26]. Esta técnica se aplica a conjuntos de datos con desbalanceo de clase, con la intención de crear instancias pertenecientes a la clase minoritaria, para así conseguir un conjunto de datos con una distribución de clase equilibrada[26].

Para crear nuevas instancias, SMOTE escoge las instancias de la clase minoritaria ya existente y selecciona las "k" instancias de la misma clase más cercanas a cada una de ellas, siendo "k" un valor que depende de cuántas instancias se quieran crear. Posteriormente crea las nuevas instancias interpolando los valores entre los vecinos y las instancias ya existentes.

El uso de esta técnica puede mejorar la clasificación de la clase minoritaria, dado que genera más muestras distintas de esta clase. Por otro lado, el que las instancias de la clase minoritaria sean muy similares entre sí, puede llegar a generar problemas de sobreajuste.

4.3. Seleccion de atributos

Los métodos de "Feature Selection" o selección de atributos en castellano son aquellas técnicas que a través de distintos procesos o algoritmos intentan seleccionar el mejor subconjunto de atributos para una tarea en específico. Hay distintos tipos de técnicas, por un lado podemos encontrar aquellas que evalúan los subconjuntos a través del resultado de la clasificación mientras que también hay otras que únicamente utilizan la relación de los atributos y la clase.

El uso de estos métodos permite deshacerse de atributos irrelevantes que pueden añadir ruido a la hora de realizar la clasificación. Además permite disminuir la dimensión de los datos para poder reducir el coste computacional en casos de alta dimensionalidad.

4.3.1. Correlation-based Feature Selection

"Correlation-based Feature Selection" (CFS) o, en castellano, Selección de Atributos basada en la Correlación es un método de selección de atributos, que como su nombre indica, elige los atributos en función de la correlación. Este método es ampliamente utilizado en el aprendizaje automático, debido a su eficiencia computacional.

El método selecciona aquellos atributos que, mayor correlación tienen con la clase, pero que menos correlación tienen entre sí. Como la correlación se define como la medida que mide la relación lineal entre dos variables, por tanto el CFS selecciona los atributos que más información aportan a la hora de clasificación, dejando fuera aquellos atributos cuya información ya la aporta otra característica [27].

4.3.2. Wrapper

"Wrapper" es un método de selección de atributos que intenta seleccionar el mejor subconjunto de atributos para la clasificación, probando directamente los subconjuntos sobre un clasificador [28].

El método genera subconjuntos de atributos y utiliza un modelo para evaluar la utilidad de cada atributo, entrenando el modelo con diferentes combinaciones de atributos y seleccionando las que mejor rendimiento tienen.

Para generar los subconjuntos, el algoritmo comienza con un conjunto vacío o completo de atributos, e iterativamente añade o elimina una característica cada vez. Para la selección de dichos atributos utiliza un algoritmo de búsqueda como la búsqueda en anchura o algún algoritmo genético.

Este método presenta una gran ventaja al usar un clasificador para elegir el subconjunto, lo que permite obtener el mejor subconjunto de atributos para ese clasificador. Por otro lado, este tipo de método es mucho más costoso que CFS y los resultados obtenidos no suelen ser mejores que los de este [29].

4.4. Optimización de Hiperparámetros

La optimización de hiperparámetros es un conjunto de técnicas que permiten encontrar los valores óptimos de los hiperparámetros de un algoritmo para una tarea en particular.

En el aprendizaje automático estas técnicas o métodos suelen aplicarse a los modelos de clasificación. Para ello, se prueba a utilizar el clasificador con diferentes valores de hiperparámetros y se comprueba el resultado.

Existen diversos métodos para buscar estos valores, los más utilizados son: búsquedas exhaustivas como el "gridsearch", algoritmos de búsqueda aleatorios o algoritmos más eficientes basados en probabilidad o evolutivos como el "gradient descent" o la optimización bayesiana.

4.5. Criterios de bondad

4.5.1. Matriz de confusión

La matriz de confusión es una herramienta gráfica que permite evaluar un clasificador. Esta matriz se representa en forma de tabla que permitirá ver los fallos y aciertos del clasificador dividiendo los casos según su clase real y su clase predicha por el clasificador.

		Valores Reales	
		Positivos	Negativos
Valores Predichos	Positivos	TP	FP
	Negativos	FN	TN

Tabla 4.1: Matriz de Confusión

En el caso de un problema binario la tabla tendrá dimensiones 2x2 como se puede ver en la Tabla 4.1. Los casos estarán divididos de la siguiente forma:

- Casos que siendo positivos han sido clasificados como tal denominados TP (verdaderos positivos).
- Casos que siendo positivos fueron clasificados como negativos, denominados FN (Falsos Negativos).
- Casos negativos que fueron debidamente clasificados, conocidos como TN (verdaderos negativos).
- Casos que siendo negativos fueron clasificados como positivos, denominados FP (Falsos positivos).

A partir de estas métricas se podrá calcular otras como:

- El "Accuracy" que nos indica el porcentaje de instancias correctamente clasificadas:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.6}$$

Aunque parezca que con el porcentaje de acierto es suficiente para evaluar un clasificador, la realidad es que hay en ciertas ocasiones en las que el "Accuracy"

no refleja de forma correcta la eficiencia del clasificador (como en el caso del presente estudio) [30]. Por lo tanto es mejor utilizar otras métricas que aportan más información del resultado, como las siguientes:

- La "Precision" que indica dentro de las instancias clasificadas como positivas que porcentaje ha sido correctamente clasificado:

$$Precision = \frac{TP}{TP + FP} \tag{4.7}$$

- El "Recall" que indica cuantas instancias pertenecientes a la clase positivas han sido debidamente clasificadas:

$$Recall = \frac{TP}{TP + FN} \tag{4.8}$$

- El "F1-score" que junta los resultados de la "Precision" y el "Recall" para crear una única métrica:

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \tag{4.9}$$

- La métrica "Fβ-score" que es una generalización de "F1-score" y que a través del parámetro "β" nos permite dar más importancia a la "Precision" o al "Recall":

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{Precision} \cdot \text{Recall}}{(\beta^2 \cdot \text{Precision}) + \text{Recall}}. \tag{4.10}$$

4.6. Estrategia de validación

Debido a que la idea detrás de los algoritmos de clasificación es que consigan una buena generalización de los datos con los que se ha entrenado, el conjunto de datos ha de ser dividido en subconjuntos.

Por un lado existe el denominado subconjunto de entrenamiento, que estará compuesto por aquellos datos que se utilizarán para entrenar el clasificador. Mientras que el conjunto de prueba que será únicamente utilizado para determinar la eficacia del clasificador. Además, en algunos casos se utiliza un tercer subconjunto denominado subconjunto de validación que estaría compuesto por datos que no serán utilizados durante el entrenamiento. Estos datos serán utilizados para las tomas de decisiones a la hora del desarrollo del clasificador como puede ser en el ajuste de hiperparámetros.

Es importante ser precavido a la hora de dividir el conjunto inicial dado que un conjunto de datos de entrenamiento muy pequeño puede generar problemas de sobreajuste o un conjunto de test muy pequeño puede hacer que el resultados obtenido no pueda ser generalizado.

Existen distintas técnicas para realizar esta separación, la principal y más simple es el "Houldout", donde el conjunto de datos se divide según un porcentaje (80/20 % por ejemplo) o como "Cross Validation" que permite que a partir de un conjunto de datos pequeño se pueda sacar un resultado válido.

4.6.1. Cross Validation

La validación cruzada es una estrategia ampliamente utilizada de aprendizaje automático para validar el rendimiento de un modelo. Esta estrategia crea múltiples conjuntos de datos a partir del conjunto de datos original, con la idea de obtener una mejor estimación del comportamiento del modelo a la hora de la generalización.

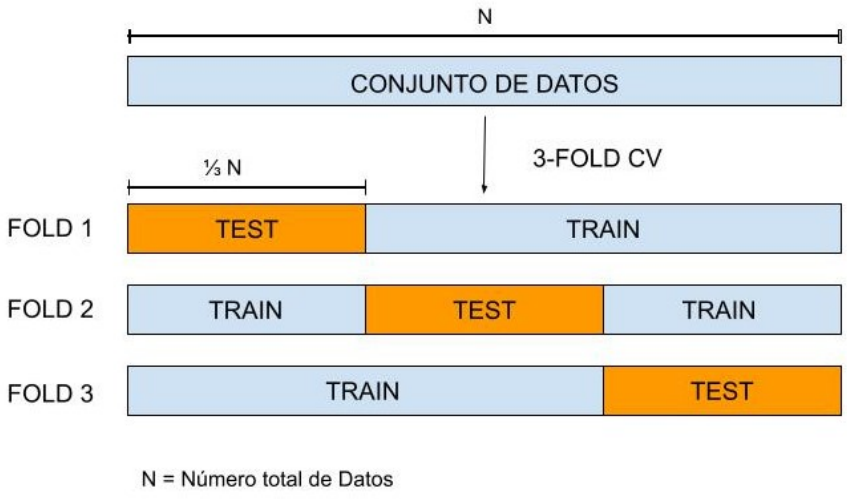

Figura 4.1: Ejemplo 3-Fold Cross Validation

La idea principal de este método es dividir la totalidad del conjunto de datos en múltiples subconjuntos, para después realizar un "Houldout"(utilizar un porcentaje de los datos para entrenamiento y otro para test), agrupando de distintas formas estos subconjuntos múltiples veces. De esta manera, a partir de un único conjunto de datos se pueden obtener más de un conjunto de datos que, aunque compartan algunos datos de entrenamiento son probados con datos distintos (Figura 4.1).

De entre las diferentes técnicas de validación cruzada, este proyecto utiliza "k-fold Cross Validation", en la cual el conjunto de datos es dividido en "k" partes iguales y con las cuales se formarán "k" conjuntos de datos distintos en los que el conjunto de prueba estará formado por una de estas "k" partes. El conjunto de entrenamiento lo formarán el resto de las partes que anteriormente dividimos como se puede observar en la siguiente Figura 4.1.

Además, para intentar mantener la misma distribución de datos en cada uno subconjuntos, se puede realizar un muestreo estratificado en vez de realizarlos de forma aleatoria. De esta manera, se intenta que la distribución de la clase sea similar entre cada subconjunto y el conjunto inicial.

CAPÍTULO 5

Tecnologías

5.1. Introducción

En este apartado se van a presentar las principales herramientas y tecnologías utilizadas durante el proyecto y la razón de su elección.

5.2. Python

Python es un lenguaje de programación de alto nivel que junto a R es uno de los lenguajes más ampliamente utilizado en el campo del aprendizaje automático y la computación. Esto se debe a que gracias a su sencillez y flexibilidad facilita el desarrollo de librerías y herramientas para el aprendizaje automático. Además, también permite que profesionales del ámbito científico puedan utilizarlo sin la necesidad de tener un amplio conocimiento en el campo de la programación.

El presente proyecto se ha realizado utilizando Python ya que existen librerías para este lenguaje que implementan gran parte de los algoritmos y herramientas presentados en el capítulo anterior.

Python tiene también sus desventajas, siendo una de las mayores su poca eficiencia. Sin embargo esto puede ser solucionado utilizando ciertas librerías que agilizan las operaciones matemáticas.

Dentro de la gran cantidad de librerías que se pueden utilizar en Python a continuación se enumeran las más importantes que se han utilizado en este proyecto:

5.2.1. Librería Numpy

"Numpy" es una librería de Python especialmente diseñada para trabajar con vectores y matrices añadiendo herramientas matemáticas para poder operar con estas estructuras de forma eficiente.

Esta, es una librería fundamental para el uso de Python ya que facilita el trabajo con matrices. Además, esta librería es también utilizada por la mayoría de librerías de aprendizaje automático, incluyendo el resto de librerías mencionadas en este capitulo.

5.2.2. Librería Pandas

"Pandas" es la principal librería de Python para el uso y manejo de datos. Esta librería permite tener el conjunto de datos en un tipo de estructura denominada "DataFrame" con el que se puede trabajar y operar. Al igual que "Numpy" muchas de las librerías de aprendizaje automático trabajan utilizando la estructura "DataFrame" por lo que esta es una herramienta fundamental para poder utilizar otras librerías como por ejemplo "Scikit-learn". Además, permite importar datos con formato ".csv" y trabajar con ellos.

5.2.3. Librería scikit-learn

"Scikit-learn" es la principal librería en Python para aprendizaje automático. Aunque existen otras librerías como "Pytorch" o "Keras", estas se centran en redes neuronales o ámbitos más precisos del aprendizaje automático. "Scikit-learn" , en cambio, contiene algoritmos de regresión y clasificación más tradicionales o que no utilizan redes neuronales. Se ha optado por el uso de clasificadores más tradicionales en este trabajo debido a dado que el conjunto de datos es de tamaño reducido, y estos clasificadores se adaptan mejor a esta situación.

Esta, por lo tanto, es la librería que tendrá implementaciones de algoritmos como "KNN", "SVM", "NB" y la mayoría de clasificadores que se han usado en este proyecto y que se han descrito en el apartado anterior. Además, también implementa las herramientas para realizar el "Cross-Validation" o el "Wrapper".

5.2.4. Librería Hyperopt

Hyperopt es una librería especializada en la optimización de hiperparámetros. Implementa distintos algoritmos de optimización de hiperparámetros, entre ellos la optimización bayessiana a través del Tree-structured Parzen Estimator o Estimador

de Parzen estructurado en árbol. Esta librería será utilizada en el proyecto para intentar mejorar el resultado del algoritmo SVM.

5.2.5. Otras librerías

Además de las librerías anteriormente mencionadas, se han utilizado en menor medida otras para ciertas herramientas. Entre estas, destacan las librerías que contienen el algoritmo de selección de atributo CFS o las que implementan los algoritmos de C4.5 o SMOTE.

5.3. Weka

Waikato Environment for Knowledge Analysis (Weka) es un software libre desarrollado para el aprendizaje automático por la universidad de Waikato en Nueva Zelanda.A diferencia de Python, este no utiliza librerías si no que utiliza paquetes, los cuales permiten implementar herramientas o algoritmos de forma sencilla.

La razón principal del uso de este software ha sido la necesidad de utilizar el algoritmo CTC. Este algoritmo solo esta implementado en un paquete de este programa.

5.3.1. Paquete J48Consolidated

El único paquete o módulo de WEKA utilizada en este proyecto es J48Consolidated. Esta herramienta que implementa el algoritmo de clasificación CTC es un módulo diseñado por profesores de esta facultad y es una implementación que permite el uso y modificación de este algoritmo de forma sencilla.

CAPÍTULO 6

Estudio de la base de datos

6.1. Introducción

En este apartado se va a describir la base de datos utilizada. Los datos utilizados para el proyecto pertenecen a los estudios PGA y DEBA de la fundación cita Alzheimer. Estos datos se dividen en dos bases de datos distintas las cuales se han juntado en este trabajo para poder trabajar con ambas simultáneamente. Ambas comparten parte de los sujetos, por lo que ha sido posible unirlas.

La primera de las bases de datos contenía los datos de las resonancias magnéticas, así como los resultados de las punciones lumbares, mientras que la segunda presentaba los resultados de pruebas cognitivas y clínicas de los sujetos.

A continuación, se detalla de una manera más exhaustiva los datos presentes en cada una de las bases de datos originales, así como la distribución de la base de datos final.

6.2. Datos Volumétricos

La base de datos volumétricos está compuesta por 537 casos, 210 casos pertenecientes al estudio DEBA y 327 casos pertenecientes a pacientes del estudio PGA.

Además de género y edad, la base de datos contiene los siguientes atributos:

- Datos volumétricos:

  La base de datos está compuesta por los resultados de la resonancia magnética. Contiene el volumen intracraneal, así como volúmenes de fluido cerebroespi-

nal, materia gris y materia blanca de 68 zonas del cerebro, cada una dividida en dos volúmenes según su hemisferio.

- Punción lumbar:

  Con la punción lumbar se consiguen los valores de las proteínas Beta-amiloide, Tau y Phospho-tau, las cuales sirven para clasificar el problema. En el presente trabajo se ha utilizado como clase el marcador de la proteína Beta-amiloide, ya que es la más importante de las tres de acuerdo con el criterio de los expertos de la fundación CITA .

6.3. Datos Cognitivos y Clínicos

Esta base de datos también fue obtenida con los datos de los estudios DEBA y PGA, y contiene 687 sujetos.

La información de los sujetos es la siguiente:

- Datos demográficos:

  En estos datos están recogidos la edad, sexo y el nivel educativo del sujeto.

- Datos cognitivos:

  En este apartado se encuentran los resultados de diferentes exámenes cognitivos como el test de figura compleja de Rey o "Trail Making Test" entre otros.

- Datos clínicos:

  En este grupo se enmarcan las métricas de "Critical Dementia Risk" o del "Caide Risk score".

- Datos genéticos:

  El único dato que pertenece al ámbito genético es relativo a los tipos de alelos del gen APOE que porta cada sujeto.

6.4. Base de Datos

Tras juntar ambos conjuntos de datos se han seleccionado aquellos de los que se tienen los resultados de la punción lumbar y en los que el porcentaje de datos perdidos es reducido. Finalmente se hace un cribado reduciendo el número de casos

a 359 casos. Además, en las Tablas 1 y 2 del Apéndice se listan la totalidad de atributos presentes en la base de datos.

En la Tabla 6.1 se puede ver la distribución de los sujetos que se sometieron a la punción lumbar según el estudio.

Estudio	Casos	Clasificados	Sin Clasificar
DEBA	210	157	53
PGA	327	202	125
Total	537	359	178

Tabla 6.1: Número de los casos clasificados según el estudio

De esos 359 casos en los que se tienen resultados de la punción lumbar, 107 casos dieron positivo en el indicador de la proteína Beta-amiloide frente a 252 que dieron negativo, tal y como se puede apreciar en la Figura 6.1.

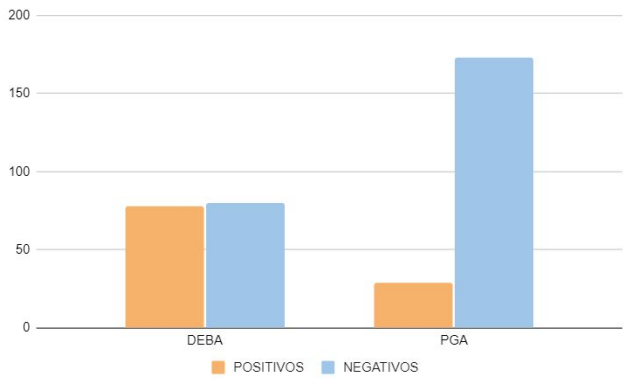

Figura 6.1: Distribución de clase en cada estudio

Como se puede apreciar hay una gran diferencia entre la distribución de los dos estudios. Esto se puede deber a que la edad media en cada uno de los estudios es muy diferente, siendo la edad media de los sujetos de PGA de 57 años y la correspondiente a los del DEBA 67.

A partir de este punto se va a presentar la distribución de ciertas características dentro de la base de datos que van a ser de utilidad durante el proyecto. Se va a trabajar con los 359 casos que están clasificados mediante punción lumbar, utilizando para su clasificación el aprendizaje supervisado. El resto de datos no clasificados no van a ser utilizados en el proyecto.

Dentro de esos 359 casos, hay 192 mujeres frente a 167 hombres (Tabla 6.2). Además, también se puede observar la distribución de los casos según la edad, que se muestra dividida en 3 grupos.

	Casos	<60	<60 y >70	>70
Mujeres	192	79	77	36
Hombres	167	51	95	21
Total	359	130	172	57

Tabla 6.2: Distribución de la base de datos según género y edad

Debido a que la edad es un factor de riesgo importante a la hora de padecer la enfermedad, así que vamos a comentar un poco la distribución de los sujetos según la edad. Así en la Figura 6.2 podemos observar cómo según la edad aumenta el número de casos con Alzheimer sigue la misma tendencia. Además, en las Figuras 6.3 6.4 se muestra la distribución de la clase entre varones y mujeres respectivamente.

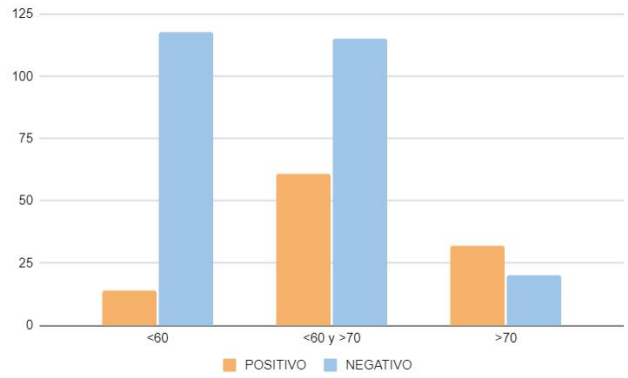

Figura 6.2: Distribución de la clase según los grupos de edad

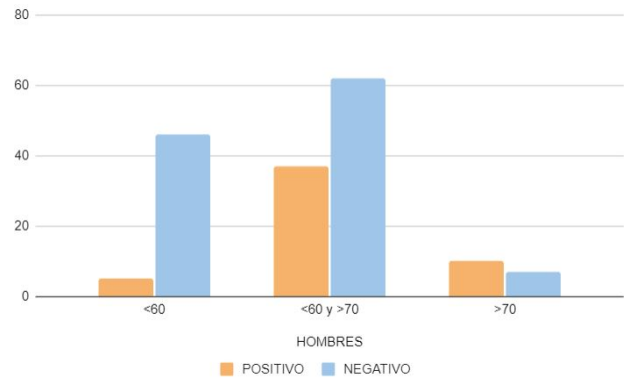

Figura 6.3: Distribución de la clase según los grupos de edad en el caso de los hombres

Analizando estos datos, se observa que aun habiendo muchos menos casos de más de 70 años, el número de casos positivos es muy elevado entre los mismos. Esto es especialmente significativo entre las mujeres, en las cuales hay proporcionalmente

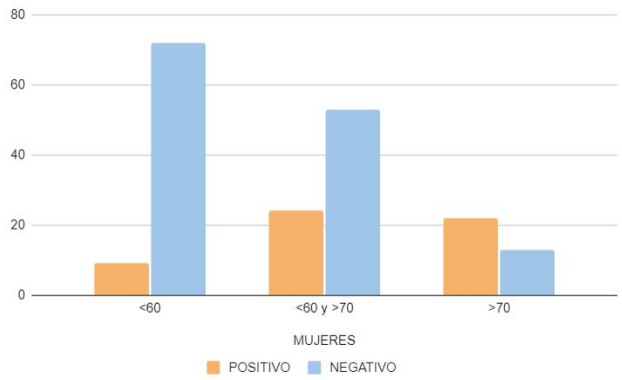

Figura 6.4: Distribución de la clase según los grupos de edad en el caso de las mujeres

el doble de casos de positivos en el grupo de mas de 70 años respecto al grupo de 60 y 70 años.

Finalmente la Figura 6.5 muestra la segregación de los casos por sexos, mostrando valores similares en cuanto al número de positivos de mujeres y hombres, siendo este último algo mayor.

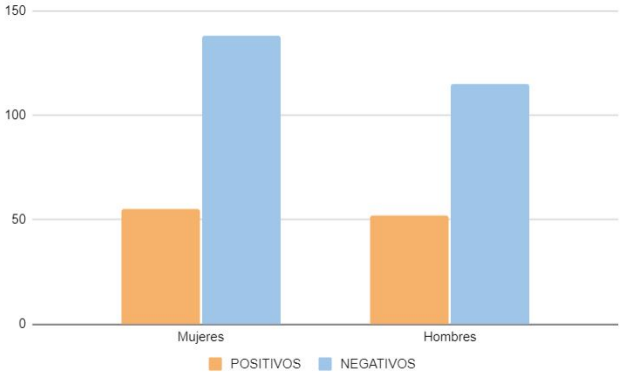

Figura 6.5: Distribución de la clase según el género

De todo lo comentado anteriormente se pueden sacar las siguientes conclusiones:

Existen dos grandes problemas con la base de datos:

- La alta dimensionalidad de los datos (hay 235 atributos y solo 359 casos).
- El desbalanceo de la clase (la distribución es un 70/30).

La edad es un atributo muy importante ya que, como se puede observar en la Figura 6.2, en el grupo de casos de mayor edad el número de casos positivos es muy alto. También se puede observar en la Figura 6.5 que la distribución de casos positivos entre hombres y mujeres es similar.

CAPÍTULO 7

Desarrollo del proyecto

En este apartado se va a describir el desarrollo del proyecto, así como los principales resultados obtenidos. El proyecto se ha desarrollado de forma iterativa, es decir, primero se ha comenzado desde una clasificación sencilla y paso a paso se han ido introduciendo mejoras hasta llegar al resultado final. En la Figura 7.1 se muestra el esquema del desarrollo del proyecto.

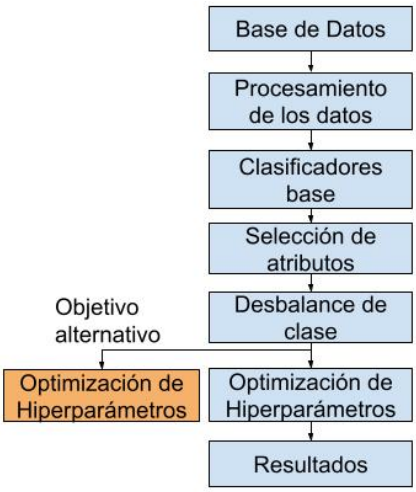

Figura 7.1: Esquema del desarrollo del proyecto

Como se puede observar, el desarrollo del proyecto está dividido en 7 partes, cada una de ellas pudiendo tener a su vez subapartados. En las siguientes secciones de este capítulo se comentará detalladamente el trabajo realizado en cada una de dichas partes.

7.1. Preprocesamiento de los datos y estrategia de validación

7.1.1. Preprocesamiento de los datos

Antes de empezar a resolver el problema de clasificación, se ha realizado el tratamiento de datos. Analizando la base de datos, se observa una gran cantidad de datos "missing". Además, también es necesario la normalización de los datos y en algunos casos también habrá que tomar una decisión sobre la codificación de aquellos que no están representados de forma numérica.

Para tratar los datos "missing", se han completado con el valor medio de ese atributo para su clase. También se ha decidido a eliminar ciertos atributos que no aportaban ninguna información. Como por ejemplo atributos que, debido a la precisión de los instrumentos médicos utilizados para las mediciones, tenían, en la totalidad de las instancias, el valor 0.

Otro problema a tratar es que los atributos volumétricos presentan magnitudes muy diferentes entre sí. Para solucionar esto, primero se ha sustituido el valor de los datos volumétricos por el porcentaje respecto al volumen craneal total de cada individuo. Posteriormente, se ha calculado el "Z-score" de los valores, por lo que los atributos se mostrarán como la desviación de ese atributo respecto al valor medio.

7.1.2. Estrategia de validación

Debido al escaso número de casos en nuestra base de datos se ha tenido que desarrollar una estrategia de validación compleja. Esta estrategia permite trabajar con un conjunto de test y validación de un tamaño viable. Así, se ha podido asegurar que los resultados obtenidos se comportaran de forma similar a la hora de realizar la generalización con datos desconocidos.

Inicialmente se ha realizado una estrategia de "Holdout" de un 66 % con la cual se dejaron un 33 % de los datos (120 sujetos) para el conjunto de datos de test y se ha utilizado el 66 % restante (239 casos) para el entrenamiento.

Debido a que se han utilizado técnicas como el "Hyperopt" o técnicas de selección de atributos como "Wrapper" o CFS, se necesita un conjunto de datos de validación para poder evaluar los resultados de estas técnicas sin la necesidad de recurrir al conjunto de prueba. Por ello, dentro del conjunto de entrenamiento, se ha realizado un 10 Fold Cross Validation, que generara 10 conjuntos de datos. Estos conjuntos

han permitido decidir los valores de los hyperparámetros o de la selección de atributos.

Además, a la hora de separar el conjunto de datos, tanto en el "Houldout" como en el "Cross-Validation", se ha tenido en cuenta la distribución de ciertos atributos para que mantengan una distribución similar entre todos los subconjuntos de datos.

Estos atributos han sido la clase, el género y la edad, la cual se ha agrupado en tres grupos: los casos menores de 60 años, los mayores de 70 y los que se encuentran entre 60 y 70 años.

En la Figura 7.2, se puede observar un esquema de la estrategia de validación completa.

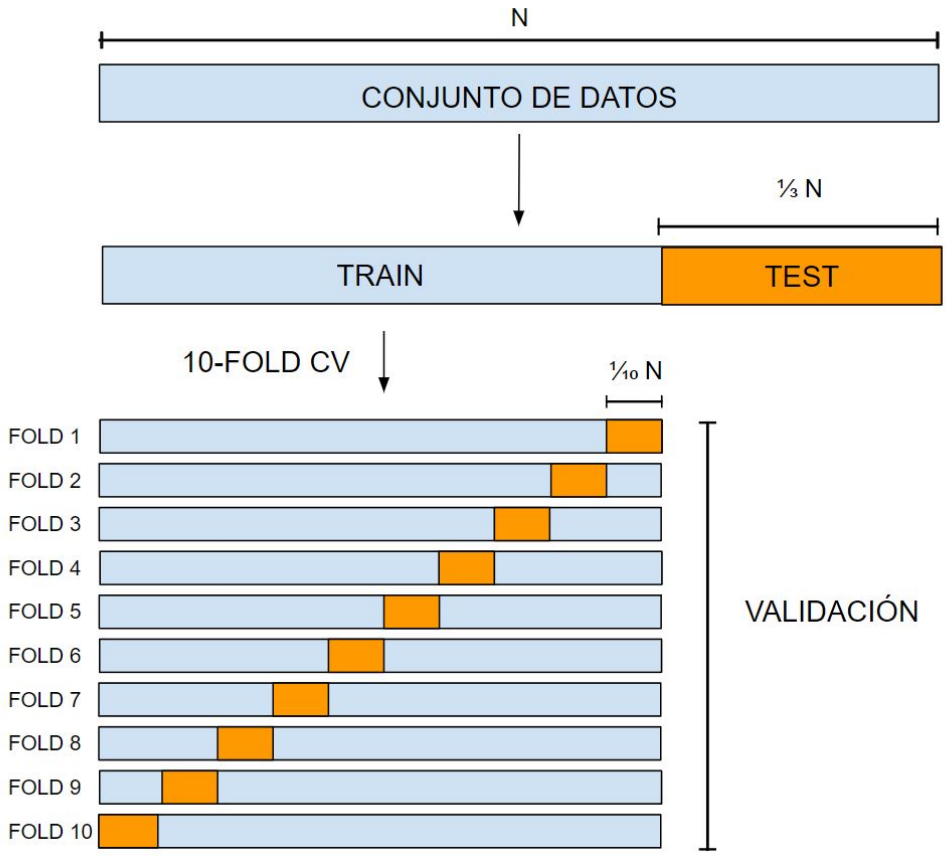

Figura 7.2: Esquema estrategia de validación

Por otro lado la Tabla 7.1 muestra la proporción de cada uno de esos atributos dentro de uno de los Folds pertenecientes al 10-Fold de validación.

	Clase		Género		Edad		
	1	0	1	0	<60	>=60 , <=70	>70
Porcentaje	30 %	70 %	53 %	47 %	37 %	47 %	16 %
Total	36	84	64	56	44	57	19

Tabla 7.1: Tabla distribución de los datos dentro de 1 Fold perteneciente al 10 Fold de validación

7.2. Clasificadores base

Inicialmente se ha realizado la selección de los clasificadores, los cuales serán utilizados durante el transcurso del proyecto. Además, se ha usado el resultado de estos clasificadores como resultados base, dado que no hay resultados de proyectos previos con los que comparar.

Los clasificadores seleccionados son el KNN, C4.5, Random Forest y SVM. La principal razón de esta elección radica en que estos clasificadores son adecuados para entornos con un número reducido de datos, por lo que se adaptan a la base de datos. En este primer apartado se han utilizado los parámetros por defecto de los clasificadores. Por lo tanto, en el KNN se utilizaron 3 vecinos mientras que en el caso del SVM se empleó un kernel lineal.

La Tabla 7.2 presenta cada uno de estos clasificadores en el conjunto de datos de test. Además de los datos pertenecientes a la matriz de confusión, también están representadas las métricas de "Accuracy", F1, "Precision" y "Recall".

	TP	FP	TN	FN	Precision	Recall	Accuracy	F1
3NN	17	13	71	19	0,57	0,47	0,73	0,51
C4.5	10	20	64	26	0,33	0,28	0,62	0,3
RF	12	9	75	24	0,57	0,33	0,72	0,42
SVM	20	17	67	16	0,55	0,54	0,72	0,54

Tabla 7.2: Tabla de los resultados clasificadores base en los datos de test

En la Tabla 7.2 se puede observar como los resultados de estos clasificadores básicos no son lo suficientemente precisos para ser aplicados en la práctica, siendo muy similares a clasificar al azar. Dado que, si la clase mayoritaria es un 70 % de los casos, con clasificar la totalidad de estos como negativos se obtendría un 70 % de "Accuracy".

Debido a esto, la métrica "Accuracy" no es una buena métrica sobre la que trabajar, por lo tanto, se planteó el uso de la métrica F1. Esta métrica presenta una ventaja sobre el "Accuracy" para este problema en cuestión ya que solo tiene en cuenta los valores de TP, FP y FN y permite obtener unos resultados en los que la gran cantidad de casos negativos no afecten demasiado al resultado. Viendo los resultados de esta métrica se observa que 3NN tienen un resultado de 0,51 y SVM tienen un resultado 0,54, mientras que RF tiene 0,42 y C4.5 tiene un 0,3. Con estos resultados se comprueba que el SVM y el 3NN han funcionado mejor que los clasificadores basados en árboles de decisión C4.5 y "Random Forest".

7.3. Selección de atributos

Dado que se trabaja con datos de alta dimensionalidad y que los algoritmos de clasificación utilizados funcionan mejor cuando el número de atributos es bastante menor al de instancias se ha realizado un proceso de selección de atributos. El objetivo ha sido disminuir el número de estos y así reducir el ruido causado por la gran cantidad de atributos, aumentando además la eficiencia de los algoritmos en cuanto a tiempo de ejecución. Para ello se han probado dos métodos distintos: el Wrapper y el CFS.

Para poder dar validez a los resultados obtenidos se ha usado el 10 Fold de validación. Como el Wrapper y CFS funcionan de manera muy distinta, a la hora de la utilización de los 10 Folds se ha utilizado una estrategia distinta para cada caso. A continuación, en cada uno de ellos se explica la estrategia utilizada.

Como solo se va a utilizar un subconjunto de atributos se ha realizado una comparación de la eficiencia de ambos subconjuntos de atributos para elegir el mejor.

7.3.1. Wrapper

Debido a que este proceso se basa en la evaluación de una cantidad muy elevada de subconjuntos, se ha de escoger un clasificador que sea muy eficiente y que computacionalmente sea barato. Es por eso que se ha utilizado un 3NN para realizar la selección.

Para escoger el subconjunto, se ha utilizado un algoritmo de tipo "greedy". Este algoritmo empieza con un subconjunto vacio e irá añadiendo uno a uno los atributos,

escogiendo el que mejor resultado dé al clasificar utilizando la métrica F1.

Como el Wrapper funciona evaluando los subconjuntos de atributos, se tiene que escoger un conjunto de datos sobre el que probarse. Para esto el Wrapper evalúa los subconjuntos realizando un 10 Fold Cross Validation con el conjunto de validación.

Los atributos seleccionados por el Wrapper han sido los siguientes:

- Age.
- Educ.
- APOE-1.
- Sup_Temp_Post_R_WM_VOL.12.
- Accumb_L_CSF_VOL.37.
- Pallidum_L_CSF_VOL.43.
- Pallidum_R_CSF_VOL.44.
- Lat_Temp_Ventr_L_CSF_VOL.47.
- Lat_Temp_Ventr_R_GM_VOL.48.
- Sup_Front_Gyr_R_WM_VOL.60.
- Lingual_Gyr_R_WM_VOL.66.

De acuerdo a lo mostrado, solo se han seleccionado 11 atributos, siendo la mayoría de estos datos volumétricos, pero también están presentes algunos datos no volumétricos como Educ o APOE-1.

7.3.2. CFS

A diferencia del Wrapper, el CFS no necesita de ningún clasificador y utiliza la correlación entre atributos para la selección del subconjunto. Para realizar el CFS se han seleccionado los 10 conjuntos de entrenamiento que forman el 10 Fold Cross Validation, y se ha realizado el CFS con cada uno de los conjuntos. Como resultado se han obtenido 10 conjuntos distintos de atributos. Para seleccionar el conjunto final, se han contado en cuantos de los conjuntos aparece cada atributo, y se han seleccionado aquellos que estén presentes en 5 o más conjuntos.

En la Figura 7.3 se puede observar la cantidad de veces que ha sido elegido cada uno de los atributos.

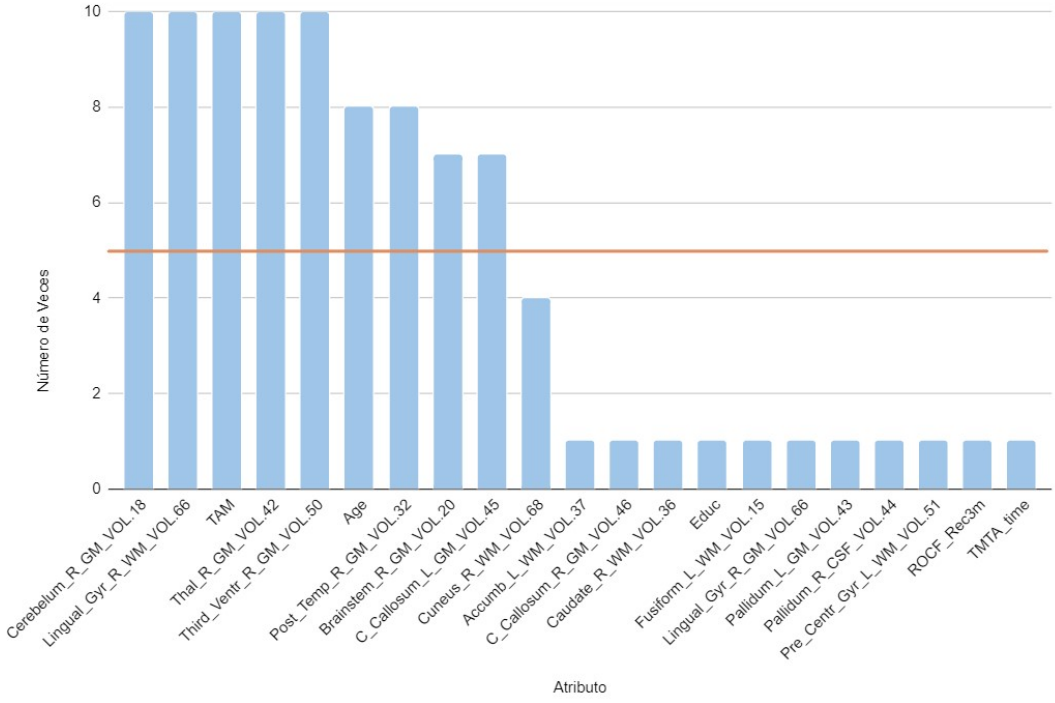

Figura 7.3: Gráfica atributos seleccionados por el CFS en los 10 Folds

Tras dicho proceso se han seleccionado 9 atributos, siendo casi la totalidad de estos datos volumétricos, a excepción la edad y el TAM. Es importante señalar que no comparte ningún atributo con el subconjunto seleccionado por el método Wrapper a excepción de la edad y Lingual_Gyr_R_WM_VOL.66.

7.3.3. Comparación de métodos de selección de atributos

Para realizar la comparación entre ambos métodos de selección de atributos, se ha realizado una clasificación utilizando los 10 Cross-Validation de validación, usando un SVM como clasificador,

En la Tabla 7.3 se presentan los valores de las métricas "Accuracy" y F1.

	Accuracy	F1
WRAPPER	0,72	0,38
CFS	0,77	0,58

Tabla 7.3: Métricas de los conjuntos seleccionados por Wrapper y CFS, probados en el conjunto de validación

También en la Figura 7.4 se pueden ver las matrices de confusión de cada uno de los resultados.

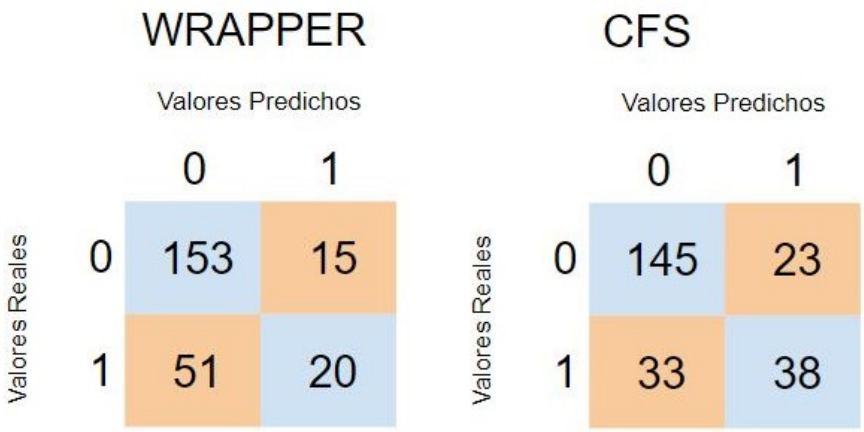

Figura 7.4: Matrices de confusión utilizando los conjuntos seleccionados por Wrapper y CFS, probados en el conjunto de validación

Analizando los resultados obtenidos se observa que el subconjunto seleccionado por el CFS tiene un resultado mucho mejor que el del Wrapper. Esto se puede deber a que el Wrapper se ha desarrollado utilizando un 3NN y ha sido probado con un SVM. Es probable que si la comparación se hubiera hecho con 3NN los resultados serían similares entre WRAPPER Y CFS o incluso mejores en el caso del WRAPPER. Pero debido a que no se va a utilizar un único clasificador y que además el mejor resultado inicial era con el SVM, tiene sentido utilizar este clasificador para escoger el conjunto de atributos. También el CFS, al basarse en la correlación de atributos tendrá un funcionamiento similar para todos los clasificadores por lo que tiene más sentido utilizar el conjunto de atributos seleccionado por este método.

Es interesante comentar, que ciertos atributos que se suponían importantes no han sido seleccionados. Un ejemplo de esto es el APOE, el cual fue señalado desde la fundación CITA como importante. Debido a que desde la fundación CITA se le había dado una especial importancia a dicho atributo, se realizaron análisis adicionales incluyendo entre los atributos seleccionados el APOE, obteniéndose en este caso peores resultados.

7.4. Desbalanceo de clase

Como se ha señalado previamente, el conjunto de datos con el que se ha trabajado está desbalanceado, siendo la clase positiva un 30 % del total de los casos. Trabajar con conjuntos de datos desbalanceados suele empeorar el rendimiento de la clasificación, debido a que los clasificadores tienden a sobre clasificar la clase mayoritaria. Para solucionar esto se han probado dos estrategias distintas.

La primera vía explora el uso de técnicas que se pueden aplicar sobre los clasificadores que estamos utilizando, mientras que la segunda que se basa en la utilización de clasificadores diseñados para tratar con el desbalanceo de clase.

7.4.1. SMOTE

La primera estrategia que se han utilizado para tratar el problema de desbalanceo de clase es el uso de SMOTE. Con esta técnica se han generado instancias sintéticas de la clase positiva con el objetivo de lograr un conjunto de datos balanceado.

De esta manera se han creado 170 instancias sintéticas para ser utilizadas en el proceso de entrenamiento del clasificador.

Como con esta técnica se genera un nuevo conjunto de entrenamiento, se ha decidido que sería positivo comprobar cómo afecta el aplicar el CFS antes o después de realizar el SMOTE. Para ello, se ha calculado otro nuevo CFS habiendo aplicado antes SMOTE y se ha comparado los resultados de ambos a través del 10 Cross Validation de validación, utilizando como siempre un SVM.

En la Figura 7.5 se puede observar las matrices de confusión de ambas opciones(“PRE SMOTE” hace referencia a haber utilizado el CFS antes de aplicar el SMOTE. Mientras que “POST SMOTE” será aquel que primero realiza el SMOTE y luego utiliza CFS).

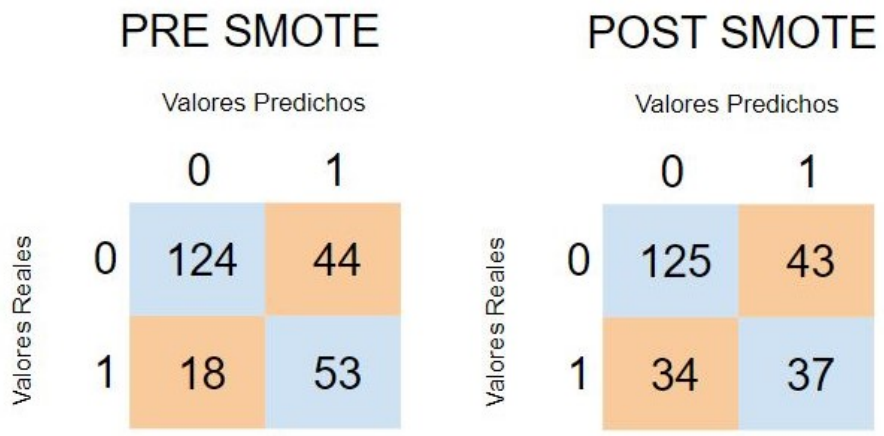

Figura 7.5: Matrices de confusión utilizando los conjuntos seleccionados por el CFS antes o después del realizar el SMOTE, probados en el conjunto de validación

Por otro lado, para poder ver otras métricas, en la Tabla 7.4 se muestran los valores del Accuracy y F1.

	TP	FP	TN	FN	Accuracy	F1
PRE SMOTE	53	44	124	18	0,74	0,63
POST SMOTE	37	43	125	34	0,68	0,49

Tabla 7.4: Métricas de los conjuntos seleccionados por el CFS antes o después del realizar el SMOTE, probados en el conjunto de validación

Como se puede observar, hay una gran diferencia entre el uso del CFS antes y después del SMOTE, y utilizar la selección de atributos antes da mejores resultados. Debido a esto, se ha seleccionado este método para los clasificadores finales.

7.4.2. CTC

Por otro lado, se ha decidido utilizar el algoritmo de clasificación CTC, el cual incorpora dentro del clasificador estrategias para tratar el desbalanceo de clase sin la necesidad de crear nuevos casos sintéticos.

Como este clasificador no genera un nuevo conjunto de datos que pudieran modificar el resultado del CFS, en este clasificador se ha utilizado la selección de atributos que ya se tenía.

Dado que el CTC construye un árbol, también vamos a visualizar el árbol para observar las variables seleccionadas y compararlas con las seleccionadas por el método CFS.

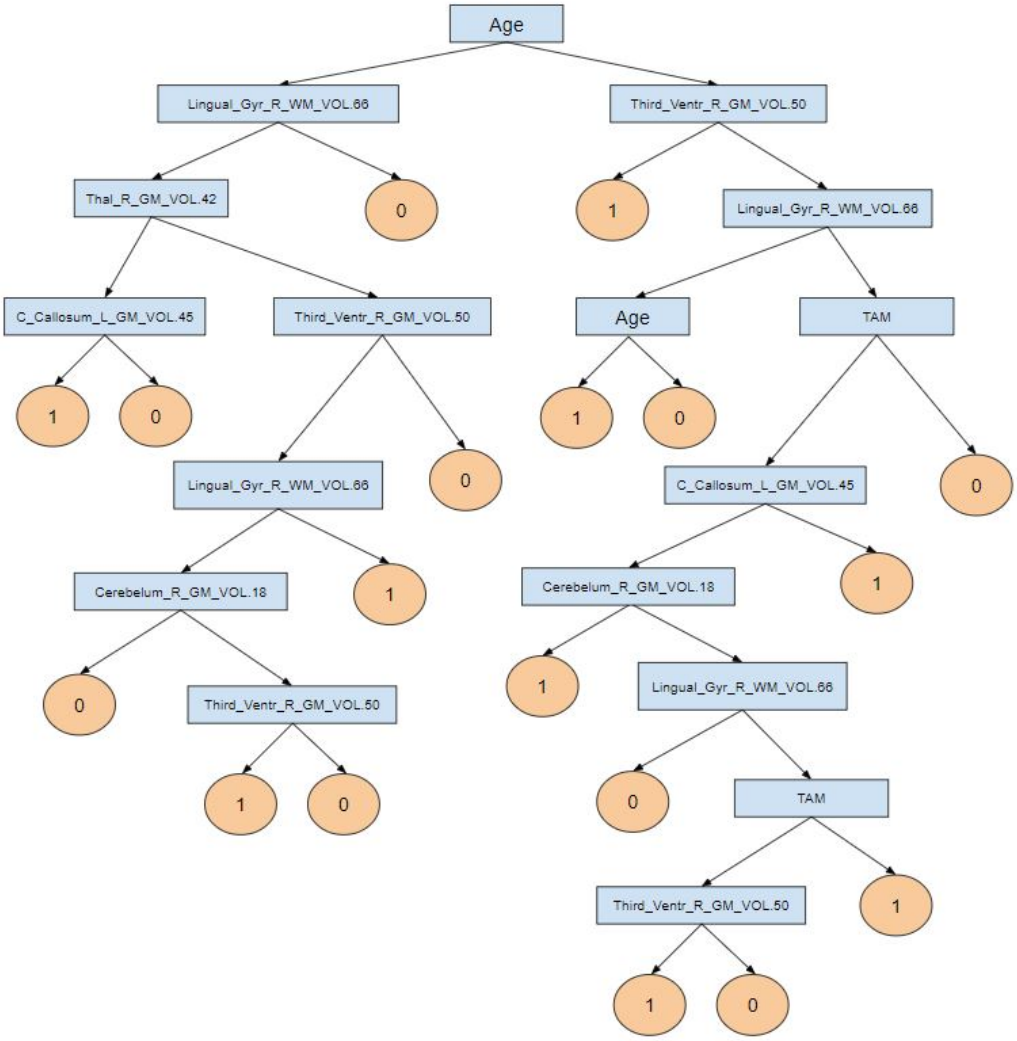

Figura 7.6: Árbol de decisión del CTC

Analizando la Figura 7.6 se puede observar el árbol de decisión que ha construido el algoritmo CTC. Comparando esta figura y la Figura 7.3 del CFS, se observa que la mayoría de atributos seleccionados por el CTC son aquellos que estaban presentes en los 10 folds. Concretamente entre los 5 atributos que estaban en los 10 folds han sido elegidos 13 veces, mientras que entre los 4 atributos que aparecían entre 9 y 5 veces solo han sido seleccionados 4 veces.

7.5. Optimización de Hiperparámetros

Una vez se ha seleccionado el subconjunto de atributos con el que se va a trabajar y que se ha solucionado el problema de desbalanceo de clase con el uso de SMOTE, se ha realizado un proceso de optimización de hiperparámetros. Con este método se pretende conseguir la mejor configuración de los parámetros para cada uno de los clasificadores. El parámetro utilizado por el hyperopt para medir la eficacia de los hiperparámetros seleccionados ha sido la métrica F1, como siempre utilizando el 10 Fold Cross Validation de validación.

A través de esta herramienta se han optimizado los algoritmos KNN, SVM y "Random Forest". El algoritmo C4.5 no ha sido optimizado debido a que la única implementación del clasificador que existe en Python carece de hiperparámetros que configurar. Para cada uno de los clasificadores se ha configurado:

- KNN:

  -Número de vecinos.

- SVM:

  -Tipo de kernel.

  Y dependiendo del tipo de kernel elegido:

  - Coste.
  - Gamma.
  - Grado.

- Random Forest:

  -Número de árboles.

7.6. Discusión de los resultados

Una vez realizado el procedimiento explicado anteriormente se han obtenido los siguientes resultados:

	Algoritmos	TP	FP	TN	FN	Precision	Recall	Accuracy	F1
BASE	3NN	17	13	71	19	0,57	0,47	0,73	0,51
	C4.5	10	20	64	26	0,33	0,28	0,62	0,3
	RF	12	9	75	24	0,57	0,33	0,72	0,42
	SVM	20	17	67	16	0,55	0,54	0,72	0,54
FINALES	5NN	26	24	60	10	0,52	0,72	0,72	0,6
	C4.5	26	25	59	10	0,51	0,72	0,71	0,59
	RF	23	21	63	13	0,52	0,64	0,71	0,57
	SVM	26	22	62	10	0,54	0,72	0,73	0,61
	CTC	25	16	68	11	0,61	0,69	0,77	0,64

Tabla 7.5: Comparación de los resultados de los clasificadores iniciales del proyecto (sección 7.2) y el resultado de los clasificadores tras el desarrollo del proyecto

En la Tabla 7.5, se pueden observar los resultados que se han obtenido sobre el conjunto de prueba final. Los mejores resultados obtenidos son con el clasificador SVM y con el CTC.

Los resultados de F1 que se obtienen con SVM y CTC son cercanos a un 0,1 mejores que el mejor clasificador base con el que se ha empezado. En especial, con el clasificador CTC se puede ver que mejora en prácticamente todos los aspectos de la matriz de confusión respecto al SVM base, que era el que mejor F1 tenía.

Por otro lado, los clasificadores RF y C4.5 han dado un resultado peor, de un F1 de 0,57 en el caso del RF y 0,59 en el caso del clasificador C4.5. Una de las razones por la que la mejora ha sido menor es que en este clasificador no ha sido posible aplicar la hyperoptimización de parámetros. Aunque, comparando con esos mismos clasificadores antes de realizar la selección de atributos y la creación de instancias sintéticas, los resultados obtenidos son de un F1 0,15 mejor. Se puede concluir por lo tanto que el proceso realizado ha servido para mejorar la clasificación sobre la que partíamos al principio del desarrollo.

7.7. Objetivo Alternativo

Dado que los resultados obtenidos en el proyecto no han sido totalmente satisfactorios, se ha decidido intentar disminuir el número de falsos negativos.

Este nuevo objetivo busca lograr un clasificador que, aunque no clasifique bien todos los datos, no genere falsos negativos intentando mantener el número de verdaderos positivos alto. Con esto se lograría que si bien, no se diagnosticara la enfermedad aun dando positivo, se podría por lo menos asegurar que si el test da negativo, el sujeto no padece la enfermedad con un porcentaje de fiabilidad alto. Esto generaría así un método que evitaría a una parte de los casos a someterse a la punción lumbar.

Para lograr este objetivo se ha realizado una optimización de hyperparámetros con el clasificador SVM, pero en vez de intentar mejorar el F1-score, se ha intentado mejorar el "Fβ-score", dándole a β distintos valores. Al modificar el valor de β se le da más importancia a la "Precision" o al "Recall". Además, este objetivo perseguía disminuir el número de FN, pero manteniendo el número de TN. Para ello se ha invertido los valores de la clase durante el proceso de optimización con el objetivo de que el "Fβ-score" tenga en cuenta el TN, FN y FP.

En este caso dándole más importancia a la "Precision". Para esto, se han probado los valores 1, 0,5, 0,25 y 0,05.

También se debe de tener en cuenta que al disminuir β se premia a que el clasificador tienda a clasificar como la clase negativa. Por lo que disminuyendo mucho β se puede llegar al caso de que el clasificador clasifique todo como negativo, por lo tanto, es importante llegar a un punto medio.

En la Figura 7.7 se pueden ver las matrices de confusión para cada uno de los valores de β, mientras que la Tabla 7.6 se muestran lo valores obtenidos para "Recall", "Precision", "Accuracy" y F1.

Figura 7.7: Matriz de Confusión de los distintos clasificadores dependiendo de la β utilizada para la optimización de los hiperparámetros

	TP	FP	TN	FN	Precision	Recall	Accuracy	F1
SVM F1	13	14	70	23	0,482	0,361	0,6917	0,4127
SVM F0,5	26	23	61	10	0,531	0,722	0,7250	0,6118
SVM F0,25	26	22	62	10	0,542	0,722	0,7333	0,6190
SVM F0,1	30	29	55	6	0,509	0,833	0,7083	0,6316

Tabla 7.6: Métricas de los distintos clasificadores dependiendo de la β utilizada para la optimización de los hiperparámetros

Analizando la tabla se pueden ver cómo según el valor de β disminuye, el número de FN también lo hace. El valor más bajo de FN obtenido es 6. Aplicando esta solución se lograría que de 120 casos iniciales, 61 casos no se realizasen la punción, pero se cometerían un 10 % de fallos.

En conclusión, se lograría que la mitad de los sujetos no se realizasen la punción, pero habría un 5 % de los casos que siendo positivos no se les haría la punción lumbar y, por lo tanto, no podrían ser diagnosticados.

CAPÍTULO 8

Conclusiones

En este proyecto se ha intentado diagnosticar la enfermedad del Alzheimer utilizando para ello volúmenes de regiones cerebrales obtenidos mediante resonancias magnéticas. Para esto, se han utilizado distintas estrategias, como algoritmos de selección de atributos, métodos para intentar tratar el desbalanceo de clase, así como distintos algoritmos de clasificación.

Tras haber realizado el proyecto, los resultados obtenidos no son lo suficientemente buenos para poder diagnosticar la enfermedad con precisión. Utilizando como métrica el F1-score, se ha conseguido un valor de 0,64. Este valor, aunque bastante mejor que los clasificadores que se han utilizado como base, sigue siendo muy inferior a lo necesario para poder utilizar este método en la vida real de una manera fiable.

Teniendo en cuenta los resultados obtenidos con los métodos utilizados y los datos de los que se dispone, está claro que todavía hay un largo camino para llegar a conseguir un clasificador capaz de solucionar el problema planteado en este trabajo. Sin embargo, este trabajo puede servir de base para futuros desarrollos en el campo que permitan finalmente dar con una solución satisfactoria.

Por otro lado, se ha intentado desarrollar un método con el cual sea posible reducir el número de personas que se tengan que someter a la punción lumbar. En este apartado se ha conseguido un clasificador que de los 120 casos de prueba que se tienen consigue evitar que 62 se realizan la punción lumbar, aunque indicaría a 10 casos positivos que no se hicieran la punción. Por lo que alrededor de un 13 % de los casos a los que se evitaría la punción lumbar se les diagnosticaría erróneamente.

Este resultado, aunque todavía mejorable, es más prometedor que el obtenido en el objetivo principal.

8.1. Trabajo futuro

Dado que los resultados no han sido del todo satisfactorios, se plantea necesario continuar la investigación por diferentes vías con el objetivo de obtener una solución al problema planteado. En este contexto se plantean dos líneas de trabajo posible.

- Continuando con el objetivo principal del proyecto, todavía hay trabajo que se puede realizar. El resultado obtenido tras utilizar el algoritmo CTC ha sido el mejor. Este clasificador, en el que no se ha profundizado mucho, tiene muchos parámetros que pueden ser configurados. Por lo que sería interesante profundizar en la utilización de este o algoritmos similares para realizar la tarea.
- La otra línea de desarrollo se centraría en intentar reducir el número de gente que necesite realizarse la punción en vez del objetivo principal, sería interesante llevar a cabo un proyecto con este objetivo. Dado que esta idea, al surgir a partir de los resultados del objetivo principal, no se ha llegado a profundizar demasiado y es posible que se pueda conseguir un buen resultado.

apéndice

Atributos de la base de datos

Age	PFEFFER	Digit_Total	ROCF_Rec3m
Sex-numeric	RM_ARWMC_FAZEKAS_NUMBER	Digit_Forw_Total	ROCF_Rec30m
Educ	MMSE	Digit_Forw_Span	SVF_Anim
APOE-1	TAM	Digit_Back_Total	PVF_P
CDR_SB	ROCF_Copy_total	Digit_Back_Span	TMTA_time
CAIDE_Modelo1	ROCF_Copy_Time	T15OBJV1	TMTB_time

Tabla 1: Lista de atributos no volumétricos

TIV	Fusiform_R_WM_VOL.16	Inf_Lat_Pariet_R_GM_VOL.34	Pre_Centr_Gyr_L_CSF_VOL.51
GM_VOL	Fusiform_R_CSF_VOL.16	Inf_Lat_Pariet_R_WM_VOL.34	Pre_Centr_Gyr_R_GM_VOL.52
WM_VOL	Cerebelum_L_GM_VOL.17	Inf_Lat_Pariet_R_CSF_VOL.34	Pre_Centr_Gyr_R_WM_VOL.52
CSF_VOL	Cerebelum_L_WM_VOL.17	Caudate_L_GM_VOL.35	Pre_Centr_Gyr_R_CSF_VOL.52
BPF	Cerebelum_L_CSF_VOL.17	Caudate_L_WM_VOL.35	Straight_Gyr_L_GM_VOL.53
GM_BPF	Cerebelum_R_GM_VOL.18	Caudate_L_CSF_VOL.35	Straight_Gyr_L_WM_VOL.53
WM_BPF	Cerebelum_R_WM_VOL.18	Caudate_R_GM_VOL.36	Straight_Gyr_L_CSF_VOL.53
GM_VOL_hipp_L	Cerebelum_R_CSF_VOL.18	Caudate_R_WM_VOL.36	Straight_Gyr_R_GM_VOL.54
WM_VOL_hipp_L	Brainstem_L_GM_VOL.19	Caudate_R_CSF_VOL.36	Straight_Gyr_R_WM_VOL.54
CSF_VOL_hipp_L	Brainstem_L_WM_VOL.19	Accumb_L_GM_VOL.37	Straight_Gyr_R_CSF_VOL.54
GM_VOL_hipp_R	Brainstem_L_CSF_VOL.19	Accumb_L_WM_VOL.37	Ant_Orbit_Gyr_L_GM_VOL.55
WM_VOL_hipp_R	Brainstem_R_GM_VOL.20	Accumb_L_CSF_VOL.37	Ant_Orbit_Gyr_L_WM_VOL.55
CSF_VOL_hipp_R	Brainstem_R_WM_VOL.20	Accumb_R_GM_VOL.38	Ant_Orbit_Gyr_L_CSF_VOL.55
Amygd_L_GM_VOL.3	Brainstem_R_CSF_VOL.20	Accumb_R_WM_VOL.38	Ant_Orbit_Gyr_R_GM_VOL.56
Amygd_L_WM_VOL.3	Insul_L_GM_VOL.21	Accumb_R_CSF_VOL.38	Ant_Orbit_Gyr_R_WM_VOL.56
Amygd_L_CSF_VOL.3	Insul_L_WM_VOL.21	Putamen_L_GM_VOL.39	Ant_Orbit_Gyr_R_CSF_VOL.56
Amygd_R_GM_VOL.4	Insul_L_CSF_VOL.21	Putamen_L_WM_VOL.39	Inf_Front_Brocca_L_GM_VOL.57
Amygd_R_WM_VOL.4	Insul_R_GM_VOL.22	Putamen_L_CSF_VOL.39	Inf_Front_Brocca_L_WM_VOL.57
Amygd_R_CSF_VOL.4	Insul_R_WM_VOL.22	Putamen_R_GM_VOL.40	Inf_Front_Brocca_L_CSF_VOL.57
Ant_temp_med_L_GM_VOL.5	Insul_R_CSF_VOL.22	Putamen_R_WM_VOL.40	Inf_Front_Brocca_R_GM_VOL.58
Ant_temp_med_L_WM_VOL.5	Lat_Occip_L_GM_VOL.23	Putamen_R_CSF_VOL.40	Inf_Front_Brocca_R_WM_VOL.58
Ant_temp_med_L_CSF_VOL.5	Lat_Occip_L_WM_VOL.23	Thal_L_GM_VOL.41	Inf_Front_Brocca_R_CSF_VOL.58
Ant_temp_med_R_GM_VOL.6	Lat_Occip_L_CSF_VOL.23	Thal_L_WM_VOL.41	Sup_Front_Gyr_L_GM_VOL.59
Ant_temp_med_R_WM_VOL.6	Lat_Occip_R_GM_VOL.24	Thal_L_CSF_VOL.41	Sup_Front_Gyr_L_WM_VOL.59
Ant_temp_med_R_CSF_VOL.6	Lat_Occip_R_WM_VOL.24	Thal_R_GM_VOL.42	Sup_Front_Gyr_L_CSF_VOL.59
Ant_temp_lat_L_GM_VOL.7	Lat_Occip_R_CSF_VOL.24	Thal_R_WM_VOL.42	Sup_Front_Gyr_R_GM_VOL.60
Ant_temp_lat_L_WM_VOL.7	Ant_Cingul_L_GM_VOL.25	Thal_R_CSF_VOL.42	Sup_Front_Gyr_R_WM_VOL.60
Ant_temp_lat_L_CSF_VOL.7	Ant_Cingul_L_WM_VOL.25	Pallidum_L_GM_VOL.43	Sup_Front_Gyr_R_CSF_VOL.60
Ant_temp_lat_R_GM_VOL.8	Ant_Cingul_L_CSF_VOL.25	Pallidum_L_WM_VOL.43	Post_Centr_Gyr_L_GM_VOL.61
Ant_temp_lat_R_WM_VOL.8	Ant_Cingul_R_GM_VOL.26	Pallidum_L_CSF_VOL.43	Post_Centr_Gyr_L_WM_VOL.61
Ant_temp_lat_R_CSF_VOL.8	Ant_Cingul_R_WM_VOL.26	Pallidum_R_GM_VOL.44	Post_Centr_Gyr_L_CSF_VOL.61
Parahipp_L_GM_VOL.9	Ant_Cingul_R_CSF_VOL.26	Pallidum_R_WM_VOL.44	Post_Centr_Gyr_R_GM_VOL.62
Parahipp_L_WM_VOL.9	Post_Cingul_L_GM_VOL.27	Pallidum_R_CSF_VOL.44	Post_Centr_Gyr_R_WM_VOL.62
Parahipp_L_CSF_VOL.9	Post_Cingul_L_WM_VOL.27	C_Callosum_L_GM_VOL.45	Post_Centr_Gyr_R_CSF_VOL.62
Parahipp_R_GM_VOL.10	Post_Cingul_L_CSF_VOL.27	C_Callosum_L_WM_VOL.45	Precuneus_L_GM_VOL.63
Parahipp_R_WM_VOL.10	Post_Cingul_R_GM_VOL.28	C_Callosum_L_CSF_VOL.45	Precuneus_L_WM_VOL.63
Parahipp_R_CSF_VOL.10	Post_Cingul_R_WM_VOL.28	C_Callosum_R_GM_VOL.46	Precuneus_L_CSF_VOL.63
Sup_Temp_Post_L_GM_VOL.11	Post_Cingul_R_CSF_VOL.28	C_Callosum_R_WM_VOL.46	Precuneus_R_GM_VOL.64
Sup_Temp_Post_L_WM_VOL.11	Mid_Front_L_GM_VOL.29	C_Callosum_R_CSF_VOL.46	Precuneus_R_WM_VOL.64
Sup_Temp_Post_L_CSF_VOL.11	Mid_Front_L_WM_VOL.29	Lat_Temp_Ventr_L_GM_VOL.47	Precuneus_R_CSF_VOL.64
Sup_Temp_Post_R_GM_VOL.12	Mid_Front_L_CSF_VOL.29	Lat_Temp_Ventr_L_WM_VOL.47	Lingual_Gyr_L_GM_VOL.65
Sup_Temp_Post_R_WM_VOL.12	Mid_Front_R_GM_VOL.30	Lat_Temp_Ventr_L_CSF_VOL.47	Lingual_Gyr_L_WM_VOL.65
Sup_Temp_Post_R_CSF_VOL.12	Mid_Front_R_WM_VOL.30	Lat_Temp_Ventr_R_GM_VOL.48	Lingual_Gyr_L_CSF_VOL.65
Mid_Inf_Temp_L_GM_VOL.13	Mid_Front_R_CSF_VOL.30	Lat_Temp_Ventr_R_WM_VOL.48	Lingual_Gyr_R_GM_VOL.66
Mid_Inf_Temp_L_WM_VOL.13	Post_Temp_L_GM_VOL.31	Lat_Temp_Ventr_R_CSF_VOL.48	Lingual_Gyr_R_WM_VOL.66
Mid_Inf_Temp_L_CSF_VOL.13	Post_Temp_L_WM_VOL.31	Third_Ventr_L_GM_VOL.49	Lingual_Gyr_R_CSF_VOL.66
Mid_Inf_Temp_R_GM_VOL.14	Post_Temp_L_CSF_VOL.31	Third_Ventr_L_WM_VOL.49	Cuneus_L_GM_VOL.67
Mid_Inf_Temp_R_WM_VOL.14	Post_Temp_R_GM_VOL.32	Third_Ventr_L_CSF_VOL.49	Cuneus_L_WM_VOL.67
Mid_Inf_Temp_R_CSF_VOL.14	Post_Temp_R_WM_VOL.32	Third_Ventr_R_GM_VOL.50	Cuneus_L_CSF_VOL.67
Fusiform_L_GM_VOL.15	Post_Temp_R_CSF_VOL.32	Third_Ventr_R_WM_VOL.50	Cuneus_R_GM_VOL.68
Fusiform_L_WM_VOL.15	Inf_Lat_Pariet_L_GM_VOL.33	Third_Ventr_R_CSF_VOL.50	Cuneus_R_WM_VOL.68
Fusiform_L_CSF_VOL.15	Inf_Lat_Pariet_L_WM_VOL.33	Pre_Centr_Gyr_L_GM_VOL.51	Cuneus_R_CSF_VOL.68
Fusiform_R_GM_VOL.16	Inf_Lat_Pariet_L_CSF_VOL.33	Pre_Centr_Gyr_L_WM_VOL.51	

Tabla 2: Lista de atributos volumétricos

Bibliografía

[1] Cleusa P Ferri, Martin Prince, Carol Brayne, Henry Brodaty, Laura Fratiglioni, Mary Ganguli, Kathleen Hall, Kazuo Hasegawa, Hugh Hendrie, Yueqin Huang, and et al. Global prevalence of dementia: A delphi consensus study. *The Lancet*, 366(9503):2112–2117, 2005. Ver página 1.

[2] Grigorios Tsoumakas and Ioannis Katakis. 2020 alzheimer's disease facts and figures. *Alzheimer's , Dementia*, 16(3):391–460, 2020. Ver página 1.

[3] Kumar B. Rajan, Jennifer Weuve, Lisa L. Barnes, Elizabeth A. McAninch, Robert S. Wilson, and Denis A. Evans. Population estimate of people with clinical alzheimer's disease and mild cognitive impairment in the united states (2020–2060). *Alzheimer's & Dementia*, 17(12):1966–1975, 2021. Ver página 1.

[4] Adela Emilia Gómez Ayala. Factores de riesgo en la enfermedad de alzheimer. *Farmacia Profesional*, 21(2):62–67, Feb 2007. Ver página 9.

[5] Sanidad Ministerio De. *Madrid: Ministerio de Sanidad.* Consumo y Bienestar Social, 2019. Ver página 9.

[6] Bryan D. James, Robert S. Wilson, Lisa L. Barnes, and David A. Bennett. Late-life social activity and cognitive decline in old age. *Journal of the International Neuropsychological Society*, 17(6):998–1005, 2011. Ver página 10.

[7] MICHAEL J. VALENZUELA and PERMINDER SACHDEV. Brain reserve and dementia: A systematic review. *Psychological Medicine*, 36(4):441–454, 2005. Ver página 10.

[8] Zeinab Breijyeh and Rafik Karaman. Comprehensive review on alzheimer's disease: Causes and treatment. *Molecules*, 25(24):5789, 2020. Ver página 10.

[9] Mark A. Mintun, Albert C. Lo, Cynthia Duggan Evans, Alette M. Wessels, Paul A. Ardayfio, Scott W. Andersen, Sergey Shcherbinin, JonDavid Sparks, John R. Sims, Miroslaw Brys, Liana G. Apostolova, Stephen P. Salloway, and Daniel M. Skovronsky. Donanemab in early alzheimer's disease. *New England Journal of Medicine*, 384(18):1691–1704, 2021. PMID: 33720637. Ver página 11.

[10] Christopher H. van Dyck, Chad J. Swanson, Paul Aisen, Randall J. Bateman, Christopher Chen, Michelle Gee, Michio Kanekiyo, David Li, Larisa Reyderman, Sharon Cohen, Lutz Froelich, Sadao Katayama, Marwan Sabbagh, Bruno Vellas, David Watson,

Shobha Dhadda, Michael Irizarry, Lynn D. Kramer, and Takeshi Iwatsubo. Lecanemab in early alzheimer's disease. *New England Journal of Medicine*, 388(1):9–21, 2023. PMID: 36449413. Ver página 11.

[11] A.P. Porsteinsson, R.S. Isaacson, S. Knox, M.N. Sabbagh, and I. Rubino. Diagnosis of early alzheimer's disease: Clinical practice in 2021. *The Journal Of Prevention of Alzheimer's Disease*, page 1–16, 2021. Ver página 11.

[12] Marianna Inglese, Neva Patel, Kristofer Linton-Reid, Flavia Loreto, Zarni Win, Richard J. Perry, Christopher Carswell, Matthew Grech-Sollars, William R. Crum, Haonan Lu, and et al. A predictive model using the mesoscopic architecture of the living brain to detect alzheimer's disease. *Communications Medicine*, 2(1), 2022. Ver página 12.

[13] Eugene Lin, Chieh-Hsin Lin, and Hsien-Yuan Lane. Deep learning with neuroimaging and genomics in alzheimer's disease. *International Journal of Molecular Sciences*, 22(15):7911, 2021. Ver página 12.

[14] Carol Y Cheung, Vincent Mok, Paul J Foster, Emanuele Trucco, Christopher Chen, and Tien Yin Wong. Retinal imaging in alzheimer's disease. *Journal of Neurology, Neurosurgery & Psychiatry*, 92(9):983–994, 2021. Ver página 12.

[15] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 9 1995. Ver página 13.

[16] Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification. *Database Technologies*, page 309–319, 2009. Ver página 13.

[17] Vladimir Naumovic Vapnik. *Estimation of dependences based on empirical data*. Springer, 2006. Ver página 14.

[18] Shan Suthaharan. *Machine learning models and algorithms for Big Data Classification: Thinking with examples for effective learning*. Springer, 2016. Ver página 14.

[19] Jingwen Sun, Weixing Du, and Niancai Shi. A survey of knn algorithm. *Information Engineering and Applied Computing*, 1(1), 2018. Ver página 15.

[20] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967. Ver página 15.

[21] Geoffrey I. Webb, Eamonn Keogh, Risto Miikkulainen, Risto Miikkulainen, and Michele Sebag. Naïve bayes. *Encyclopedia of Machine Learning*, page 713–714, 2011. Ver página 15.

[22] D. Barber. *Bayesian reasoning and Machine Learning*. Cambridge University Press, 2018. Ver página 16.

[23] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993. Ver página 16.

[24] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001. Ver página 17.

[25] Jesús M. Pérez, Javier Muguerza, Olatz Arbelaitz, and Ibai Gurrutxaga. Consolidated tree construction algorithm: Structurally steady trees. In *International Conference on Enterprise Information Systems*, 2004. Ver página 17.

[26] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002. Ver página 18.

[27] Mark A. Hall. Correlation-based feature selection for discrete and numeric class machine learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, page 359–366, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. Ver página 19.

[28] Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1):273–324, 1997. Relevance. Ver página 19.

[29] Mark A. Hall and Lloyd A. Smith. Feature selection for machine learning: Comparing a correlation-based filter approach to the wrapper. In *The Florida AI Research Society*, 1999. Ver página 19.

[30] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera. A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(4):463–484, 2012. Ver página 21.