UNIVERSIDAD DE MÁLAGA

E.T.S. INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE MÁLAGA

Graduado en Ingeniería de la Salud

Aprendizaje Computacional aplicado al diagnóstico de autismo mediante conectomas funcionales

Computational Learning applied to autism diagnosis using functional connectomes

Realizado por
Clara Jiménez Valverde

Tutorizado por
Rafael Marcos Luque Baena
Domingo López Rodríguez

Departamento
Lenguajes y Ciencias de la Computación

MÁLAGA, junio de 2021

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

GRADUADO EN INGENIERÍA DE LA SALUD

Aprendizaje Computacional aplicado al diagnóstico de autismo mediante conectomas funcionales

Computational Learning applied to autism diagnosis using functional connectomes

Realizado por
Clara Jiménez Valverde

Tutorizado por

Rafael Marcos Luque Baena
Domingo López Rodríguez

Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA

MÁLAGA, JUNIO DE 2021

Fecha defensa: julio de 2021

Abstract

This project's goal is to aid in the diagnosis of Autism Spectrum Disorder (ASD), which is a challenging task due to its complexity. Between 1 % and 2 % of children are estimated to be in the spectrum, and even though there are multiple diagnosis methods directed at the visual characteristics of autism such as family and social settings, common behaviour and cognitive intelligence, its manifestations vary between sexes and individuals, sometimes making these tests an unreliable method. ASD investigation has shown that the disorder is caused majorly by the differences in brain connections rather than on the brain's anatomy. These connections are shown in patients' functional connectomes.

This project employs connectomes from individuals with and without ASD to create classification models specific to men and women, and also generic ones. They are computed applying a tool designed by the Brain Image and Analysis Center from Duke University to the functional magnetic resonance images collected in the ABIDE initiative. Five different classification algorithms have been used and the ones with the better precission values have been integrated in an application. It allows the user to enter connectome or fMRI data from a patient and then receive a prediction from each of the five models, getting help with diagnosis.

Keywords: Bioinformatics, Computational Learning, ASD, Classification Algorithms

Resumen

Este proyecto ha sido creado con el objetivo de ayudar en el diagnóstico del Trastorno del Espectro Autista (TEA), tarea dificultada por la complejidad del mismo. Se estima que entre el 1 % y el 2 % de los niños lo padecen, y aunque existen numerosos métodos de diagnóstico enfocados en las características visibles del autismo (ámbito social y familiar, comportamientos comunes, inteligencia cognitiva), sus manifestaciones difieren entre sexos e individuos, por lo que los resultados no siempre son fiables. La investigación del TEA ha revelado que el trastorno se debe más a las diferencias en las conexiones del cerebro que en su anatomía, y estas conexiones se muestran en los conectomas funcionales de los pacientes.

Este trabajo usa conectomas de individuos con y sin TEA para crear modelos de clasificación específicos para mujeres y hombres, además de otros genéricos. Estos se extraen aplicando una herramienta creada en el Centro de Imagen y Análisis de la Universidad de Duke a las resonancias magnéticas funcionales recopiladas en el proyecto ABIDE. Se han empleado cinco algoritmos de clasificación diferentes, y los más precisos de cada uno están integrados en una aplicación que permite al usuario introducir los datos de conectoma o fMRI de un paciente para que se realice una predicción con los cinco modelos seleccionados, sirviendo esta de ayuda al diagnóstico.

Palabras clave: Bioinformática, Aprendizaje Computacional, TEA, Algoritmos de clasificación.

A mis tutores, Rafael Luque y Domingo López, por ayudarme y animarme durante este trabajo y enseñarme tanto en el proceso.

A mi familia, en especial a mis padres, por haberme repetido miles de veces que lo más importante es ser feliz.

A mis amigas, las que he conocido en esta etapa y aquellas que han estado ahí toda mi vida, en las que siempre me podré apoyar pase lo que pase.

Y a mis cuatro compañeras de piso, que me han acompañado en mis altibajos durante los tres años que hemos compartido y me han ayudado a crecer enormemente como persona.

Índice

1 Introducción

1.1. Motivación

Los **Trastornos del Espectro Autista** (TEA) son una serie de problemas que afectan a las habilidades emocionales, sociales y de comunicación de aquellos que los padecen. Sus síntomas suelen aparecer en la infancia, pero siempre ha habido **problemas para diagnosticarlo**. Hoy en día, el diagnóstico se ha mejorado mucho, pasándose a creer que su **prevalencia de entre el 1 % y el 2 %** [1], [2] en niños y niñas de 8 años, frente al 0,04 % que se creyó en un principio, hacia los años 50 [3].

Pero esto no quita que haya problemas a la hora de identificar a las personas con autismo. Al ser este un trastorno comprendido dentro de un **espectro**, los identificadores y comportamientos van a variar de un paciente a otro, lo que significa que un médico debe **tener mucha experiencia** para diagnosticarlo correctamente. Se dan casos en los que hay sospecha de que el paciente pertenece al espectro autista, pero no se ahonda en esta suposición, ya sea por falta de experiencia o de ganas [4]. Estos retrasos en el diagnóstico pueden afectar mucho la vida del paciente [4].

Muchos de los **métodos usados hoy en día** para diagnosticar TEA se basan en listas referentes al comportamiento de los pacientes. Se han creado métodos de *Machine Learning* basados en estas características [5], [6], pero algunas de ellas son subjetivas o no se dan en pacientes con un grado de autismo menor, por lo que **no siempre son útiles**.

Pero estos no son los únicos métodos disponibles para diagnosticar el trastorno. Los **conectomas** (mapas de las conexiones entre las neuronas del cerebro) en pacientes con TEA **son diferentes a los pertenecientes a pacientes neurotípicos** [7], [8], [9], por lo que podría servir como método de diagnóstico. Este método no está tan explorado como otros, por lo que proporciona **nuevas posibilidades**.

Otro de los problemas a la hora de conseguir un diagnóstico surge con las **diferentes manifestaciones que tiene el autismo en niñas y niños**. Dado que los modelos de diagnóstico se han creado basándose en los comportamientos que exhibían niños con autismo, estos contemplan mayormente la manera en la que se manifiesta el autismo en el sexo masculino, que tiene algunas **diferencias esenciales** con la manera en la que se manifiesta en el sexo femenino (como es el caso de las habilidades de comunicación) [10]. Estas diferencias han causado que históricamente a las **mujeres se les diagnostique más tarde que a los hombres** [10], [11], dificultando su vida y atrasando su acceso a profesionales o técnicas que puedan ayudarles a vivir y adaptarse a la sociedad.

1.2. Objetivos

El fin de este trabajo es desarrollar un **modelo de aprendizaje computacional** que sea capaz de usar el **conectoma funcional** de un paciente para poder identificar si este padece o no un Trastorno del Espectro Autista. Para poder conseguir esto, habrá que **obtener los conectomas a partir de imágenes MRI** tanto de pacientes que tengan autismo como de aquellos que no, para el grupo control.

Posteriormente, deberemos seleccionar los **modelos de aprendizaje computacional** a usar, y entrenarlos para que puedan desarrollar el objetivo buscado. Además, se aplicará un **proceso de validación y evaluación** de los modelos seleccionados utilizando los conjuntos de datos de entrenamiento, validación y test. Una vez tengamos un modelo capaz de determinar si un paciente pertenece al espectro autista o no, pasaremos a integrar el modelo obtenido en una herramienta que permita el uso de este modelo por parte de un usuario, que podrá usarlo con datos nuevos para obtener un diagnóstico orientativo.

1.3. Estructura del documento

Esta memoria contiene trece capítulos y dos apéndices que dividen su contenido en las siguientes secciones:

- **Introducción** (Capítulo 1): Este primer capítulo presenta el tema del proyecto, con un breve contexto teórico y los objetivos del mismo, seguidos por dos apartados que describen la estructura del documento y las tecnologías usadas en el desarrollo del proyecto.

- **Conceptos teóricos** (Capítulo 2): El segundo capítulo ahonda en los tres conceptos teóricos necesarios para comprender el proyecto: el Trastorno del Espectro Autista, los MRIs (Imágenes por Resonancia Magnética funcional), y los conectomas.
- **Cálculo de los conectomas** (Capítulo 3): Este capítulo describe el proceso necesitado para la adquisición de los fMRIs usados, y su transformación en conectomas, con descripciones de los pasos que sigue esta.
- **Modelos de predicción** (Capítulo 4): El cuarto capítulo describe la metodología general seguida para crear los modelos de predicción, y los algoritmos de clasificación y métodos de ajuste de hiperparámetros empleados para ello.
- Los capítulos del 5 al 9, **K vecinos más cercanos**, **Árboles de decisión**, ***Random forests***, **Máquinas de vectores de soporte** y **Perceptrones multicapa** describen las estrategias seguidas para obtener los modelos de clasificación para cada uno de estos métodos, indicando los parámetros explorados y sus resultados.
- **Modelos elegidos** (Capítulo 10): El décimo capítulo recopila los mejores resultados de precisión para cada tipo de algoritmo y conjunto de datos, pues serán los usados en la interfaz.
- **Interfaz** (Capítulo 11): En este capítulo se describe la creación de la interfaz y su funcionamiento.
- **Conclusiones y líneas futuras** (capítulo 12): Este último capítulo relata las conclusiones extraídas del proyecto y las posibles líneas futuras por las que se podría conducir.
- **Entidades participantes en ABIDE** (Apéndice A): Contiene una tabla que recopila las entidades participantes en la iniciativa ABIDE y la contribución de cada una de ellas.
- **Manual de instalación** (Apéndice B): Contiene la información necesaria para poder lanzar la aplicación en un ordenador.

1.4. Tecnologías usadas

- **ABIDE**. El *Autism Brain Imaging Data Exchange* es una iniciativa con dos ediciones publicadas en 2012 y 2016. Surge con el objetivo de crear una base de datos de gran ta-

maño, necesaria para la investigación del TEA, pues este es altamente complejo. Para conseguirlo, se recopilaron MRIs funcionales y estructurales a nivel mundial de pacientes con y sin TEA, consiguiendo la colaboración de 26 centros (Información disponible en el Apéndice A). Estos datos están disponibles para aquel que los necesite, pues ABIDE cree en el principio de la Ciencia Abierta.

- **NITRC**. El *NeuroImaging Tools & Resources Collaboratory* es una plataforma web de acceso a datos y software de neuroinformática en la que están almacenados los datos del proyecto ABIDE.

- **LONI Image & Data Archive**. Se trata de otra plataforma de acceso a datos neurocientíficos, que contiene los datos de la primera edición de ABIDE.

- **Python** (versión 3). Python es un lenguaje de programación interpretado fácil de usar gracias a su simplicidad, que además tiene muchas funcionalidades por los muchos paquetes que integra. Con él se crearán tanto los modelos de predicción como la interfaz de este proyecto.

- ***Python/FSL Resting State Pipeline***. Como su nombre indica, es una *pipeline* que emplea Python y su paquete FSL (descrito más adelante) para procesar imágenes de resonancia magnética funcionales y devolver su conectoma. Fue creada por el Centro de Imágenes y Análisis Cerebral de la Universidad de Duke, y se debe llamar desde consola con una serie de argumentos que dependen del archivo usado y la acción necesitada.

- **FSL**. FSL es una librería de herramientas de análisis de imágenes cerebrales que está disponible para sistemas Unix. La *pipeline* mencionada en el punto anterior emplea algunas de sus funciones.

- **BXH/XCEDE Tools**. Son una serie de herramientas disponibles en NITRC para el procesado y análisis de imágenes, usadas mucho para neuroimágenes. La herramienta de Duke emplea algunas de sus utilidades.

- ***scikit-learn***. Es un paquete disponible para Python que incluye gran cantidad de herramientas de aprendizaje computacional, y en concreto para la creación de modelos de clasificación que necesitaremos en este proyecto.

- **Flask**. Flask es otro paquete de Python creado para facilitar la creación de aplicaciones web.
- **Bootstrap**. Esta herramienta ayuda al diseño web teniendo disponibles plantillas genéricas para todo tipo de componentes web que se pueden aplicar en HTML.
- **Google Colab**. Se trata de una herramienta online de Google que permite ejecutar programas Python usando servidores y procesadores de Google. Se empleará para crear los modelos de clasificación.

Las instalaciones requeridas para poder usar todos los componentes del proyecto se incluyen en el archivo *README.txt* entregado junto a esta memoria.

2

Conceptos teóricos

Como ocurre con cualquier trabajo, para entenderlo es necesario comprender su contexto teórico. En este primer capítulo se incluye información sobre los conceptos más importantes que serán manejados durante el texto, con el objetivo de facilitar su comprensión.

2.1. Trastorno del espectro autista

El **trastorno del espectro autista**, o TEA, es hoy en día uno de los trastornos más famosos y diagnosticados en el mundo [1]. Desde que se describió por primera vez hasta hoy, las características que lo definen han sido revisadas y modificadas numerosas veces, al igual que los diferentes enfoques de la sociedad sobre él. Algo que prevalece son las **diferencias en el diagnóstico en mujeres y hombres**, como ya se comentó en la introducción. En esta sección se tratarán este y otros temas, empezando por la definición.

2.1.1. Definición

Aunque en muchos casos su primera descripción se atribuya a **Leo Kanner** en 1943 [12] o a **Hans Asperger** en 1944 [13], fue la doctora **Grunya Efimovna Sukhareva** la primera en detallar las características del TEA en 1926 [14], [15]. La autora **Sula Wolf** indica, en su traducción del artículo de Sukhareva, que el trabajo de Asperger era muy parecido a este [16].

Esta primera descripción relataba la actitud de los niños con TEA como solitaria, impulsiva, con problemas de adaptación, altibajos emocionales, fuertes fijaciones en intereses específicos y sensibilidad a olores y ruidos, entre otros detalles[15].

Por su parte, el estudio del Dr. Kanner concluía que los niños observados habían nacido incapaces de entablar relaciones de afecto, buscando estar solos aunque aceptando progresivamente la presencia de otros, ganando también capacidades de comunicación con el tiempo [12]. De nuevo, se mencionan la sensibilidad a olores y ruidos y las fijaciones, además de berrinches

repentinos, posiblemente equivalentes a los altibajos emocionales descritos por Sukhareva. El Dr. Kanner, al igual que otros después de él, como Bruno Bettelheim, sugirió en un principio que el autismo podía deberse a la falta de cariño de los padres, y más en concreto de las madres, aunque más tarde cambió su opinión [17].

El Dr. Asperger, por su parte, describió un comportamiento muy similar al de la doctora, pero más tarde su nombre se ha asociado con lo conocido como *Autismo de capacidad intelectual alta* o *de alto funcionamiento*, referido a individuos cuyo TEA es menos aparente y por tanto influye menos en su vida. Este término, junto a *Trastorno de Asperger*, dejó de ser un diagnóstico oficial en 2013, pues se encapsula dentro del espectro autista.

Desde el siglo pasado, el **Trastorno del Espectro Autista** ha tomado muchas terminologías (empezó considerándose como un síntoma de esquizofrenia), y, lo que es más importante, han surgido hipótesis de todo tipo acerca de su procedencia. Desde la ya comentada frialdad maternal, a la genética, pasando por causas neurológicas. Incluso se le ha echado en repetidas ocasiones (notablemente los autores Rimland y Wakefield) la culpa a las vacunas, causando movimientos antivacunas que perviven hasta hoy, aunque esta idea haya sido probada falsa por múltiples estudios.. [17].

Hoy en día, el TEA aúna todos los tipos de autismo, y se sabe que se debe más a diferencias en la conectividad neuronal que a la anatomía del sistema nervioso. Asimismo, también se han encontrado algunas alteraciones neuroquímicas y razones genéticas (es muy heredable, por encima del 80 %) con alrededor de 1000 genes relacionados [18].

2.1.2. Métodos de diagnóstico

El **Manual Diagnóstico y Estadístico de los Trastornos Mentales** (DSM-5 en inglés) [19] y la **Clasificación Internacional de Enfermedades** (CIE-11) [20] tienen definiciones algo diferentes respecto a los criterios diagnóticos del TEA, pero comparten los siguientes puntos:

- Dificultades continuas en comunicación e interacción social.
- Patrones de comportamiento e intereses fijos y repetitivos.
- Estas características causan problemas a nivel social, laboral, familiar, educativo o similares.

Todos estos rasgos pueden ser más o menos acentuados dependiendo del paciente, ya que este trastorno es un espectro. Se buscan en cuestionarios como el ADOS-G, *Autism Diagnostic Observation Schedule*, que evalúa comportamientos sociales, comunicativos, de juegos, etc. de pacientes con sospecha de TEA. Otro ejemplo es ADI-R, *Autism Diagnostic Interview*, una entrevista para padres de niños con sospecha de pertenecer al espectro autista.

2.1.3. Diferencias en mujeres y hombres

El TEA se ha considerado desde siempre un *"trastorno de hombres"*, y aunque esto podría tener sentido dependiendo de sus causas, también se sabe que las proporciones han sido exageradas, creándose una impresión pública de que las mujeres no pueden tener autismo. Dicha creencia es solo uno de los muchos factores que han contribuido a un diagnóstico tardío, erróneo, o directamente inexistente en mujeres en el espectro.

En un artículo publicado en octubre de 2020 [10], la **Doctora Georgia Lockwood Estrin**, junto a otras autoras, realiza una revisión sitemática de veinte artículos, de donde extraen conclusiones sobre este tema. En este apartado, las resumiré con el objetivo de dar un contexto acerca de la problemática.

Respecto a los **comportamientos típicos** de las personas con autismo, el artículo explica que las niñas diagnosticadas comienzan a mostrar características sociales propias del espectro mucho después que los niños, aunque estos últimos parezcan tener mejores habilidades de comunicación. Además, ellas también muestran por lo general menor inteligencia cognitiva, y aquellas que tienen mayor capacidad son diagnosticadas más tarde, aunque lo contrario no se cumpla. Esto se asocia con un tema recurrente en la publicación: *para que una mujer consiga su diagnóstico parecen ser necesarios muchos más signos que para el diagnóstico de un hombre.*

En sus **relaciones**, las niñas muestran las mismas habilidades que los niños neurotípicos. No es hasta que estas relaciones se observan de cerca que se aprecia la distancia entre las niñas con TEA y los que los rodean, mientras que en niños esta separación es mucho más evidente. Estos detalles, y otros relacionados con la dificultad de detectar las características del autismo en niñas vienen de la mayor capacidad de estas de *enmascararlas*, sea o no conscientemente.

Derivado del problema de que se perciba al trastorno como uno que aparece mayormente en niños derivan muchos otros, desde que los padres lo piensen y por tanto no se planteen la posibilidad de que sus hijas tengan TEA a que los profesionales médicos se decanten por otros

diagnósticos a menos que el trastorno sea agudo, como **hiperactividad** o **déficit de atención**.

Todos estos obstáculos en el diagnóstico forman un **ciclo**, pues al diagnosticar predominantemente a hombres, se alimenta la idea de que el TEA se da casi exclusivamente en ellos; y al tener las mujeres diagnosticadas muchas más características del espectro, los estudios efectuados sobre ellas no representan a la totalidad de mujeres con autismo, sino a aquellas con autismo más agudo.

Por ello muchos autores y profesionales sugieren la creación de **criterios de diagnóstico exclusivos para mujeres**, pues la manifestación del autismo en ellas varía mucho de los hombres, causando una menor tasa de diagnóstico que afecta a sus vidas gravemente.

2.2. MRIs

Las **Imágenes por Resonancia Magnética** (MRIs por sus siglas en inglés) son esenciales para este trabajo, pues calcularemos los conectomas a partir de ellas. Su tecnología crea **imágenes 3D altamente detalladas**, usadas predominantemente en medicina para diagnóstico, detección y seguimiento de tratamiento [21]. Como su nombre indica, funcionan aplicando **campos magnéticos** muy potentes que afectan a los átomos con número impar de protones (principalmente el átomo de hidrógeno), alineándolos de manera paralela a estos. Además, los protones realizan un movimiento denominado de precesión cuya frecuencia depende del campo magnético. Si a estos núcleos les aplicamos una **onda de radiofrecuencia** con frecuencia idéntica a la de precesión, entran en **resonancia**, absorbiendo la energía de la onda. Cuando termina la absorción, la energía se devuelve y los protones se vuelven a alinear con el campo magnético.

Para obtener una imagen donde se pueda discernir entre tejidos, se envían **pulsos de diferente intensidad y duración** para modificar el tiempo que tarda en volver la energía. Se aplican varios gradientes magnéticos para poder obtener información de todos los planos.

Las imágenes pueden medir el tiempo de relajación longitudinal (imágenes potenciadas en T1), es decir, el necesario para que se produzca el regreso de los núcleos atómicos a su estado energético inicial; o el tiempo de relajación transversal (imágenes potenciadas en T2), el que tardan en desfasarse los protones que están en movimiento de precesión síncrono. Antes se medía también la cantidad de energía emitida por un plano (densidad protónica) pero cada vez se usan menos. Las imágenes usadas en este estudio estarán en T1.

2.2.1. fMRIs

Las imágenes concretas que usaremos serán **Imágenes por Resonancia Magnética funcional** (**Figura 1**) tomadas en estado de reposo. El objetivo de las fMRIs es **mapear la actividad cerebral** partiendo del hecho de que esta conlleva un aumento de metabolismo en el lugar donde se de, y por tanto mayor necesidad de oxígeno, que deberá ser extraído de la sangre. Una consecuencia directa de esta extracción es un aumento del flujo sanguíneo a las neuronas activas, con el que se reduce la proporción entre hemoglobinas oxi y deoxi, que se mide y es usada para tomar la imagen [22].

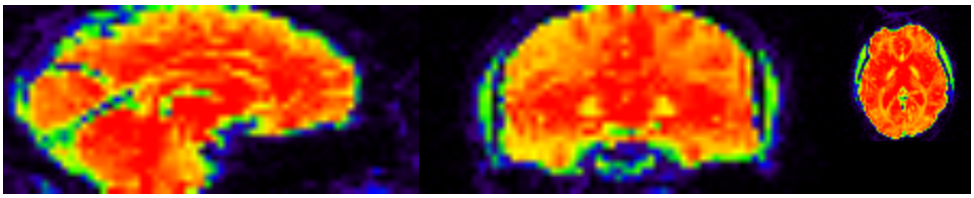

Figura 1: Resonancia magnética funcional de un paciente con TEA

En **autismo**, estos análisis se han usado en estudios dirigidos a las diferentes características del trastorno. Estos han encontrado **diferencias en la activación** de varias áreas del cerebro en individuos con y sin TEA en situaciones de **reconocimiento facial, procesamiento del lenguaje y funciones ejecutivas** (habilidades cognitivas relacionadas con la realización de tareas) [23], [24], algunas de las características principales del TEA.

Las fMRIs se pueden considerar como una serie temporal de volúmenes, y serán precisamente estas series las que usemos en el cálculo de conectomas para nuestro estudio.

2.3. Conectomas

El concepto de conectoma lo acuñó por primera vez el **Doctor Olaf Sporns** en un artículo publicado en 2005 [25] donde discutía diversas estrategias que existían en el momento de crear una **red estructural de los elementos y conexiones del cerebro humano**. Estas tenían el objetivo de aumentar la comprensión de cómo se relacionaban los estados funcionales del cerebro con su anatomía.

El artículo describió el conectoma como la **unión de los elementos del cerebro y sus conexiones**, ilustradas por una **matriz donde las filas serían las fuentes de las señales**

y las columnas sus objetivos, que se rellenaría con ceros y unos según ausencia o presencia de conexión. Uno de los problemas iniciales al diseñar el conectoma era el nivel al que se debían considerar los elementos. Existía la posibilidad de considerar los más pequeños del cerebro (las **neuronas**) o los que abarcaban mayor superficie (las **regiones cerebrales**), pasando por usar **grupos neuronales**. Lo más fácil, indica el autor, es emplear los segundos, pues aunque no existiera (ni exista) consenso sobre estas zonas, las áreas del cerebro ya se habían usado para realizar estudios similares sobre otros animales.

Otro hecho comprobado era que los **conectomas variarían de una persona a otra**, no solo por las patologías que esta tuviera, sino por temas de edad, genética, desarrollo y plasticidad neuronal, pues en estudios invasivos postmortem ya se había comprobado. Este último término, la plasticidad, supone que el cerebro es capaz de adaptarse modificando su estructura y dinámica, creando nuevas conexiones entre neuronas y formando nuevas redes. Este fenómeno se da durante el aprendizaje, pero también tras enfermedades o lesiones que afecten al sistema nervioso, y se puede dar de manera diferente si una persona tiene algún trastorno.

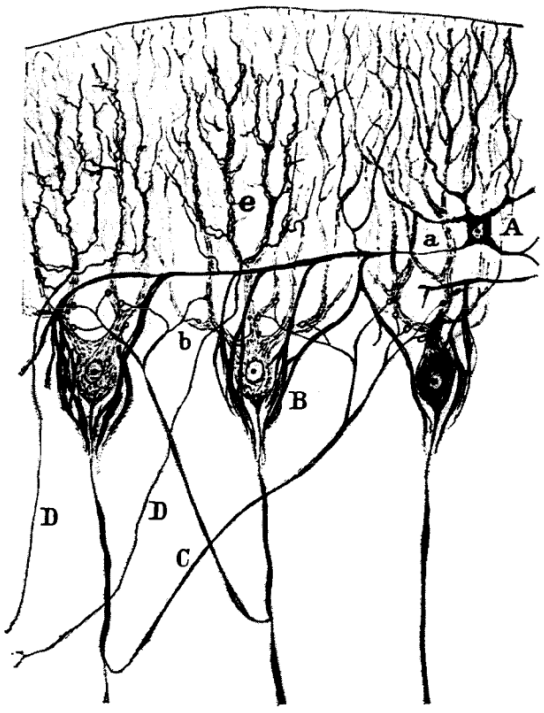

Figura 2: Conexiones entre células de Purkinje y el cerebelo. Ilustración de Ramón y Cajal para su charla tras recibir el premio Nobel [26].

Hoy en día esta definición ha cambiado levemente, ya que en vez de solamente ceros y

unos tenemos un rango de decimales entre ellos que indican el nivel de correlación de dos regiones cerebrales, dejando de ser esta una cuestión de sí o no, sino de cuánto. El proceso de cálculo de conectomas se ha ido estableciendo poco a poco, conforme la comprensión del cerebro y la disponibilidad de métodos ha aumentado.

2.3.1. Cálculo de conectomas

Aunque los conectomas existan hace tiempo, **no hay un proceso estandarizado para calcularlos**. Existen **pasos comunes**, pero muchos de ellos tienen variantes y diferentes opciones que serán seleccionadas según el estudio. Se pueden extraer de diferentes tipos de MRIs aunque en nuestro caso nos enfocaremos en las fMRIs. Se debe tener en cuenta que estos provienen de aparatos de resonancia magnética de diferentes marcas y antigüedad, y en ocasiones cada empresa tiene su propio estándar, lo que crea más variabilidad.

Con esto en mente, usaremos un artículo de **Kamalaker, D.** y compañeros donde se hace una serie de pruebas de los procedimientos seguidos para el cálculo de conectomas a partir de fMRIs en estado de reposo y su uso en predicción de diferentes trastornos [27].

La publicación define las **partes más importantes del cálculo de conectomas** como la definición de las regiones de interés del cerebro, los métodos usados en la eliminación de ruido y el cálculo de la correlación:

Las **regiones de interés** (ROIs) del cerebro se pueden establecer de tres maneras:

- Usando **esferas** de radios entre 5 y 10 milímetros centradas en coordenadas específicas, dadas por **atlas** como el *Power*, con 264 coordenadas para esferas de 5mm.

- Con **altas anatómicos de referencia**, **atlas basados en los surcos cerebrales**, o con **puntos de referencia basados en conectividad**. Tres de los atlas estándares reciben los nombres *Automated Anatomical Labeling* (AAL, con 116 regiones), *Harvard Oxford* (HO, con 118 regiones) y *Bootstrap Analysis of Stable Clusters* (BASC, con entre 36 y 444 regiones).

- Con **métodos de análisis de datos**, como las **k-medias**, el **método de Ward de mínima varianza**, el **Análisis de Componentes Independientes** o el **aprendizaje de diccionarios**.

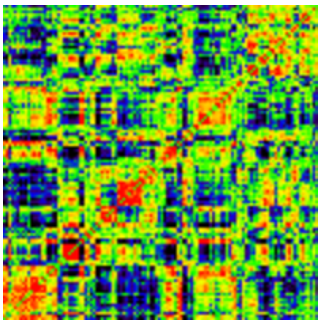

Figura 3: Conectoma de un paciente con TEA.

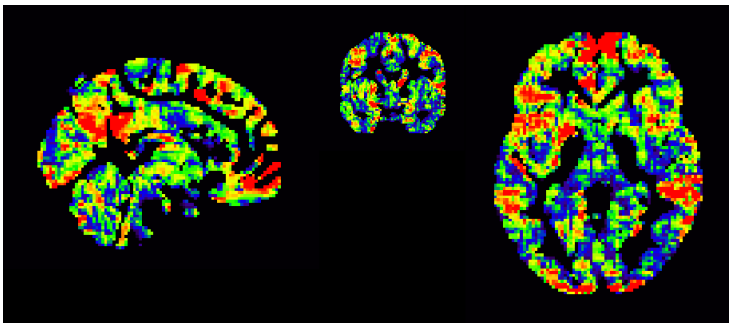

Figura 4: Mapeo del conectoma de un paciente con TEA sobre el cerebro.

Además, estas regiones estudiadas pueden ser **regiones locales** del cerebro o **redes funcionales** del mismo, que pueden abarcar varias regiones a la vez.

Para cada región, se extrae la **serie temporal** de la fMRI, y esta se puede pasar o no por una serie de **estrategias de eliminación de artefactos**. Estas son la **eliminación de la señal causada por movimiento, normalización de la señal, paso por un filtro** (que puede ser de paso banda, bajo o alto) y **regresión de señal**. Este último método puede llevarse a cabo de diferentes maneras.

La **regresión de señal** consiste en eliminar la parte de la señal generada en la adquisición que **no pertenece al vóxel** analizado en cada momento. Esta señal a desechar se puede aproximar de dos maneras principales. La primera es usar la **media de la imagen completa** en cada uno de los instantes de la resonancia para crear una máscara, y la segunda, algo más compleja y precisa, es usar la **media en cada instante de la sustancia blanca y el líquido cefalorraquídeo**, que no deberían de activarse durante el estudio, y crear la máscara a partir de ella. Esta se emplea para eliminar la señal indeseada.

El **cálculo de la relación entre regiones** para definir los conectomas se puede hacer aplicando **correlación total, parcial** o varias modalidades de **matrices complejas de covarianza**.

A estos pasos básicos se les **pueden añadir otros según el método usado**, además de que algunos de ellos pueden o no usarse, como la regresión de señal los filtros, y la normalización. Además, para muchos de los procesos mencionados existen **archivos estándares**, como los atlas, que surgen y cambian conforme pasa el tiempo.

En la *pipeline* empleada en este trabajo se añade el **cambio de orientación a LAS**, co-

rrección de errores causados por los **milisegundos de diferencia entre las capas de los MRIs**, cálculo de **media temporal de la imagen y eliminación de tejido no cerebral**, **normalización** de la imagen con el archivo estándar *MNI152_T1_2mm_brain*, y el **mapeo de densidad de conectividad funcional sobre el cerebro**. Estos pasos se definirán más a fondo en el siguiente capitulo.

3

Cálculo de los conectomas

3.1. Obtención de los MRIs

Las **Imágenes por Resonancia Magnética funcionales** (fMRIs por sus siglas en inglés) usadas en este trabajo se han obtenido de las dos ediciones de la iniciativa **ABIDE** (*Autism Brain Imaging Data Exchange*), publicadas en 2012 y 2016, respectivamente. La recopilación de imágenes de 26 entidades hace que la cantidad de la que partimos sea grande, algo necesario para este tipo de estudios, pues al ser los conectomas tan complejos, es necesaria una muestra amplia para llevar a cabo un buen trabajo.

	Individuos con autismo		Individuos neurotípicos		Totales
	Mujeres	Hombres	Mujeres	Hombres	
ABIDE I	65	473	99	473	1110
ABIDE II	77	444	181	412	1114
Totales	142	917	280	885	**2224**
	1059		1165		

Tabla 1: Detalle de los individuos estudiados en ABIDE

En la **Tabla 1** se aprecia que en total se reunieron 2224 fMRIs, pero también que el reparto no está compensado entre mujeres y hombres, ya que las imágenes de hombres representan alrededor del 80 % de las capturadas. Con esto se añade un ejemplo más de falta de equidad en los estudios, como los ya mencionados en la sección de teoría. Además, mientras que las cantidades de hombres con y sin autismo estudiados apenas se diferencian una de la otra, la cantidad de mujeres con autismo estudiadas es la mitad de la cantidad de mujeres neurotípicas estudiadas.

La recopilación de estas imágenes se hizo previa autorización de acceso al recurso **Neuro-Imaging Tools & Resources Collaboratory** (NITRC) [28], en el que se encuentran recopiladas. En el caso de la primera edición de ABIDE existen varios medios a través de los que es posible descargarse las imágenes, de entre los que se ha elegido la plataforma **LONI Image & Data Archive** [29], que aporta mayor velocidad de descarga que sus compañeras. Por desgracia, las imágenes de ABIDE II solamente se pueden obtener a través de los enlaces disponibles en la web de ABIDE [30], lo que supone tener que acceder a ellas entidad por entidad, con menor velocidad debido a que el protocolo FTP con el que funciona solamente permite descargar dos archivos a la vez.

Finalizada la descarga, podemos explorar los datos. En el directorio de cada estudio se incluyen carpetas nombradas por el identificador numérico de cada paciente. Este identificador cuenta con cinco dígitos, que en el caso de ABIDE I están en el rango de 50002 a 51607, y en el de ABIDE II de 28675 a 30256. Por ahora, se sigue trabajando en directorios separados para cada estudio por razones de formato, ya que las subcarpetas en las que se encuentran los MRIs de uno son diferentes a las del otro. Para poder crear la división entre hombres y mujeres con y sin autismo, se accede a las **tablas de datos clínicos de cada estudio**, de las que se extraen los identificadores numéricos de los pacientes de cada categoría, empleados luego en un script bash que mueve los archivos a cuatro carpetas nombradas según su contenido, dentro de los directorios de cada estudio.

3.2. Código usado

Una vez hecho esto, se puede comenzar a trabajar con la *pipeline* **Python/FSL Resting State** [31]. Pero antes de poder ejecutarla, hay que solucionar los errores existentes en sus dos archivos. Estos provienen principalmente de que ambos códigos están creados para la segunda versión de Python, que en enero de 2020 dejó de tener soporte. Con ello, muchos paquetes no están disponibles, por lo tanto no existe la opción de usar Python 2 para ejecutar el proceso. En su lugar, se deben realizar modificaciones en el código para actualizarlo a Python 3, tanto por cuestiones de cambio de formato a la hora de realizar llamadas a funciones como por casos en los que las funciones usadas se han eliminado de su paquete y deben cambiarse. Estos cambios se complican por la falta de comentarios en el código, pues sus casi mil quinientas líneas tienen menos de doscientos comentarios que las expliquen.

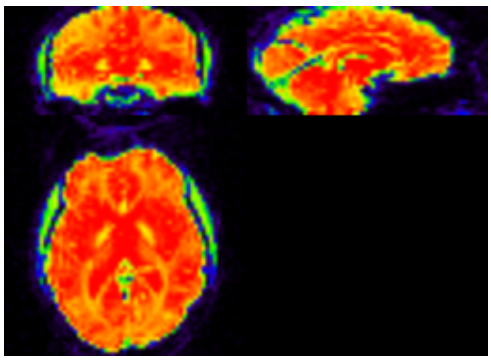

Figura 5: Resonancia magnética funcional obtenida de ABIDE.

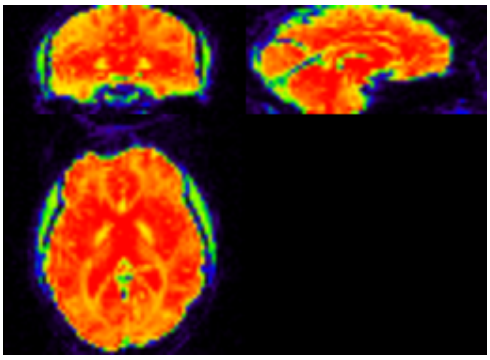

Figura 6: Corrección de pérdida de información con slicetimer (paso 1).

Para poder ejecutar el código, primero observamos lo que contiene. Al ser un flujo de trabajo, cuenta con un total de nueve pasos, cada uno de ellos cumpliendo las siguientes tareas, cuyos resultados ilustraremos empleando las imágenes más significativas obtenidas del procesamiento de la fMRI del sujeto 50005 (**Figura 5**).

- **Paso 0**: Toma la imagen y la reorienta a orientación de ejes *LAS*, en la que las coordenadas del eje X toman valores de izquierda a derecha, el eje Y corresponde al plano antero-posterior, y las coordenadas del eje Z indican altura. Esta conversión se realiza porque LAS es la orientación de la plantilla T1 de FSL. En esta paso se crean los archivos *func_LAS.bxh, func_LAS.nii.gz* y sliceorder.txt. Los dos primeros son la imagen en el nuevo sistema de coordenadas, con dos formatos diferentes, y el último un archivo donde se especifica el orden de las capas procesadas.
- **Paso 1**: Ejecuta la herramienta *slicetimer* de FSL, que corrige la pérdida de información y errores causados por los milisegundos de diferencia entre la toma de cada capa. Esta corrección se consigue interpolando valores de cada vóxel para ajustarlos y permitir que se considere que todas las series temporales han sido tomadas en el mismo momento. Crea una imagen *func_LAS_st.nii.gz* (**Figura 6)**, con estas correcciones.
- **Paso 2**: Ejecuta *mcflirt* de FSL, que corrige errores causados por el movimiento del paciente durante la toma de la resonancia usando regresión. Esto da lugar a cinco archivos, *func_LAS_st_mcf.nii.gz, func_LAS_st_mcfr.nii.gz (**Figura 9**), func_LAS_st_mcf.par, func_LAS_st_mcf_rot.png* (**Figura 7**) y *func_LAS_st_mcf_trans.png* (**Figura 8**). Se corresponden con las imágenes tras las correcciones, un archivo .par con los parámetros

de movimiento sobre los que se ha aplicado regresión, y dos gráficas con las estimaciones de movimiento de rotación y traslación.

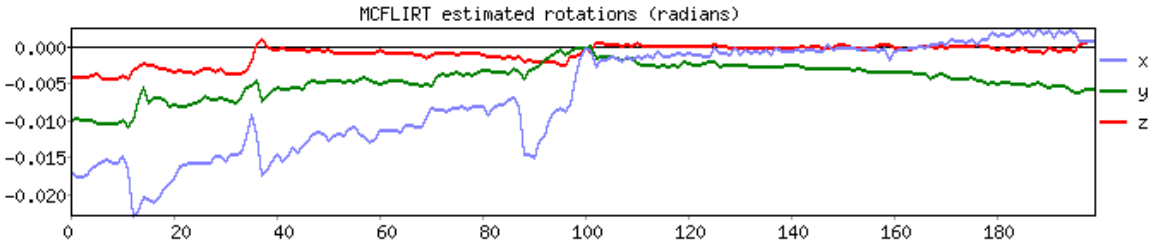

Figura 7: Estimación del movimiento de rotación (paso 2).

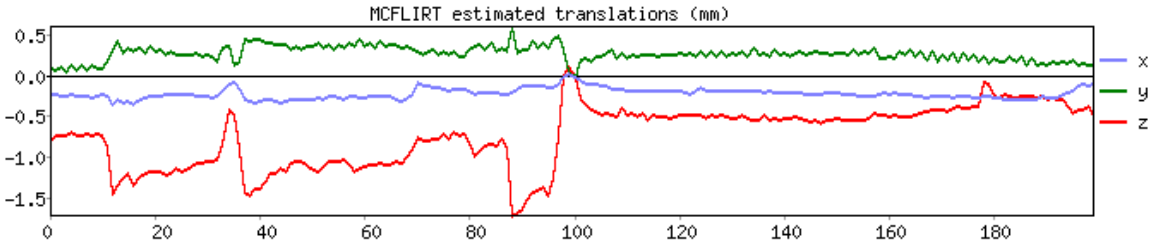

Figura 8: Estimación del movimiento de traslación (paso 2).

- **Paso 3**: Se calcula la media temporal de la imagen con *fslmaths*, y luego se le aplica *BET*, también de FSL, que elimina todo el tejido no perteneciente al cerebro de la imagen usada. Se crean los archivos mean_*func.nii.gz*, con la media temporal; *mean_func_brain_mask.nii.gz* (**Figura 10**), la máscara a aplicar para obtener solo el tejido cerebral; *mean_func_brain.nii.gz*, la media tras la máscara; y *func_LAS_st_mcfr_brain.nii.gz* (**Figura 11**), la imagen final con la máscara y la media aplicadas.

- **Paso 4**: Realiza la normalización de los datos usando *flirt*, con el estándar *MNI152_T1_2mm_brain*, creando una imagen nueva. Este estándar es una imagen cerebral de alta calidad creada por el Instituto de Neurología de Montreal, que se emplea para minimizar posibles errores en la nuestra. Crea los archivos *func_LAS_st_mcfr_brain_norm.mat* y *func_LAS_st_mcfr_brain_norm.nii.gz* (**Figura 12**), que son la matriz ASCII de trasformación y la imagen tras ser normalizada, respectivamente.

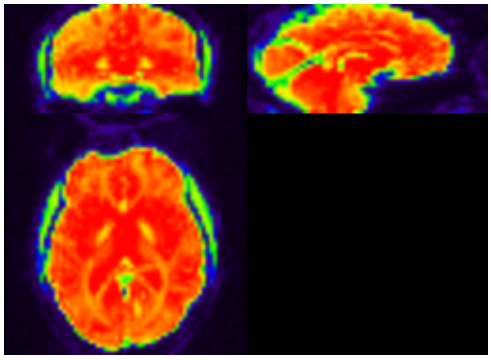

Figura 9: Corrección de movimiento con mcflirt (paso 2).

Figura 10: Máscara del tejido cerebral (paso 3).

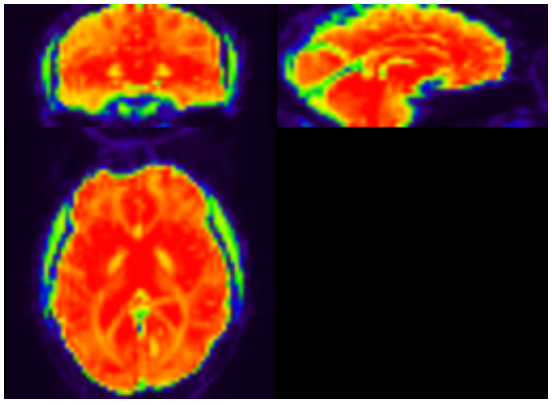

Figura 11: Imagen tras aplicar la máscara y la media temporal (paso 3).

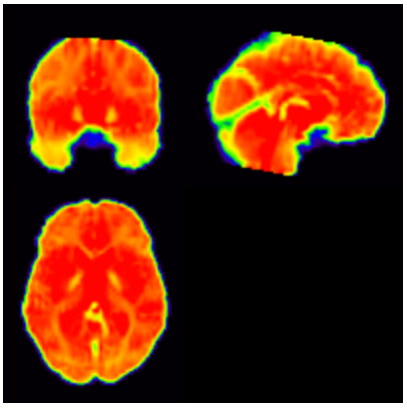

Figura 12: Normalización de la imagen (paso 4).

- **Paso 5**: Este paso realiza una **regresión de señal** con máscaras de sustancia blanca y líquido cefalorraquídeo, haciendo la media de estas dos en cada instante, pues al no haber activación en ellas debería de ser constante (aunque realmente no lo sea por ruidos y otros factores externos). Esta media se sustrae a la señal obtenida para mantener solamente la señal de cada vóxel, sin otras provenientes de otras fuentes (ruido, el gradiente electromagnético creado para realizar la resonancia, etc.). Se crea la imagen *func_LAS_st_mcfr_brain_norm_wmcsf.nii.gz* y los archivos *csf_ts.txt y wm_ts.txt*, con las máscaras de líquido cefalorraquídeo y sustancia blanca.
- **Paso 6**: Filtra los datos para eliminar frecuencias, con un **filtro de paso banda**, reteniendo las frecuencias en el intervalo [0,001, 0,08]. Produce una sola imagen con este filtro aplicado, *filt_func_LAS_st_mcfr_brain_norm_wmcsf.nii.gz*.
- **Paso 7**: Usa el archivo estándar *aal_MNI_V4* para extraer las series temporales medias de las 116 regiones de interés (ROI) del escaneo, y calcula los coeficientes de correlación de la matriz. Este archivo estándar pertenece al atlas nombrado en el capítulo anterior, *Automated Anatomical Labelling*, y como el usado en el cuarto paso, pertenece al *Montreal Neurological Insitute*. Define las correspondencias entre los vóxeles y 116 regiones anatómicas de sustancia gris. Se crean los archivos *subject.graphml* (imagen en formato graphml con las regiones y los valores de correlación; *corrlabel_ts.txt*, serie temporal de cada región; *r_matrix.nii.gz* y *r_matrix.csv*, la matriz de correlación en dos formatos; *zr_matrix.nii.gz* (**Figura 13**) y *zr_matrix.csv*, la matriz de correlación normalizada en dos formatos; y *mask_matrix.nii.gz*, una máscara de inclusión para todos los vóxeles fuera de la intersección de regiones.
- **Paso 8**: Realiza el mapeo de densidad de conectividad funcional sobre el cerebro, produciendo una imagen en tres dimensiones en la que se puede apreciar el nivel de correlación entre un vóxel y sus vecinos, es decir la correlación en una misma región. Esta imagen lleva el nombre *fcdm.nii.gz* (**Figura 14**), cuanto más rojo sea un punto mayor correlación tendrá con sus vecinos, y cuanto más se acerque al azul, menos.

Para poder calcular los conectomas necesarios, tenemos que ejecutar los pasos del 0 al 7. Este último produce siete archivos, de los que nos interesan el archivo *r_matrix.csv* y el *zr_matrix.csv*, ya que contienen los coeficientes de correlación brutos y normalizados, respec-

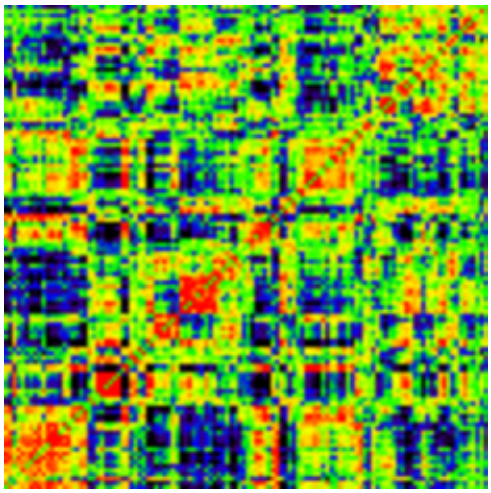

Figura 13: Normalización del conectoma (paso 7)

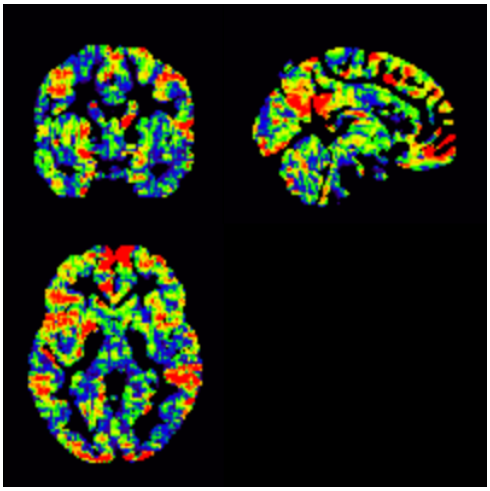

Figura 14: Conectoma mapeado sobre el cerebro (paso 8).

tivamente. Para realizar estas ejecuciones se han creado una serie de *scripts* en lenguaje bash que automatizan el proceso lo máximo posible. Completarlas supuso unos seis minutos de espera por imagen, y finalmente se obtuvieron alrededor de 2 TB de información, de los que solamente 550 MB son conectomas. El resto de espacio se ocupa por los archivos intermedios generados durante la ejecución.

Este proceso no ha sido capaz de extraer los conectomas de todos los MRIs disponibles. Cabe destacar que todas las resonancias que han dado problemas, menos una, provienen de la segunda edición de ABIDE. En algunos casos tienen tamaños demasiado grandes para realizar su normalización (algunas llegan a los 300 MB), por lo que sus conectomas no han podido ser calculados; y en otros, los errores se producían en otros pasos del cálculo, sobre todo en el séptimo. Con esto, la cantidad de datos que se podrá utilizar en este trabajo (**Tabla 2**) será menor a la original, siendo la **cifra final de conectomas 2111** (con lo que se han podido usar el 95 % de las imágenes iniciales).

Por suerte, la pérdida de datos se ha producido en todas las categorías, por lo que no hay gran cambio en las proporciones respecto a las iniciales.

Teniendo los conectomas calculados, podemos pasar a la creación de modelos de predicción.

	Individuos con autismo		Individuos neurotípicos		Totales
	Mujeres	Hombres	Mujeres	Hombres	
ABIDE I	64	473	99	473	1109
ABIDE II	70	405	163	364	1002
Totales	134	878	262	837	**2111**
	1012		1099		

Tabla 2: Detalle de la cantidad de conectomas calculados

4

Modelos de predicción

Para poder cumplir el objetivo final, crear **modelos de predicción** para **distinguir entre el conectoma de una persona perteneciente al espectro autista y el de una persona neurotípica**, tenemos que realizar varias **pruebas con diferentes tipos de clasificadores**, ajustando sus hiperparámetros para obtener las mejores combinaciones.

4.1. Metodología

Debido a las diferencias en la manifestación del autismo en mujeres y hombres, que se pueden deber a diferentes manifestaciones de este en sus conectomas, se crearán **modelos para clasificar conectomas de mujeres, de hombres, y de ambos sexos**. Con esto buscamos determinar si el mejor enfoque es **usar diferentes modelos para cada sexo o si, por el contrario, no es necesario**. Como la predicción que queremos realizar es de naturaleza discreta, teniendo las opciones de paciente TEA o neurotípico, todos nuestros modelos emplearán **métodos de clasificación**.

Debemos recordar que aunque los datos de hombres estaban bastante equilibrados en cuanto a sujetos pertenecientes al espectro y sujetos control, este no es el caso de los datos de mujeres, pues entre ambos estudios y tras pasar el proceso de cálculo de conectomas, obtenemos 134 conectomas de mujeres con autismo y 262 de mujeres sin autismo. Por tanto, los 99 conectomas de mujeres neurotípicas de ABIDE I serán apartados durante la creación de modelos, quedando finalmente 163 conectomas de mujeres del grupo control, pertenecientes a la segunda edición de ABIDE. Esta decisión se ha tomado tras varias pruebas de clasificación simples con el conjunto de datos de mujeres ampliado y reducido, en las que se ha comprobado que los resultados de precisión del segundo son mejores, pues **eliminamos el desequilibrio**, y

en los algoritmos de clasificación que vamos a usar se requiere que **no haya predominancia de una clase**.

Para facilitar el trabajo de creación de los modelos, se empleará **un único *script* de Python**, en el que se dará la opción de usar los conectomas de mujeres, hombres, o los dos, y de elegir entre los modelos de clasificación disponibles. Esto nos **evita crear múltiples programas que repitan código**, a cambio de escribir funciones que serán llamadas según sean necesarias. Una vez elegida la combinación de conjunto de conectomas y modelo de clasificación, se ejecuta un bucle for dentro del que se accede a cada carpeta de paciente dentro del directorio elegido, y se crea una variable *name* usando este *path*, del que extraemos la edición del estudio, el sexo del paciente, el grupo al que pertenece, y su identificador de cinco dígitos. Cada uno de los nombres tendrá la estructura *AI/AII + _ + fem/male + _ + autism/control + _ + ID*. Por ejemplo, el paciente 28947, que pertenece al estudio ABIDE II y es una mujer del grupo control, se correspondería con el nombre *AII_fem_control_28947*.

Además de esta variable, extraemos el **vector de la parte triangular superior a la diagonal de la matriz de cocientes de correlación normalizados** (el archivo *zr_matrix.csv*). Lo extraemos de esta manera porque la matriz del conectoma es una **matriz simétrica**, por lo que de usarla en su completitud estaríamos malgastando capacidad de proceso, pues no haríamos más que introducir los mismos datos dos veces, y teniendo en cuenta que cada matriz es de tamaño 116x116 (con un total de 13146 elementos), esto no es conveniente. Además, **no guardamos la diagonal** porque al ser esta una matriz de coeficientes de correlación entre regiones, la diagonal representa la relación de cada región consigo misma, lo que de nuevo es irrelevante pues es constante en todos los conectomas, y por tanto no influye en nuestra clasificación. A este vector se le añade al comienzo un 1 o un 0 dependiendo de si el paciente pertenece al TEA o no. De esta manera, cada vector tiene en su primera columna la **variable binaria que indica si pertenece al espectro** y en el resto los **coeficientes de correlación** de la parte superior de la matriz. La variable y el vector se introducen en un diccionario, que una vez terminado el bucle se transforma en **estructura *dataframe***, cuya primera columna es el valor 0 o 1 de la variable TEA. Este *dataframe* se divide entre esta columna y el resto, para usarlos como valor a predecir y valores con los que hacerlo en los modelos de predicción.

Para crear dichos modelos se usa el paquete *scikit-learn* [32] como apoyo principal, pues este posee tanto los algoritmos de clasificación como los de configuración de hiperparámetros

que usaremos. Los algoritmos de clasificación usados serán ***K-Nearest Neighbors, Decision Trees, Random Forest, Support Vector Machine y Neural Networks***. A los datos elegidos se les aplicará validación cruzada estratificada con k=5, manteniendo las proporciones de individuos de autismo y control ponderadas en cada experimento con la función *StratifiedKFold* de *scikit-learn*. Por cada una de estas cinco divisiones, se ejecutará la función correspondiente al modelo elegido, en las que habrá siempre un modelo creado **sin ningún ajuste** y otros creados con **métodos de ajuste de hiperparámetros**. Al final de esta ejecución se imprimirán por pantalla las **precisiones** de cada modelo, la **precisión media** y **desviación típica** y la **mejor precisión** para entrenamiento y prueba, además del **tiempo transcurrido** entre que se comenzó la creación de modelos y su finalización.

4.2. Algoritmos de clasificación

A continuación se detalla el funcionamiento de cada uno de los algoritmos de clasificación usados, junto a sus parámetros disponibles. El uso de estos se detallará en sus capítulos propios, más adelante.

4.2.1. *K-Nearest Neighbors*

El método de los **k vecinos más cercanos** usa aprendizaje supervisado tanto para problemas de regresión como de clasificación, en los que nos centraremos por ser este nuestro caso. Funciona tomando **datos de entrenamiento en los que se incluye la clase** a la que pertenece cada elemento, almacenándolos para **usarlos como método de decisión**. Se le indica también un **número k** correspondiente al **número de vecinos con los que queramos comparar** cada elemento a evaluar. A la hora de clasificar otros datos, toma cada uno de los individuos y busca sus k vecinos más cercanos dentro de los datos de entrenamiento. La **clase de cada uno de ellos cuenta como un voto**, y la **clase predicha** para el nuevo individuo será la que **más votos tenga**. Se basa, por tanto, en el concepto de que **individuos de clases similares están cercanos en el espacio**. El número k que elijamos impactará directamente en la calidad del modelo, y en el caso de tener clases pares, como es el nuestro, debemos tomar un número impar, para que los votos nunca resulten en empate.

El clasificador usado será el perteneciente al paquete *scikit-learn*, *KNeighborsClassifier*, que

contiene parámetros para el número de vecinos a tener en cuenta (*n_neighbors*), la función de pesos usada para los puntos del vecindario (*weights*), el algoritmo para calcular los vecinos más cercanos (*algorithm*), la cantidad máxima de nodos hoja de estos algoritmos (*leaf_size*), el parámetro en caso de usar distancia Minkowski (*p*), la métrica de distancia a usar (*metric*), otros parámetros para la métrica (*metric_params*) y el número de trabajos paralelos a realizar (*n_jobs*).

4.2.2. *Decision Trees*

El algoritmo de los **árboles de decisión** es otro método de aprendizaje supervisado, que además de en clasificación se puede aplicar a la regresión. Se encarga de crear un modelo basado en **reglas de decisión**, que serán más o menos complejas según la profundidad del árbol. Precisamente esta profundidad es una de las principales causas de **sobreajuste**, es decir, ajustar demasiado el modelo a los datos de entrenamiento, siendo este demasiado específico y por tanto no tan bueno para clasificar otros datos. También hay que prestar atención a los datos que se le introducen, pues si en ellos se repite una clase mucho más que otra, el modelo no será equitativo.

Para crear el modelo, el árbol de decisión toma los datos de entrenamiento y va **dividiéndose separando dos secciones de estos cada vez**. Estas separaciones dependerán de alguna característica en la que se diferencien ambos conjuntos. Las decisiones de separación vendrán influenciadas por los parámetros del algoritmo, que en nuestro caso es *DecisionTreeClassifier* de *scikit-learn*. Sus parámetros son: la función de medida de calidad de una división (*criterion*), la estrategia de división (*splitter*), la máxima profundidad o número de niveles que puede tener (*max_depth*), el mínimo de elementos que debe pertenecer en un nodo para poder dividirlo, evitando así crear divisiones muy específicos (*min_samples_split*), el número mínimo de individuos que debe haber en un nodo hoja, para no crear nodos muy específicos (*min_samples_leaf*), la fracción mínima del total de pesos que deben pertenecer a cada nodo (*min_weight_fraction_leaf*), el número de características que tener en cuenta para encontrar la mejor división (*max_features*), el número máximo de nodos hoja permitidos (*max_leaf_nodes*), la cantidad mínima de reducción de impureza que necesita dar una división (*min_impurity_decrease*), la impureza mínima requerida para dividir un nodo (*min_impurity_split*), los pesos de cada clase (*class_weight*), un parámetro de complejidad para la poda, que por de-

fecto no se usa (*ccp_alpha*), y una variable para controlar la aleatoridad del estimador (*random_state*).

4.2.3. *Random Forest*

El algoritmo *RandomForestClassifier* de *scikit-learn*, o **bosques aleatorios**, pertenece a los **métodos de ensamblado**, que usan las predicciones de varios modelos para alcanzar una decisión. En el caso de *random forest*, estos estimadores son **árboles de decisión** cuyos resultados probabilísticos (en vez de su voto) se unen para obtener la media, que supondrá la predicción final. El objetivo de estos es **evitar el sobreajuste** típico de los árboles de decisión, ya que al tener muchos árboles, los errores o problemas individuales de estos se diluyen.

Al ser un conjunto de árboles, repite muchos de sus parámetros, a saber: *criterion, max_depth, min_samples_split, min_samples_leaf, min_weight_fraction_leaf, max_features, max_leaf_nodes, min_impurity_decrease, min_impurity_split, class_weight, n_jobs, ccp_alpha y random_state*. A estos le añadimos los parámetros para cantidad de árboles a crear (*n_estimators*), evitar el uso de todo el conjunto de datos para entrenar (*bootstrap*), validar el modelo con los datos que no se hayan usado (*oob_score*, sólo si se está usando *bootstrap*) y reutilización de la solución anterior (*warm_start*).

4.2.4. *Support Vector Machines*

El algoritmo de **máquinas de vectores de soporte** es el cuarto método de aprendizaje supervisado que usaremos, de nuevo en su modalidad de clasificación. Este clasificador toma los datos de entrenamiento y crea uno o varios **hiper planos** que dividen el espacio en que están contenidos, idealmente dejando los miembros de una misma clase en una misma división. Se considera como mejor hiperplano aquel que más distancia tiene entre sí y el punto más cercano.

Cuando tenemos un **problema no lineal**, es decir, que no existen líneas rectas que puedan dividir ambas clases, la **función kernel** usada gana importancia. Esta **transforma los datos introducidos** para usarlos en el espacio con el que el algoritmo puede trabajar, donde el problema sí será lineal y por tanto el algoritmo podrá resolverlo como tal. Los *kernels* pueden ser desde **lineales a polinómicos**, y el tipo que necesitemos va a depender de las características de nuestro problema.

El método usado será SVC de *scikit-learn*, que tiene parámetros para regularización (*C*), función *kernel* (*kernel*), grado en caso de *kernel* polinómica (*degree*), coeficiente gamma para *kernels* polinómicas, sigmoides y de base radial (*gamma*), coeficiente independiente en polinómicas y sigmoides (*coef0*), uso o no de heuristica de encogimiento (*shrinking*), permitir estimaciones de probabilidad (*probability*), tolerancia de parada (*tol*), tamaño de caché de *kernel* (*cache_size*), máximo de iteraciones (*max_iter*), forma de la función de decisión (*decision_function_shape*) y si se rompen empates según la confianza de la función de decisión (*break_ties*). Además de estos, repiten los anteriormente mencionados *class_weight, verbose y random_state.*

4.2.5. *Neural Networks*

Las **redes neuronales** son el último método que usaremos, también de aprendizaje supervisado. Las redes neuronales pueden ser más o menos complejas, pero mantienen unos **componentes básicos**. Tienen una **capa de entrada y otra de salida** entre las que puede haber una o más **capas ocultas**. La capa de entrada tiene **neuronas** que representan los **datos introducidos**, y en el caso de las capas ocultas, estas neuronas tienen **pesos** que multiplican los valores que llegan a ellas, formando una suma de multiplicaciones que se une a la **función de activación**. Los valores van transformándose capa a capa hasta la capa de salida, que indica a qué clase pertenece el individuo introducido.

Durante la **fase de entrenamiento**, los pesos de las diferentes capas se entrenan para conseguir que la red neuronal de los resultados buscados. En el caso de *scikit-learn*, se emplea un **preceptrón multicapa** (MLPClassifier) que se entrena mediante *backpropagation*, es decir, tomando las salidas y corrigiendo hacia atrás. Sus parámetros permiten elegir tamaño de las capas ocultas (*hidden_layer_sizes*), función de activación de las capas ocultas (*activation*), manera de optimizar los pesos (*solver*), parámetro de penalización (*alpha*), tamaño de muestras para optimizadores estocásticos (*batch_size*), tasa de aprendizaje (*learning_rate*), tasa de aprendizaje inicial (*learning_rate_init*), exponente para tasa de aprendizaje de tipo decreciente (*power_t*), barajar o no las muestras (*shuffle*), mínimo de mejora para continuar (*tol*), ritmo de descenso del gradiente para optimización de pesos sgd (*momentum*), uso de momentum de Nesterov (*nesterovs_momentum*), parada automática si la precisión de validación no mejora (*early_stopping*), fracción a apartar para la validación (*validation_fraction*), valores para la

función de optimización de pesos adam (*beta_1, beta_2, epsilon*), número de iteraciones que esperar sin cumplir el mínimo de mejora (*n_iter_no_change*) y número máximo de llamadas a funciones de pérdidsa al usar lbfgs para optimización de pesos (*max_fun*) Repite también los parámetros *max_iter, random_state, verbose y warm_start.*

4.3. Ajuste de hiperparámetros

Como hemos podido ver, todos los algoritmos tienen varios parámetros, y para obtener la mejor combinación de ellos debemos de aplicar **métodos de ajuste**. Estos se encargan de generar **combinaciones de hiperparámetros** a partir de un **diccionario** de ellos, y crear y probar los modelos sobre el conjunto de entrenamiento hasta obtener la mejor combinación. No existe un solo enfoque para realizar esta tarea, sino varios métodos con sus ventajas y desventajas. En concreto, en este código se explorarán tres:

4.3.1. *Grid search* con validación cruzada

El método de *grid search* (*GridsearchCV* en *scikit-learn*), búsqueda **dentro de cuadrícula**, recorre **todas las combinaciones posibles** dentro del diccionario que se le da como parámetro. Además, implementa internamente **validación cruzada**, que por defecto es del tipo 5-*fold*. Con esto garantiza encontrar la mejor combinación de parámetros, pero conforme aumenta la cantidad de ellos aumenta también mucho el tiempo que emplea, por lo que es mejor usarla con **diccionarios de parámetros pequeños**.

Sus parámetros son: el modelo de estimación a optimizar (*estimator*), el diccionario de hiperparámetros para dicho modelo (*param_grid*), el método de evaluación a usar dentro de la validación cruzada (*scoring*), parámetros para ejecución en paralelo (*n_jobs y pre_dispatch*), la opción de recalcular el modelo para todos los datos introducidos con los mejores parámetros (*refit*), el tipo de validación cruzada (*cv*), la cantidad de información a imprimir en consola (*verbose*), método de gestión de errores (*error_score*).

4.3.2. Búsqueda aleatoria con validación cruzada

Este **método de búsqueda aleatoria** (*RandomizedSearchCV* en *scikit-learn*) trabaja de manera muy similar al anterior, solo que en vez de tomar todas las posibles combinaciones de

un diccionario, toma un número dado. Esto hace que **no tarde tanto tiempo** y sea un buen método para casos de **diccionarios largos y algoritmos de clasificación que consuman mucho tiempo**. Las leyes de probabilidad indican que, sin importar la cantidad de combinaciones posibles, con **60 iteraciones** se pueden conseguir modelos dentro del **5 % mejor y con una confianza del 95 %**, y con **460 aquellos dentro del 1 % mejor con confianza del 99 %** [33]. Por tanto, aunque el diccionario sea amplio no necesitará recorrerse entero para obtener un buen modelo, y el número de iteraciones que realicemos dependerá de la confianza y grado de calidad que busquemos.

Sus parámetros son iguales a los de *GridsearchCV*, con la diferencia de que *param_grid* pasa a ser *param_distributions* y además se añade el parámetro *n_iter*, correspondiente al número de combinaciones de parámetros a probar.

4.3.3. Búsqueda bayesiana con validación cruzada

Este método de búsqueda emplea optimización bayesiana para encontrar los mejores parámetros, con un uso similar a los anteriores. No pertenece a *scikit-learn*, sino a *scikit-optimize*, y en el momento del desarrollo de este trabajo no está adaptado para la versión 0.24 de *scikit-learn*, paquete que usa en repetidas ocasiones. Esto se soluciona bajando la versión de *scikit-learn* a la 0.23. Aun haciendo este cambio, no ha sido posible integrarlo en nuestro proceso de creación de modelos, pues al emplearlo surgían otros errores que no han podido solucionarse.

5

K vecinos más cercanos

Para obtener los mejores modelos para el **método de vecinos más cercanos**, hemos determinado que de los 8 parámetros disponibles para el algoritmo KNN, dejaremos tres de ellos en sus **valores predeterminados**. Estos serán el **número de trabajos paralelos ejecutados**, que se quedará como uno solo; los **parámetros adicionales para la métrica usada**, que no será ninguno, y el **número p**, que se mantendrá en 2, porque al servir para modificar la métrica Minkowski y convertirla en Manhattan o euclídea (según p sea 1 o 2), ambas métricas estando presentes ya como posibilidades, no será necesario darle varios valores y aumentar las iteraciones de la búsqueda de parámetros.

Quedan otros cinco parámetros, para los que usaremos en los **modelos con búsqueda** un diccionario donde estableceremos sus posibles valores a tomar, que iremos ajustando durante varias ejecuciones. Debido a la baja cantidad de combinaciones existentes, no usaremos la búsqueda aleatoria, al ser relativamente pequeña la cantidad de candidatos, el tiempo consumido en las iteraciones de búsqueda aleatoria serían casi el mismo que en *grid search* para poder tener una buena confianza. En el caso de los **modelos sin búsqueda**, empezaremos con valores por defecto y después ajustaremos el tamaño de hoja y el número de vecinos.

5.1. Ejecución 1

En esta ejecución dejamos los valores por defecto de todos los parámetros para el cálculo sin búsqueda de hiperparámetros, y para la búsqueda de *grid search* dejamos el número de vecinos y el tamaño de hoja por defecto, 5 y 30; y obtenemos los modelos con las diferentes combinaciones de los demás hiperparámetros, respectivamente.

Estos serán:

- **Función de peso** (*weigth*): los pesos podrán ser iguales para todos (*uniform*) o mayores cuanto más cerca estén (*distance*).

- **Algoritmo de cálculo de los vecinos más cercanos** (*algorithm*): las opciones son fuerza bruta (*brute*), *ball tree* (*ball_tree*) y árbol k-dimensional (*kd_tree*).

- **Métrica de distancia** (*metric*): euclídea (*euclidean*), Manhattan (*manhattan*), de Chebyshev (*chebyshev*) y de Minkowski (*minkowski*).

5.1.1. Resultados en mujeres

La ejecución del programa completo, es decir, la creación de un modelo KNN para cada una de las 5 distribuciones de datos de entrenamiento y prueba, requirió 1 minuto y 5 segundos, obteniéndose los resultados que podemos ver en la **Tabla 3** y la **Figura 15**.

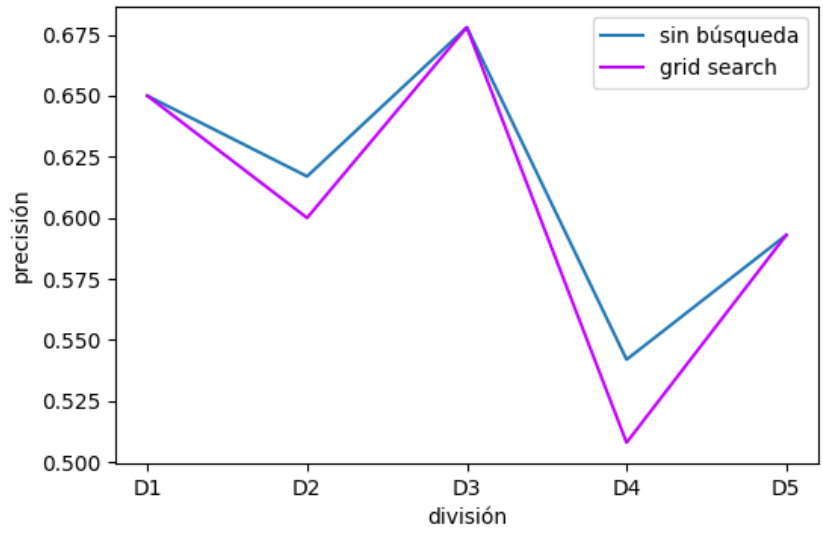

Figura 15: Valores de precisión para la ejecución 1 de KNN en mujeres.

	D1	D2	D3	D4	D5	Media	Desviación típica
Sin ajustes	0.650	0.617	**0.678**	0.542	0.593	0.616	0.052
Grid search	0.650	0.600	**0.678**	0.508	0.593	0.606	0.065

Tabla 3: Valores de precisión para la ejecución 1 de KNN en mujeres.

Se ha llegado a una precisión máxima del **67,8 %** en la distribución 3, tanto sin ajustes como con *grid search*. Este **modelo** usa algoritmo *ball tree*, métrica euclídea, peso uniforme, tamaño de hoja 30 y 5 vecinos a tener en cuenta. La **precisión media** de los modelos llega al **60 %** en ambos casos, pero es algo mejor en el cálculo del modelo sin ajustes, igual que la desviación típica. Esta es de un 5 % y un 6 % para cada método, aunque en ambos la distancia entre la mayor y menor precisión sea de más de diez puntos.

En datos de entrenamiento, la precisión es del **75 %** en ambos modelos, por lo que podemos considerar que tenemos poco sobreajuste.

5.1.2. Resultados en hombres

La creación de los modelos para los conectomas de hombres ha requerido 34 minutos y 6 segundos, mucho más que la anterior. Esto es, como se comentaba antes, debido a la diferencia entre ambas cantidades de datos. Los resultados se ilustran en la **Tabla 4** y la **Figura 16**.

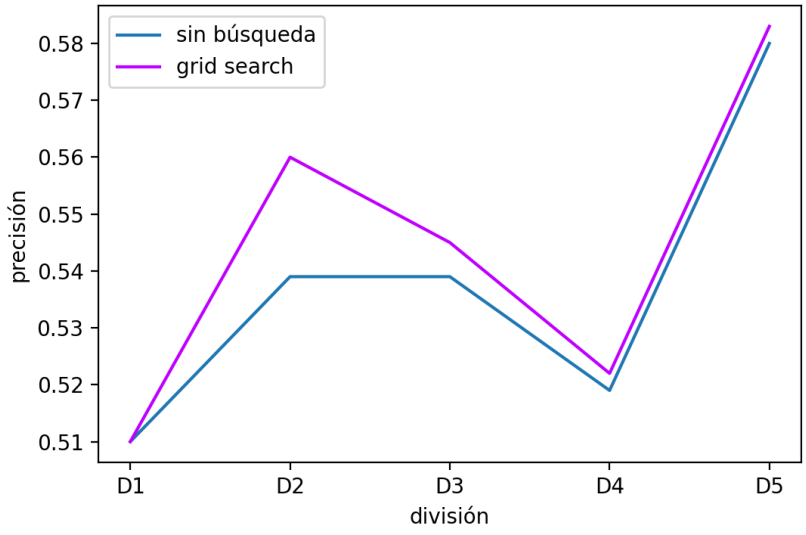

Figura 16: Valores de precisión para la ejecución 1 de KNN en hombres.

	D1	D2	D3	D4	D5	Media	Desviación típica
Sin ajustes	0.510	0.539	0.539	0.519	**0.580**	0.538	0.027
Grid search	0.510	0.560	0.545	0.522	**0.583**	0.544	0.029

Tabla 4: Valores de precisión para la ejecución 1 de KNN en hombres.

En hombres, encontramos resultados algo peores, pues las medias de precisión de ambos cálculos no llegan nunca al **55 %**. El **mejor modelo** se consigue en la quinta distribución, donde ambos métodos consiguen prácticamente el mismo resultado, un **58 %** y un **58,3 %** de precisión respectivamente. Esta se consigue con un algoritmo *ball tree*, métrica euclídea, peso dependiente de la distancia, tamaño de hoja 30 y 5 vecinos a tener en cuenta.

A diferencia de lo que encontrábamos en los modelos para mujeres, la media de precisión en entrenamiento es del **73 %** en el cálculo sin ajustes y del **89 %** en el cálculo con *grid search*, llegándose con este método hasta un **100 %** de precisión, lo que nos indica que en este caso sí se está produciendo sobreajuste.

5.1.3. Resultados en ambos sexos

Finalmente, los modelos de vecinos más cercanos para personas en general han tardado 52 minutos y 18 segundos en crearse, de nuevo más que el apartado anterior, y desde luego mucho más que los destinados a mujeres. En la **Tabla 5** y la **Figura 17** se detallan sus resultados, similares a los obtenidos en hombres.

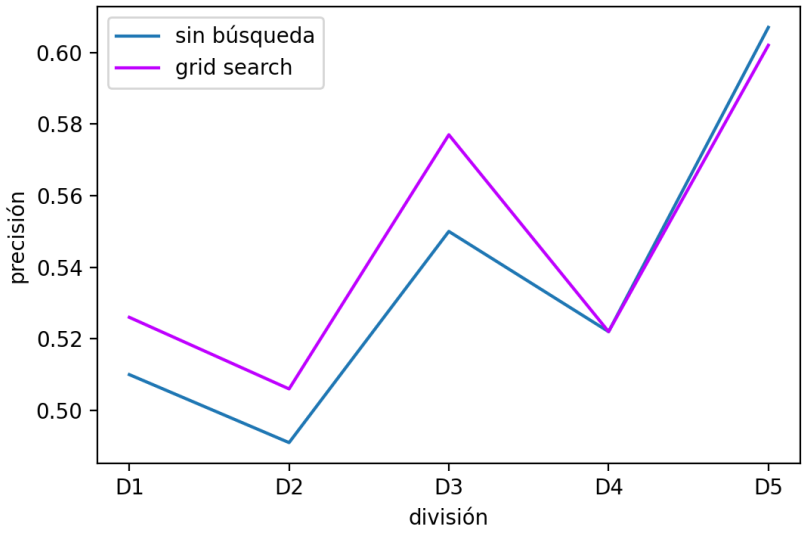

Figura 17: Valores de precisión para la ejecución 1 de KNN en ambos sexos.

Las medias de precisión son muy similares a las obtenidas con los conectomas de hombres, de nuevo sin llegar al **55 %**. El **mejor modelo**, generado por búsqueda sin ajustes, nos da un **60,7 %** de precisión con la última distribución. Sus parámetros son algoritmo automático,

	D1	D2	D3	D4	D5	Media	Desviación típica
Sin ajustes	0.510	0.491	0.550	0.522	**0.607**	0.536	0.045
Grid search	0.526	0.506	0.577	0.522	**0.602**	0.547	0.041

Tabla 5: Valores de precisión para la ejecución 1 de KNN en ambos sexos.

metrica Minkowski con parámetro p igual a 2 (por tanto equivalente a métrica euclídea), peso uniforme, tamaño de hoja 30 y 5 vecinos a tener en cuenta. Encontramos sobreajuste igual que hacíamos en el conjunto de hombres, con medias del **74 %** y **95 %** de precisión para cálculo sin y con búsqueda.

5.2. Ejecución 2

Tras esta primera ejecución, se pasa a dar valores a los parámetros de **número de vecinos** y **tamaño de hoja**. La cantidad idónea del primero va a depender del tamaño de nuestros datos, que como ya se ha comentado varía mucho. Lo único que debemos tener en cuenta es que este número debe de ser **impar**, para que los votos siempre den un resultado y no acaben en empate. Para cada conjunto se elegirá una K igual a la **raíz cuadrada del número total de sujetos** (siempre que esta no sea par, en cuyo caso se le sumará 1) para el modelo sin búsqueda; y en el caso de los modelos con búsqueda se escogerá el número impar directamente superior e inferior para crear un rango.

En el caso del tamaño de hoja, tras realizar pruebas con los tres conjuntos de datos, se ha concluido que alterar su valor no cambia los resultados de precisión obtenidos, pero si los **tiempos** en los que se consiguen. Por tanto realizaremos pruebas para los datos de mujeres y hombres en busca de la mejor reducción de tiempo, aplicando el resultado del segundo conjunto también al de ambos sexos para ahorrar tiempo. Con esta cifra pretendemos **compensar en parte la subida de tiempo de ejecución causada por el aumento de número de vecinos**.

5.2.1. Resultados en mujeres

Gracias a la pequeña cantidad de datos en mujeres, hemos podido realizar más pruebas para encontrar un buen **tamaño de hoja**, pues los resultados se obtienen más rápidamente. Se han usado valores entre 5 y 140 para el tamaño de hoja, concluyéndose así que la mejor

cifra es 120.

Tamaño de hoja	Tiempo de ejecución (s)
10	72
20	58
30	48
40	48
50	41
60	40
70	42
90	43
100	36
120	**36**
140	37

Tabla 6: Tiempos de ejecución de grid search de 72 combinaciones de KNN en mujeres con varios tamaños de hoja.

Aunque como se ve en la **Tabla 6** la diferencia es de segundos, estos se van acumulando y acaban suponiendo un ahorro de tiempo alto.

Respecto al **número de vecinos**, para **modelos con búsqueda** se añade una entrada al diccionario de hiperparámetros, permitiendo los valores 15, 17, 19 y 21 (se ha tomado un valor más aprovechando la baja cantidad de muestras, que acelera mucho la ejecución). Estos valores han sido elegidos porque tenemos un total de 297 conectomas de mujeres, y la raíz de este número es 17. Por ello se orbita alrededor de este número. En el modelo sin búsqueda se establece 17 como número de vecinos.

La ejecución, cuyos resultados se muestran en la **Tabla 7** y la **Figura 18**, ha requerido solamente de 3 minutos y 43 segundos.

Las **medias han mejorando** en tres y cuatro puntos cada una, superándose el **60 %** de precisión en ambas. La desviación típica no ha cambiado demasiado en ningún método, aunque destaque el resultado de la quinta división en ambos, por tener un valor de precisión mucho mejor, **74,6 %**. Este **mejor modelo** tiene los parámetros algoritmo *ball tree*, tamaño de hoja 120, métrica euclídea, 17 vecinos y pesos uniformes. Las medias de los conjuntos de entrenamiento

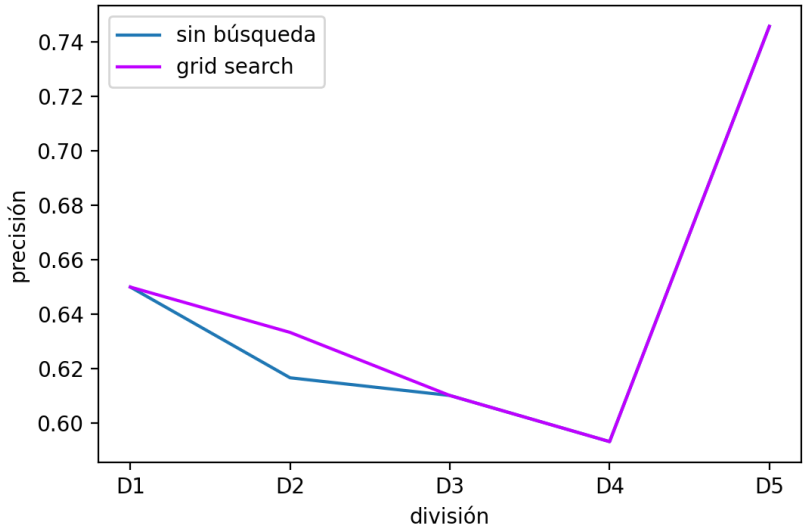

Figura 18: Valores de precisión para la ejecución 2 de KNN en mujeres.

	D1	D2	D3	D4	D5	Media	Desviación típica
Sin búsqueda	0.650	0.617	0.610	0.593	**0.746**	0.643	0.061
Grid search	0.650	0.633	0.610	0.593	**0.746**	0.646	0.060

Tabla 7: Valores de precisión para la ejecución 2 de KNN en mujeres.

son un **73 %** y un **72 %,** por lo que se ha reducido algo el sobre ajuste y hemos mejorado los resultados.

5.2.2. Resultados en hombres

Para ajustar el **tamaño de hoja** del conjunto de datos de hombres se pudieron hacer menos pruebas debido a que estas tardaban mucho más. No obstante, se consiguió reducir el tiempo requerido para ejecutar un *grid search* de 72 candidatos, pasando de 22 minutos y medio con 30 hojas a 17 minutos y 18 segundos con 300 (**Tabla 8**).

Ahora, la diferencia de tiempo es de alrededor de 5 minutos por búsqueda, que al realizar las cinco búsquedas que supone cada ejecución darán aproximadamente unos 25 minutos de ahorro.

Pasando al número de vecinos, la cantidad total de imágenes de hombre es 1715, así que establecemos valores de 39, 41 y 43 para el parámetro **n_neighbors**. Para el cálculo sin búsqueda

Tamaño de hoja	Tiempo de ejecución (min)
30	22,5
150	18,0
200	17,9
250	17,9
300	**17,3**
350	17,4

Tabla 8: Tiempos de ejecución de grid search de 72 combinaciones de KNN en hombres con varios tamaños de hoja.

usaremos 41 vecinos. Esta segunda ejecución tarda 2 horas y cuarto en completarse.

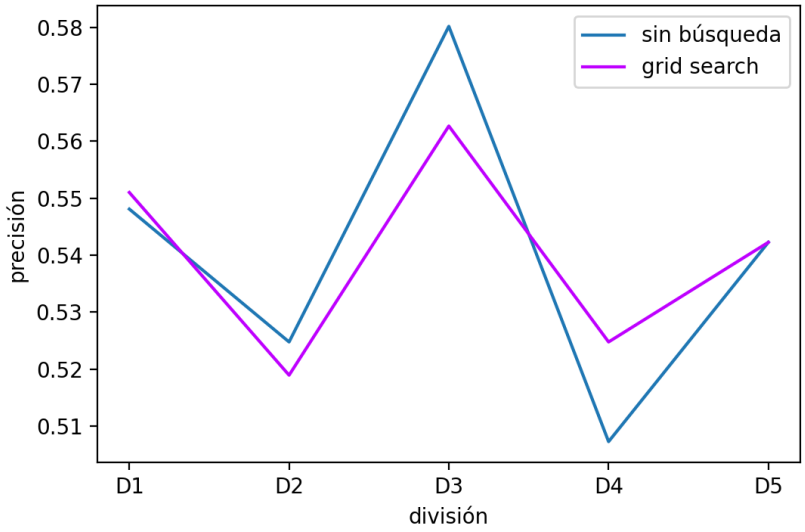

Figura 19: Valores de precisión para la ejecución 2 de KNN en hombres.

	D1	D2	D3	D4	D5	Media	Desviación típica
Sin búsqueda	0.548	0.525	**0.580**	0.507	0.542	0.541	0.027
Grid search	0.551	0.499	0.569	**0.574**	0.557	0.540	0.018

Tabla 9: Valores de precisión para la ejecución 2 de KNN en hombres.

En la **Tabla 9** y la **Figura 19** apreciamos que la precisión media prácticamente no ha variado, sólo en los modelos sin búsqueda conseguimos aumentarla en un 1 %. El **mejor modelo** que

obtenemos da prácticamente la misma precisión, incluso menos, un **58 %**, que el mejor obtenido en la primera ejecución. Usa algoritmo automático, tamaño de hoja 300, métrica Minkowski, 41 vecinos y pesos uniformes.

Esta segunda ejecución no ha mejorado prácticamente nada respecto a la primera, aumentándose el sobreajuste, pues los modelos en entrenamiento consiguen medias del **70 %** en cálculo sin búsqueda y del **99 %** en cálculo con *grid search*.

5.2.3. Resultados en ambos sexos

Con el conjunto de datos de ambos sexos hemos repetido los mismos parámetros que los usados para hombres, consiguiéndose una mejora de unos 4 minutos por búsqueda, que ahora requiere 24 minutos. Usamos de nuevo 300 como tamaño de hoja, y como tenemos un total de 2012 sujetos, establecemos 45 como número de vecinos en modelos sin búsqueda y el trío 43, 45, 47 para modelos con búsqueda. La ejecución completa del programa tardó 3 horas y 20 minutos, mejorando muy poco la anterior.

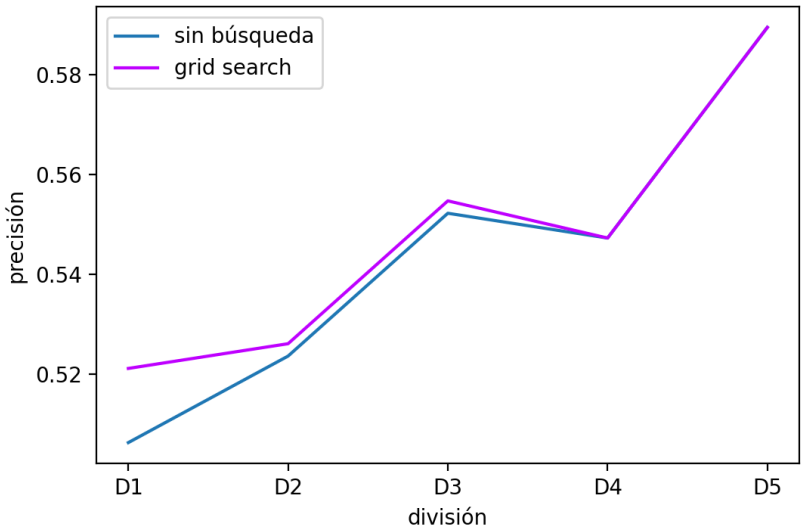

Figura 20: Valores de precisión para la ejecución 2 de KNN en ambos sexos.

Las medias no han mejorado demasiado, y se ha reducido la desviación típica, obteniéndose una mejor precisión menor en ambos cálculos, **59 %**. Este **modelo** emplea algoritmo de fuerza bruta, tamaño de hoja 300, métrica euclídea, 45 vecinos y pesos acordes a la distancia. Teniendo en cuenta que la ejecución ha requerido el doble de tiempo y hemos realizado más ajustes, los

	D1	D2	D3	D4	D5	Media	Desviación típica
Sin búsqueda	0.506	0.526	0.552	0.547	**0.590**	0.544	0.032
Grid search	0.521	0.526	0.555	0.547	**0.590**	0.548	0.027

Tabla 10: Valores de precisión para la ejecución 2 de KNN en ambos sexos.

resultados no son lo esperados, al igual que ocurría en hombres. El sobreajuste ha descendido en los modelos sin búsqueda, que tienen una media del **65 %** de precisión, pero han aumentando en los modelos con búsqueda *grid search*, pues su media es un **100 %** de precisión.

Este sobreajuste podría explicar la falta de mejora tanto en hombres como en ambos sexos, pues los modelos han sido demasiado optimizados hacia los datos de entrenamiento, sin tener capacidad de adaptarse a otros diferentes.

5.3. Conclusiones finales

Tras ambas ejecuciones hemos podido determinar los mejores modelos del algoritmo de los vecinos más cercanos para cada uno de los tres conjuntos de datos. Los parámetros explorados han sido los siguientes:

- **Número de vecinos** (*n_neighbors*): para mujeres, 5, 17, 19 o 21; para hombres 39, 41 o 43; y para ambos sexos 43, 45 o 47. Las diferencias se deben a que tenemos más datos de hombres que de mujeres. Para establecer ambos rangos se han llevado a cabo pruebas, y tanto en el caso de las mujeres como de los hombres, valores menores o mayores comenzaban a dar valores de exactitud peores. Todos los números son impares ya que, como se comentó en la sección anterior, esto nos evita tener un empate entre las clases.
- **Función de peso** *weight*: los pesos pordrán ser iguales para todos (uniform) o mayores cuanto más cerca estén (distance).
- **Algoritmo de cálculo de los vecinos más cercanos** (*algorithm*): las opciones son fuerza bruta (*brute*), *ball tree* (*ball_tree*) y árbol k dimensional (*kd_tree*).
- **Métrica de distancia** (*metric*): euclídea (*euclidean*), Manhattan (*manhattan*), de Chebyshev (*chebyshev*) y de Minkowski (*minkowski*).
- **Tamaño de hoja**: 120 para mujeres y 300 para hombres y ambos sexos.

Con ellos, hemos seleccionado los siguientes modelos como mejores:

- En **mujeres**, algoritmo *ball tree*, métrica euclídea, pesos uniformes, tamaño de hoja 120 y 17 vecinos a tener en cuenta, lo que nos da una precisión del **74,6 %**.
- En **hombres**, algoritmo de *ball tree*, métrica euclídea, peso dependiente de la distancia, tamaño de hoja 30 y 5 vecinos a tener en cuenta, lo que nos da una precisión del **58,3 %**.
- En **ambos sexos**, algoritmo automático, métrica Minkowski con parámetro p igual a 2 (por tanto equivalente a métrica euclídea), peso uniforme, tamaño de hoja 30 y 5 vecinos a tener en cuenta, lo que nos da una precisión del **60,7 %**.

La mejor precisión se ha obtenido claramente en mujeres, superando esta en dieciséis puntos a la obtenida con hombres y en casi catorce a la obtenida para ambos. En estos dos grupos no hemos conseguido mejorar los modelos, quedándonos con precisiones algo bajas a pesar de realizar búsqueda por *grid search*.

Destaca también que en todas las ocasiones los modelos encontrados con y sin búsqueda dan resultados muy parecidos, algunos incluso mejores, que los encontrados con búsqueda. El hecho de que los primeros se puedan calcular en mucho menos tiempo que los segundos nos indica que en este conjunto de datos no vale la pena realizar búsquedas para crear modelos de KNN, pues los hallados sin usar estos métodos tienen la misma precisión.

6

Árboles de decisión

Los árboles de decisión del paquete *scikit-learn* tienen 13 parámetros, pero no todos se incluyen en el diccionario de hiperparámetros. Para poder tener resultados reproducibles, el valor de *random_state*, será siempre 3, en modelos creados sin y con búsqueda. Además, no usaremos *ccp_alpha*, pues este se orienta a reducir los costes asociados con complejidad. Respecto al resto de parámetros, en un principio daremos valores al **criterio de calidad, estrategia de división, y profundidad máxima del árbol**, para luego ajustar el resto de parámetros.

Usaremos los conectomas de mujeres para realizar el mayor número de pruebas, cuyos resultados luego trasladaremos a los demás conjuntos. Esto se debe a que nos es imposible realizar tantas ejecuciones en los conjuntos de hombres y ambos sexos, pues cada una requiere demasiado tiempo.

También para ahorrar tiempo se empleará el **método de búsqueda aleatoria**, porque las posibles combinaciones de los diccionarios serán muchas. Por ello se realizarán 460 iteraciones de búsqueda aleatoria para garantizar una buena calidad. Además, en las ocasiones en las que el número de combinaciones posibles no llegue a 460, la búsqueda aleatoria actuará como búsqueda con *grid search*.

6.1. Conectomas de mujeres

6.1.1. Ejecución 1

Para la primera ejecución, usaremos los siguientes parámetros:

- **Medida de calidad de división** (*criterion*), que podrá ser impureza Gini (*gini*) ganancia de información (*entropy*).
- **Estrategia de división** (*splitter*), que podrá ser elegir la mejor partición (*best*) o una

aleatoria (*random*).

- **Profundidad máxima del árbol** (*max_depth*), que para los modelos con búsqueda tendrá disponibles los valores 1, 6, 11 y 16, y para modelos sin búsqueda será 8.

Con esta configuración, obtenemos los resultados ilustrados en la **Tabla 11** y la **Figura 21**, con un tiempo de ejecución de 2 minutos y medio.

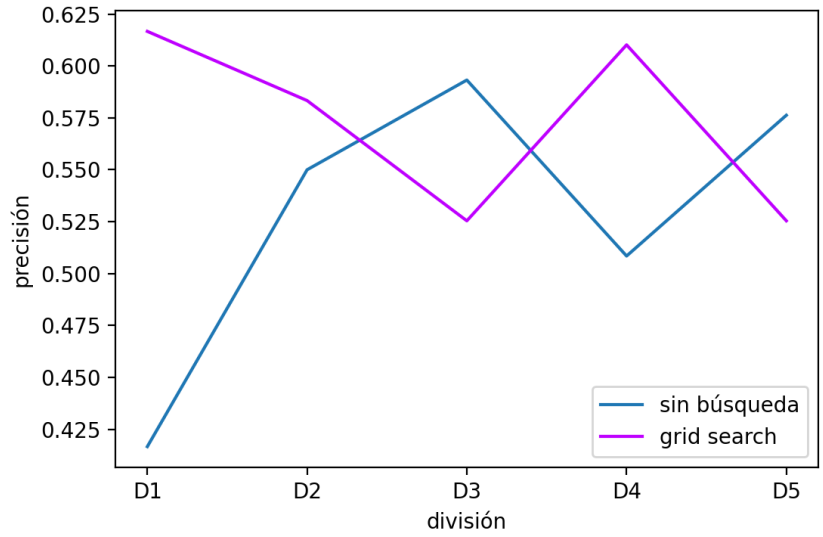

Figura 21: Valores de precisión para la ejecución 1 de DT en mujeres.

	D1	D2	D3	D4	D5	Media	Desviación típica
Sin ajustes	0.417	0.550	**0.593**	0.508	0.576	0.528	0.080
Búsq. aleatoria	**0.617**	0.583	0.525	0.610	0.525	0.572	0.044

Tabla 11: Valores de precisión para la ejecución 1 de DT en mujeres.

Esta primera ejecución nos ha dado unas medias de precisión no muy buenas, **53 %** y **57 %**, obteniéndose la **máxima precisión** en búsqueda aleatoria, con un modelo de árbol de decisión que mide la calidad de una división según la ganancia de información, usa estrategia de división aleatoria y tiene una profundidad de 1.

La precisión media para el conjunto de entrenamiento es de un **99,8 %** en el caso de los modelos sin ajustes y de un **86,2 %** para los otros, por lo que tenemos **mucho sobreajuste**. En las siguientes ejecuciones intentaremos reducirlo.

6.1.2. Ejecución 2

Ahora procederemos a buscar los **mejores rangos de los hiperparámetros restantes** para árboles de decisión en mujeres. Para ello llevaremos a cabo una serie de pruebas, añadiendo un parámetro cada vez, y determinando su mejor configuración fijándonos en la precisión media obtenida. Esta sección, por tanto, realmente no es una sola ejecución, sino varias cortas para poder hacer pruebas seguidas de una final.

Cambiaremos también la profundidad máxima en modelos sin ajustes a 11, para intentar mejorar sus resultados.

Tras realizar pruebas para todos los hiperparámetros, se ha concluido que sus mejores rangos son los siguientes:

- **Profundidad máxima** (*max_depth*): se establecen valores entre 1 y 11.
- **Características a tener en cuenta** (*max_features*): toma tanto el logaritmo en base 2 y la raíz cuadrada de la cantidad total de características como los valores del 10 % al 50 %, de 10 % en 10 %.
- **Reducción mínima de impureza** (*min_impurity_decrease*): establecemos valores entre el 1 % y el 5 % de impureza, con saltos de un 1 %.
- **Individuos mínimos para realizar una división** (*min_samples_split*): serán entre 2 y 18 individuos con saltos de 2.
- **Número máximo de nodos hoja** (*max_leaf_nodes*): entre 60 y 90, de 10 en 10.

Con este diccionario de hiperparámetros realizamos una ejecución final de 15 minutos, con la que conseguimos mejorar los resultados, como se aprecia en la **Tabla 12** y la **Figura 22**.

	D1	D2	D3	D4	D5	Media	Desviación típica
Sin ajustes	0.533	0.617	0.576	0.525	**0.695**	0.589	0.069
Búsq. aleatoria	0.750	**0.667**	0.695	0.593	0.475	0.636	0.106

Tabla 12: Valores de precisión para la ejecución 2 de DT en mujeres.

Las **medias han subido** en ambos casos, aunque también la desviación típica. Ahora las medias de modelos sin ajustes y con búsqueda aleatoria son, respectivamente, **58,9 %** y **63,6 %**,

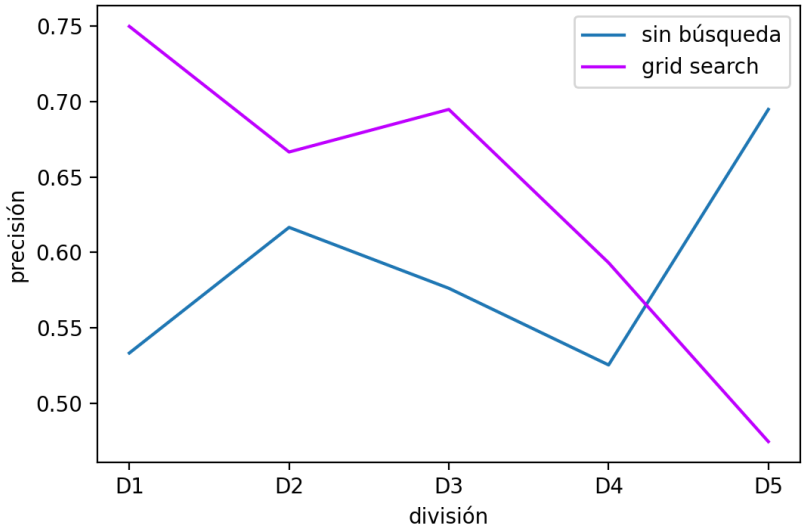

Figura 22: Valores de precisión para la ejecución 2 de DT en mujeres.

siendo el **mejor modelo** el obtenido con el nuevo diccionario de hiperpaŕametros, en concreto con la combinación de un criterio de calidad de impureza Gini, división aleatoria, profundidad máxima 8, 6 muestras mínimas para dividir, reducción mínima de impureza 0,02, máximo de 80 nodos hoja y un 40 % de características a tener en cuenta. Este nos da una precisión del **75 %**.

El sobreajuste ha aumentado en los modelos sin búsqueda, siendo la precisión media de estos en datos de entrenamiento un **100 %**, y se ha reducido levemente en modelos con búsqueda aleatoria, pasando a ser del **81 %**.

Ahora que hemos hecho pruebas en el conjunto más pequeño, vamos a trasladar los mejores ajustes a los conjuntos de hombres y ambos sexos.

6.2. Conectomas de hombres

6.2.1. Ejecución 1

En vez de partir de cero, vamos a probar en hombres la misma configuración de la segunda ejecución de árboles de decisión para mujeres, solo que **aumentando la profundidad del árbol**, ya que nuestro conjunto de datos es mayor. En la nueva configuración, esta pasa a tener un rango de 1 a 21, saltando de dos en dos. Ejecutarla nos lleva 1 hora y 3 minutos,

obteniendo así los resultados ilustrados en la **Tabla 13** y la **Figura 23**.

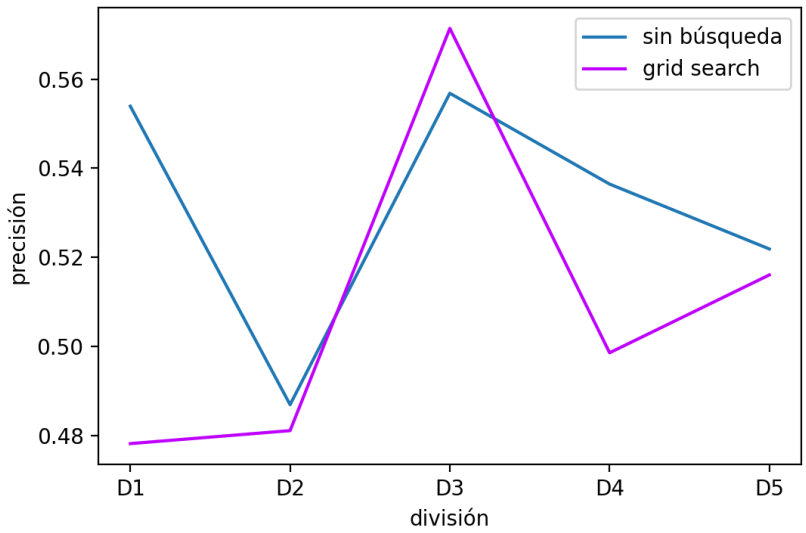

Figura 23: Valores de precisión para la ejecución 1 de DT en hombres.

	D1	D2	D3	D4	D5	Media	Desviación típica
Sin ajustes	0.554	0.487	**0.557**	0.536	0.522	0.531	0.029
Búsq. aleatoria	0.478	0.481	**0.571**	0.499	0.516	0.509	0.038

Tabla 13: Valores de precisión para la ejecución 1 de DT en hombres.

Las medias, un **53,1 %** de precisión para modelos sin ajustes y un **50,9 %** modelos con búsqueda aleatoria, no son buenas, ni varían mucho de las algoritmo de los k vecinos más cercanos. El **mejor modelo** lo encontramos en la tercera división, donde la búsqueda aleatoria ha encontrado la configuración de medida de mejora según ganancia de información, uso de la mejor división posible, profundidad máxima de 5, mínimo de 16 muestras y mejora de impureza del 1 % para dividir, 10 % de características a tener en cuenta y 70 nodos hoja como máximo. Su precisión es del **57,1 %**, menor que la que habíamos encontrado para KNN.

La desviación no es demasiado grande en ningún caso, pero sí que tenemos sobreajuste, pues los modelos sin búsqueda dan una media del **99 %** de precisión y los modelos con búsqueda aleatoria una del **72 %**.

6.2.2. Ejecución 2

Como los modelos obtenidos en la ejecución 1 no empleaban una profundidad de más de 10, volvemos a establecer el rango anterior, de 1 a 11. Para intentar mejorar nuestros modelos, **cambiamos el mínimo de muestras para dividir** a un rango de entre 45 y 99, con saltos de 2 en 2. Este rango se decide después de hacer pruebas con otros menores, que no mejoran el resultado. Cambiamos este parámetro y no otro porque al tener más individuos, establecer un mínimo de muestras muy pequeño para dividir un nodo puede estar causando parte del sobreajuste.

Con esta configuración, ejecutamos durante un total de 53 minutos y obtenemos los resultados algo mejores (**Tabla 14**, **Figura 24**):

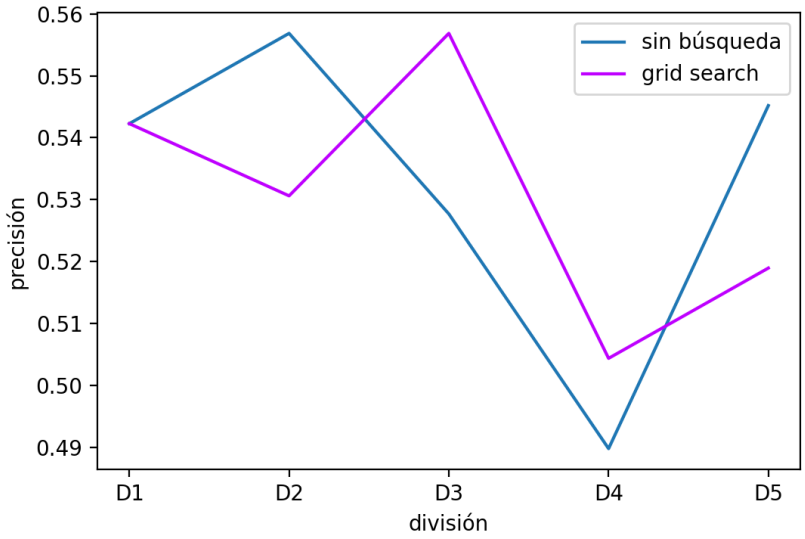

Figura 24: Valores de precisión para la ejecución 2 de DT en hombres.

	D1	D2	D3	D4	D5	Media	Desviación típica
Sin ajustes	0.542	**0.557**	0.528	0.490	0.545	0.532	0.026
Búsq. aleatoria	0.542	0.531	**0.557**	0.504	0.519	0.531	0.020

Tabla 14: Valores de precisión para la ejecución 2 de DT en hombres.

La media de los modelos sin ajustes se ha mantenido, porque no hemos cambiado sus parámetros, pero la media de los modelos con búsqueda aleatoria ha subido dos puntos, pasando a

ser un **53,1 %** y **51 %**. El **mejor modelo** lo encontramos de nuevo en la segunda división, pero da una precisión menor que el que encontrábamos anteriormente, **55,7 %**. Esta nueva configuración nos ha dado una media algo mejor porque sus precisiones son todas parecidas, con la mitad de desviación típica que la ejecución anterior.

Hemos conseguido reducir el sobreajuste, pues ahora la precisión media para modelos de búsqueda aleatoria es del **67,9 %**.

6.3. Conectomas de ambos sexos

Como tanto en hombres como en mujeres ha sido mejor tener una profundidad de entre 1 y 11, esta vez no la cambiaremos. Lo que sí modificaremos será el mínimo de muestras para dividir un nodo, que tomarán valores entre 15 y 43, de dos en dos. Los modelos sin ajustes seguirán haciéndose de la misma manera. Tras 1 hora y 10 minutos, obtenemos los resultados descritos en la **Tabla 15** y la **25**.

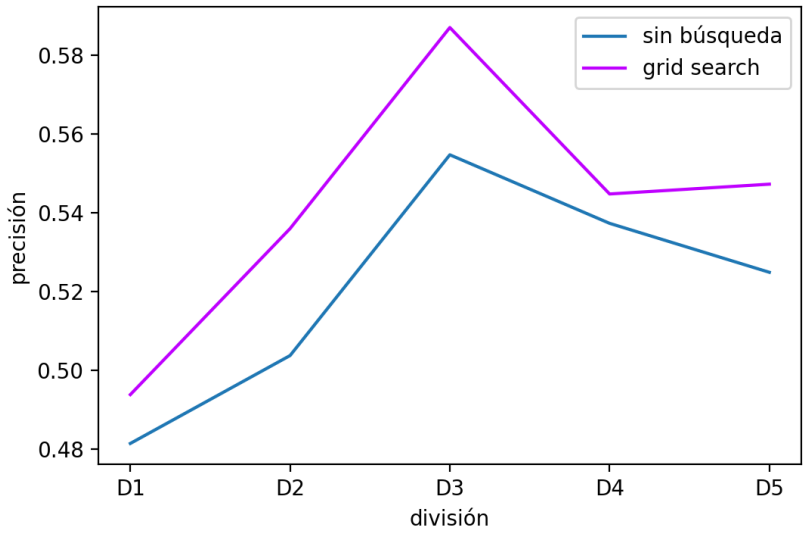

Figura 25: Valores de precisión para la ejecución 1 de DT en ambos sexos.

Obtenemos, una vez más, medias de precisión no muy buenas, pues tenemos un **52 %** para modelos sin ajuste y un **54,7 %** para modelos con búsqueda. El mejor modelo tiene una precisión del **58,7 %**, con medida de mejora según ganancia de información, uso de la mejor división posible, profundidad máxima de 4, mínimo de 21 muestras y mejora de impureza del 1 % para dividir, 50 % de características a tener en cuenta y 80 nodos hoja como máximo.

	D1	D2	D3	D4	D5	Media	Desviación típica
Sin ajustes	0.481	0.504	**0.555**	0.537	0.525	0.520	0.029
Búsq. aleatoria	0.494	0.536	**0.587**	0.545	0.547	0.542	0.033

Tabla 15: Valores de precisión para la ejecución 1 de DT en ambos sexos.

Volvemos a tener el mismo sobreajuste, un **98,4 %** de precisión en modelos sin ajuste y un **65,2 %** sobre datos de train.

6.4. Conclusiones finales

Los mejores modelos de árboles de decisión para los tres conjuntos de datos han necesitado diccionarios parecidos. En este caso, **ninguno de ellos viene de creación de modelos normal**, sino que todos han surgido de búsqueda aleatoria. Los parámetros explorados han sido los siguientes:

- **Medida de calidad de división** (*criterion*): las opciones dadas son por ganancia de información (*entropy*) o según impureza Gini (*gini*).
- **Estrategia de división** (*spliter*): puede ser ejor partición (*best*) o una aleatoria (*random*).
- **Profundidad máxima del árbol** (*max_depth*): tendrá disponibles valores del 1 al 11.
- **Características a tener en cuenta para dividir** (*max_features*): toma tanto el logaritmo en base 2 y la raíz cuadrada de la cantidad total de características como los valores del 10 % al 50 %, de 10 % en 10 %.
- **Decrecimiento mínimo de impureza** (*min_impurity_decrease*): valores entre el 1 % y el 5 % de impureza.
- **Individuos mínimos para realizar una división** (*min_samples_split*): para mujeres, entre 2 y 18, para hombres entre 45 y 100 y para ambos sexos entre 15 y 45, todos con saltos de 2.
- **Número máximo de nodos hoja** (*max_leaf_nodes*): entre 60 y 90, de 10 en 10.

Con ellos, hemos seleccionado los siguientes modelos como mejores:

- **En mujeres**, criterio de calidad de impureza Gini, división aleatoria, profundidad máxima 8, 6 muestras mínimas para dividir, reducción mínima de impureza del 2 %, máximo de 80 nodos hoja y un 40 % de características a tener en cuenta, lo que nos da una precisión del **75 %**.

- **En hombres**, medida de mejora según ganancia de información, uso de la mejor división posible, profundidad máxima de 5, mínimo de 16 muestras y mejora de impureza del 1 % para dividir, 10 % de características a tener en cuenta y 70 nodos hoja como máximo, lo que nos da una precisión del **57,1 %**.

- **En ambos sexos**, medida de mejora según ganancia de información, uso de la mejor división posible, profundidad máxima de 4, mínimo de 21 muestras y mejora de impureza del 1 % para dividir, 50 % de características a tener en cuenta y 80 nodos hoja como máximo, lo que nos da una precisión del **58,7 %**.

Volvemos a obtener la **mejor precisión en mujeres**, que además es el único conjunto de datos que ha visto mejora en su mejor modelo, aunque esta sea solamente del 0,4 %. Los modelos para hombres y ambos sexos siguen obteniendo valores de precisión muy por debajo de los de mujeres, aunque se haya realizado búsqueda aleatoria.

De nuevo, los valores de precisión de los modelos obtenidos mediante ambas estrategias son muy parecidos, prácticamente sin compensar la gran diferencia de tiempo necesaria para calcularlos por búsqueda.

7

Random forests

Para *random forest* repetiremos el método llevado a cabo en el capítulo anterior: hacer la mayoría de las pruebas sobre el conjunto de datos de mujeres, usando **búsqueda aleatoria** con 460 iteraciones. De los parámetros disponibles para el algoritmo, dejaremos en predeterminado *bootstrap, ccp_alpha, n_jobs* y *verbose*, y estableceremos en todas las ejecuciones un *random_state* de 3 (tanto para la búsqueda como para el modelo), para poder reproducir los resultados, y el parámetro *oob_score* activado, para tener el menor sobreajuste posible. En **modelos sin búsqueda**, estableceremos una profundidad máxima de 11.

7.1. Conectomas de mujeres

7.1.1. Ejecución 1

Para la primera ejecución, usaremos los siguientes parámetros:

- **Número de árboles** (*n_estimators*): entre 50 y 100, con saltos de 10 en 10.
- **Profundidad máxima** (*max_depth*): entre 2 y 20.
- **Medida de calidad de división** (*criterion*): podrá ser o medida de impureza de Gini (*gini*) o por ganancia de información (*entropy*).
- **Características a tener en cuenta** (*max_features*): Podrá ser o bien el logaritmo en base dos, la raíz cuadrada, o el 1 % de las características totales.

Esta ejecución requiere una hora y media, dando los resultados disponibles en la **Tabla** y la **Figura** .

Las primeras medias que obtenemos son mejores que las que habíamos obtenido con los modelos anteriores, un **71,7 %** y un **67,4 %** de precisión para modelos sin y con búsqueda.

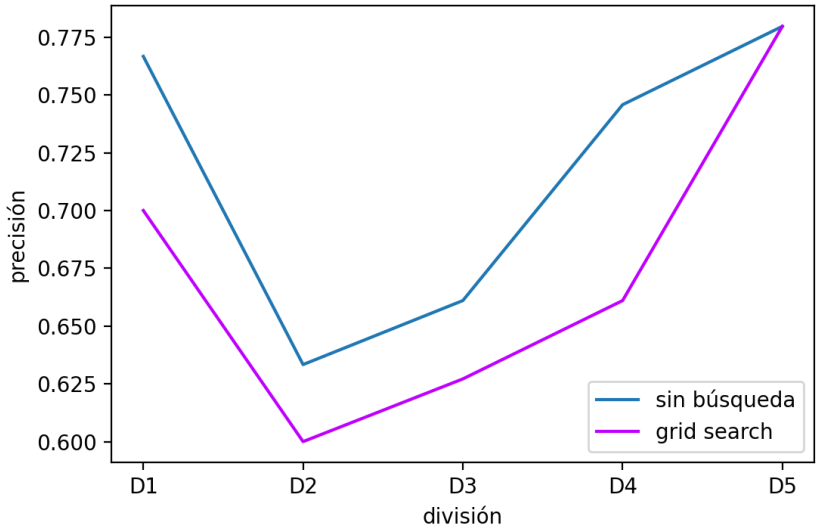

Figura 26: Valores de precisión para la ejecución 1 de RF en mujeres.

	D1	D2	D3	D4	D5	Media	Desviación típica
Sin ajustes	0.767	0.633	0.661	0.746	**0.780**	0.717	0.066
Búsq. aleatoria	0.700	0.600	0.627	0.661	**0.780**	0.674	0.070

Tabla 16: Valores de precisión para la ejecución 1 de RF en mujeres.

El **mejor modelo** lo obtenemos en la quinta división, con una precisión del **78 %** en ambas configuraciones. Este modelo usa 70 árboles, calcula la cantidad de características a tener en cuenta usando la raíz cuadrada, tiene una profundidad de 8, y usa Gini como medida de calidad de división.

Vuelve a existir **sobreajuste**, con un **100 %** de precisión de media sobre los conjuntos de entrenamiento para ambos métodos.

7.1.2. Ejecución 2

En la segunda ejecución, pasamos a usar los valores de parámetros que habíamos obtenido para árboles de decisión, así como los particulares del algoritmo *random forest*. Por tanto, usaremos:

- **Número de árboles** (*n_estimators*): entre 100 y 200, con saltos de 10 en 10.

- **Medida de calidad de división** (*criterion*), que podrá ser impureza Gini (*gini*) ganancia de información (*entropy*).
- **Estrategia de división** (*splitter*), que podrá ser elegir la mejor partición (*best*) o una aleatoria (*random*).
- **Profundidad máxima del árbol** (*max_depth*): se establecen valores entre 1 y 11.
- **Características a tener en cuenta** (*max_features*): toma tanto el logaritmo en base 2 y la raíz cuadrada de la cantidad total de características como los valores del 10 % al 50 %, de 10 % en 10 %.
- **Reducción mínima de impureza** (*min_impurity_decrease*): establecemos valores entre el 1 % y el 5 % de impureza, con saltos de un 1 %.
- **Individuos mínimos para realizar una división** (*min_samples_split*): serán entre 2 y 18 individuos con saltos de 2.
- **Número máximo de nodos hoja** (*max_leaf_nodes*): entre 60 y 90, de 10 en 10.
- **Reutilización de la solución anterior** (*warm_start*): Podrá o no usarse.

Además, vamos a usar el **número de muestras** (*max_samples*) para *bootstrap* recomendado, 17 (la raíz cuadrada del total de muestras, para intentar reducir el sobreajuste. Los parámetros no mencionados se quedarán en sus valores predeterminados.

La ejecución con esta configuración requiere de dos horas y media, y da los resultados ilustrados en la **Tabla** y la **Figura** . Como los modelos sin búsqueda siguen teniendo los mismos parámetros, mostraremos sus resultados pero no los discutiremos.

	D1	D2	D3	D4	D5	Media	Desviación típica
Sin ajustes	0.767	0.633	0.661	0.746	**0.780**	0.717	0.066
Búsq. aleatoria	**0.667**	0.583	0.627	0.627	0.661	0.633	0.033

Tabla 17: Valores de precisión para la ejecución 2 de RF en mujeres.

La media de búsqueda aleatoria ha empeorado en cuatro puntos, teniendo ahora su **mejor modelo** solamente un **66,7 %** de precisión. Se ha conseguido reducir el sobreajuste, pues ahora

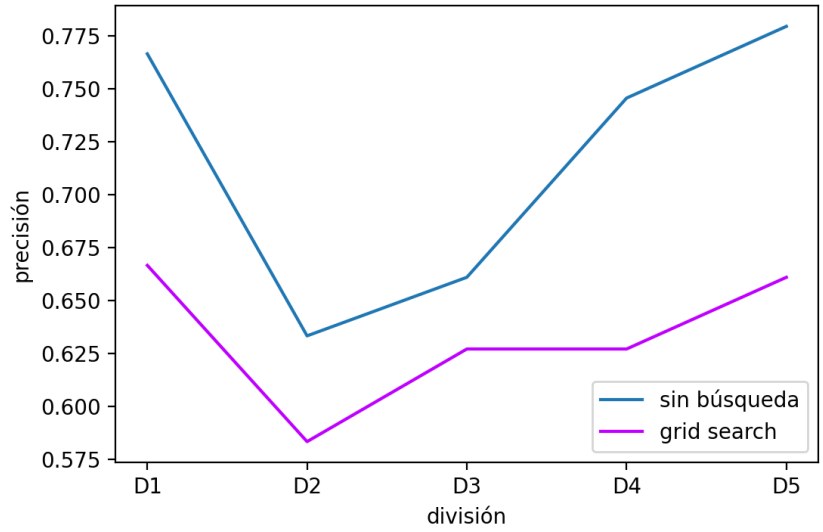

Figura 27: Valores de precisión para la ejecución 2 de RF en mujeres.

la precisión media para entrenamiento es un **75,8 %**, pero esto ha sido a cambio de empeorar también los resultados sobre los datos de prueba, por lo que no nos compensa.

7.1.3. Ejecución 3

Como al añadir parámetros hemos obtenido peores resultados, vamos a intentar volver a los parámetros anteriores, pero añadiéndoles el número de muestras para *bootstrap* de 17, para intentar mejorar el modelo reduciendo el sobreajuste. De nuevo, no comentaremos los resultados de los modelos creados sin ajuste porque no han cambiado.

Los modelos tardan 40 minutos en completarse, dando los siguientes resultados (**Tabla** y *Figura*):

	D1	D2	D3	D4	D5	Media	Desviación típica
Sin ajustes	0.767	0.633	0.661	0.746	**0.780**	0.717	0.066
Búsq. aleatoria	0.683	0.617	0.593	0.678	**0.712**	0.657	0.050

Tabla 18: Valores de precisión para la ejecución 3 de RF en mujeres.

Los resultados han mejorado algo, pues la precisión media es ahora del **65,7 %**, pero no hemos conseguido mejorar la precisión de la primera ejecución, ni su mejor modelo. El sobreajuste se ha reducido sobre esta primera ejecución, siendo ahora la media sobre datos de

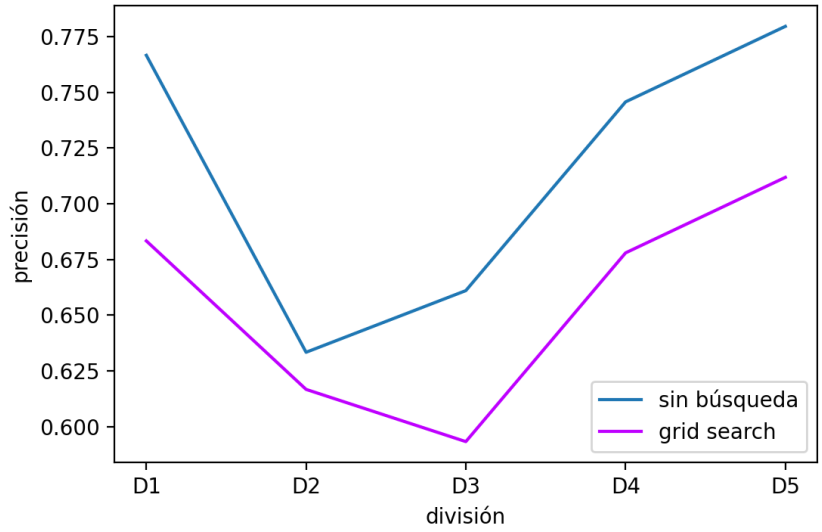

Figura 28: Valores de precisión para la ejecución 3 de RF en mujeres.

entrenamiento del **78,4 %**. Pero, de nuevo, si esta reducción es a cambio de menor precisión en el conjunto de prueba, no es útil.

7.2. Conectomas de hombres

Como se ha comentado antes, en hombres reutilizaremos los parámetros encontrados para mujeres.

7.2.1. Ejecución 1

Como hemos visto que para los datos de mujeres lo mejor es usar pocos parámetros, para esta primera ejecución aplicaremos la misma configuración que usada en la primera ejecución para mujeres.

Esta ejecución necesita 10 horas y 45 minutos, mucho más que las anteriores, y una vez más, da precisiones peores que las del conjunto de mujeres (**Tabla** y *Figura*).

	D1	D2	D3	D4	D5	Media	Desviación típica
Sin ajustes	0.557	0.557	**0.641**	0.56	0.563	0.576	0.037
Búsq. aleatoria	0.577	0.557	**0.601**	0.525	0.539	0.560	0.030

Tabla 19: Valores de precisión para la ejecución 1 de RF en hombres.

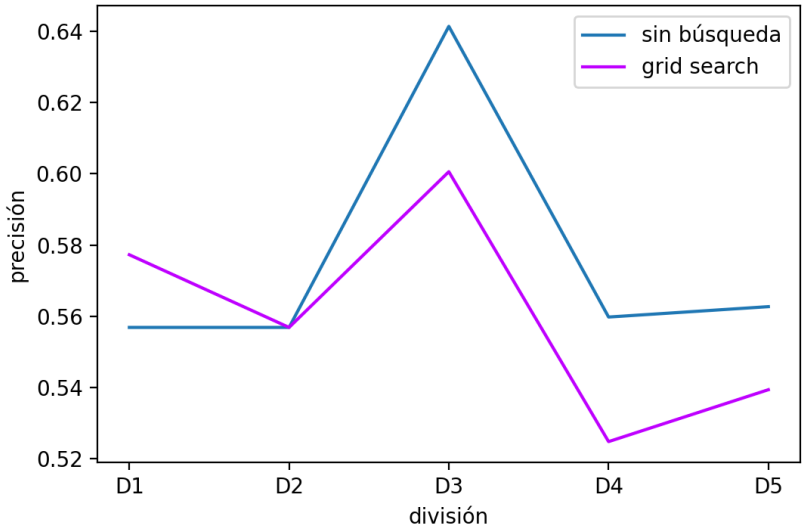

Figura 29: Valores de precisión para la ejecución 1 de RF en hombres.

Obtenemos una media de precisión del **55,1 %** para modelos sin ajustes y del **56 %** para modelos con búsqueda, siendo el **mejor modelo** el creado en la tercera división de los datos, con un **64,1 %** de precisión en el modelo sin búsqueda. Este usa criterio de calidad Gini, ninguna profundidad máxima, 100 árboles, y calcula la cantidad de características usadas automáticamente.

Las medias de precisión en datos de entrenamiento son mayores del **99 %** en ambos conjuntos, por lo que nos volvemos a encontrar con sobreajuste.

7.2.2. Ejecución 2

Aunque hemos conseguido valores buenos de precisión (si los comparamos con los que hemos obtenido para hombres en otros modelos), vamos a probar a reducir su sobreajuste con el mismo método que antes. Emplearemos el mismo diccionario de parámetros que el comentado en la segunda ejecución para mujeres, y usaremos un máximo de muestras para *bootstrap* igual a la raíz de la cantidad de datos que tenemos, es decir, 41.

Con esta configuración, la ejecución tarda 13 horas y no conseguimos mejorar los resultados, como se aprecia en lo **Tabla** y la **Figura** .

La media para búsqueda aleatoria ha empeorado, y ahora ninguno de sus modelos llega al **60 %**. Se ha reducido el sobreajuste, con una precisión para datos de entrenamiento del **67,4 %**,

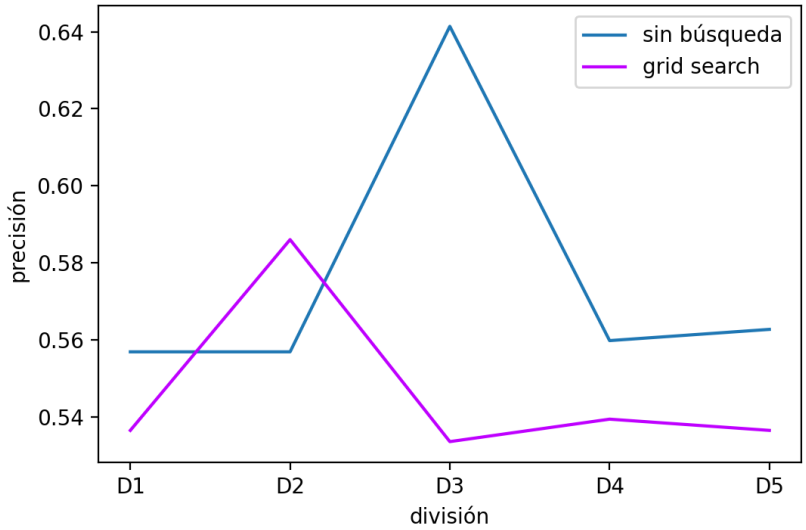

Figura 30: Valores de precisión para la ejecución 2 de RF en hombres.

	D1	D2	D3	D4	D5	Media	Desviación típica
Sin ajustes	0.557	0.557	**0.641**	0.56	0.563	0.576	0.037
Búsq. aleatoria	0.536	**0.586**	0.534	0.539	0.536	0.546	0.022

Tabla 20: Valores de precisión para la ejecución 2 de RF en hombres.

pero de nuevo esto nos cuesta mejor precisión en datos de prueba.

Dada la tardanza de estas ejecuciones, no se probarán los parámetros de la última ejecución en mujeres, pues no existe gran posibilidad de que los resultados mejoren.

7.3. Conectomas de ambos sexos

De nuevo, reutilizaremos los resultados obtenidos en los conjuntos de mujeres y hombres para este.

7.3.1. Ejecución 1

La primera ejecución es una vez más la que usa un diccionario más reducido, requiere 13 horas y da los resultados que aparecen en la **Tabla** y la **Figura** .

Las medias de precisión son de un **58,6 %** para modelos sin ajustes y del **57,9 %** para modelos con búsqueda. De nuevo, no son medias bajas si las comparamos con las obtenidas en

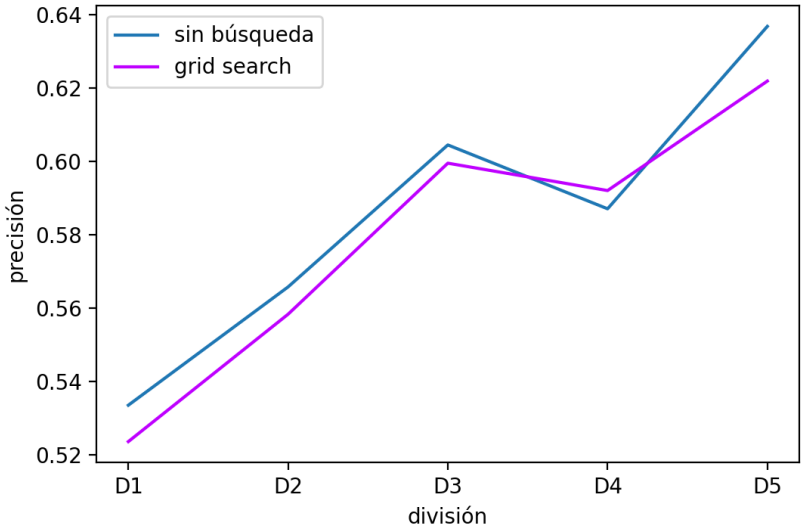

Figura 31: Valores de precisión para la ejecución 1 de RF en ambos sexos.

	D1	D2	D3	D4	D5	Media	Desviación típica
Sin ajustes	0.533	0.566	0.604	0.587	**0.637**	0.586	0.039
Búsq. aleatoria	0.524	0.558	0.600	0.592	**0.622**	0.579	0.039

Tabla 21: Valores de precisión para la ejecución 1 de RF en ambos sexos.

otros modelos. El mejor modelo se obtiene en la última división y calculándose sin búsqueda, con un **63,7 %** de precisión dado por criterio Gini, ninguna profundidad máxima, 100 árboles, y con cálculo automáticos de la cantidad de características usadas.

Las medias de precisión en entrenamiento son del **99,9 %** en modelos sin ajustes y del **93,5 %** en modelos con búsqueda, por lo que tenemos mucho sobre ajuste, de nuevo.

7.3.2. Ejecución 2

Volvemos a intentar reducir el sobreajuste usando un máximo de muestras para *bootstrap* de 45 (la raíz del total de muestras), lo que tarda 13 horas y media en completarse.

Se ha vuelto a reducir la media de precisión para modelos con búsqueda, y ahora con este método no llegamos a obtener un modelo que llegue al **60 %** de precisión. De nuevo, hemos reducido el sobreajuste, alcanzándose solo un **66,3 %** de precisión en entrenamiento, pero no nos sirve de mucho.

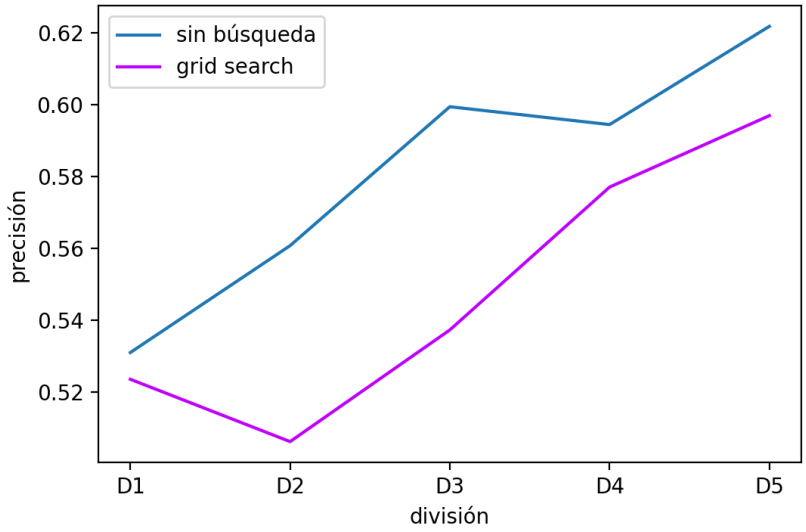

Figura 32: Valores de precisión para la ejecución 2 de RF en ambos sexos.

	D1	D2	D3	D4	D5	Media	Desviación típica
Sin ajustes	0.533	0.566	0.604	0.587	**0.637**	0.586	0.039
Búsq. aleatoria	0.524	0.506	0.537	0.577	**0.597**	0.548	0.038

Tabla 22: Valores de precisión para la ejecución 2 de RF en ambos sexos.

7.4. Conclusiones finales

Los mejores modelos de *random forest* en búsqueda aleatoria se han conseguido con el mismo diccionario para los tres conjuntos de datos. No obstante, para hombres y ambos sexos los mejores resultados se han obtenido sin búsqueda. Los parámetros de los modelos con mejor precisión para cada conjunto son:

- **En mujeres**, criterio de calidad Gini, ninguna profundidad máxima, 100 árboles, y calcula la cantidad de características usadas automáticamente, lo que nos da una precisión del **78 %**. Esta precisión se obtiene también con el modelo sin búsqueda, con configuración de criterio de calidad Gini, ninguna profundidad máxima, 100 árboles, y calcula la cantidad de características usadas automáticamente.

- **En hombres**, criterio de calidad Gini, ninguna profundidad máxima, 100 árboles, y calcula la cantidad de características usadas automáticamente, lo que nos da una precisión

del **64,4 %**.

- **En ambos sexos**, criterio Gini, ninguna profundidad máxima, 100 árboles, y con cálculo automáticos de la cantidad de características usadas, lo que nos da una precisión del **63,7 %**.

Una vez más, la mejor precisión se obtiene para los conectomas de mujeres, que obtienen un máximo del **78 %** de precisión frente al **64,4 %** y el **63,7 %** de los conjuntos de hombres y ambos sexos. Esta vez, habríamos conseguido los mismos modelos realizando o no búsqueda de parámetros, pues en los tres casos la precisión máxima se puede obtener sin ajustes. Esto destaca, ya que cada búsqueda aleatoria tarda mucho más que la creación de un modelo sin ajustes.

8

Máquinas de vectores de soporte

Las **máquinas de vectores de soporte** se crearán una vez más buscando optimizar primero los datos de mujeres y luego trasladando estos resultados a los otros dos conjuntos. Dejaremos en sus valores por defecto los parámetros *degree, shrinking, class_weight, verbose y max_iter*. Además, como **nuestra clasificación es binaria**, no tendremos que ajustar *decision_function_shape* ni *break_ties*. Usaremos un *random_state* de 3 tanto en búsqueda como en el algoritmo para poder reproducir sus resultados. El tamaño de caché (*cache_size*) será 1000, para que la ejecución tarde menos.

8.1. Conectomas de mujeres

8.1.1. Ejecución 1

Para la primera ejecución, usaremos los siguientes parámetros:

- **Regularización** (C): tomará valores entre 0,1 y 1, con saltos de 0,1.
- **Función *kernel*** (*kernel*): podrá ser lineal, polinómica, sigmoide o de base radial (rbf).
- **Coeficiente gamma** para *kernels* polinomiales, sigmoides y de base radial (*gamma*): podrá ser $\frac{1}{num\_caracteristicas}$ (*auto*) o $\frac{1}{num\_caracteristicas \times num\_individuos}$ (*scale*).

Esta configuración nos supone un tiempo de ejecución de un cuarto de hora, dando los resultados ilustrados en la **Tabla 23** y la **Figura 33**.

Las medias con y sin búsqueda son muy parecidas, un **70,1 %** y un **70,4 %** respectivamente. Esto es porque las precisiones de los modelos estudiados en cada división son casi iguales, por

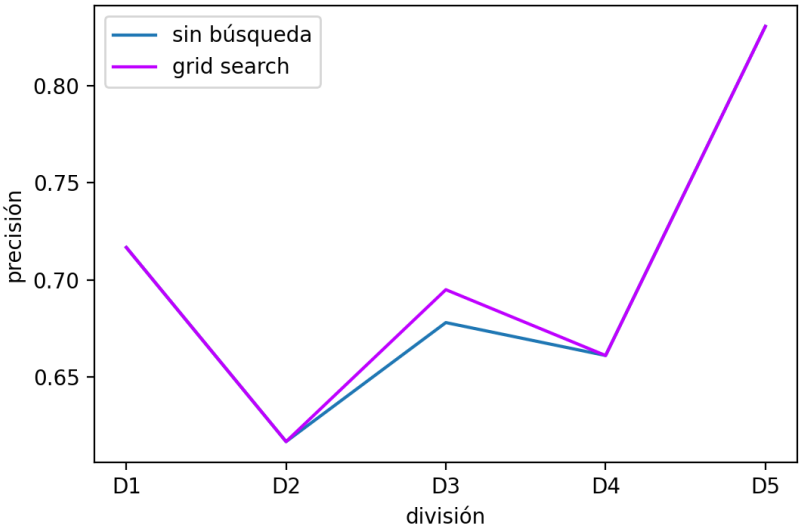

Figura 33: Valores de precisión para la ejecución 1 de SVM en mujeres.

	D1	D2	D3	D4	D5	Media	Desviación típica
Sin ajustes	0.717	0.617	0.678	0.661	**0.831**	0.701	0.081
Búsq. aleatoria	0.717	0.617	0.695	0.661	**0.831**	0.704	0.080

Tabla 23: Valores de precisión para la ejecución 1 de SVM en mujeres.

lo que podemos asumir que el **mejor modelo** que encuentra la búsqueda va a ser el mismo que se calculaba sin ajustes. Esto lo tendremos en cuenta en la siguiente ejecución. El mejor modelo se consigue en la quinta división, con un **83,1 %** de precisión conseguido por un kernel sigmoide, gamma con valor *scale*, y valor C 0,8. Pero también se consigue con la configuración predeterminada, que cambia el kernel a rbf y el valor de C a 1.

Teniendo en cuenta los valores de precisión que obtenemos, hay poco sobreajuste comparado con los modelos anteriores, pues llegamos a una media de precisión en entrenamiento del **93,8 %** para modelos sin ajustes y del **86 %** para modelos con búsqueda.

8.1.2. Ejecución 2

Para esta ejecución se realiza una exploración, haciendo pruebas con otros parámetros. Tras ellas, sacamos las siguientes conclusiones:

- Modificar el **rango de C** no afecta a la precisión final, aunque en teoría debería. Pero sí

mejora el tiempo, por lo que cambiamos este rango a entre 1 y 5, con saltos de 0,5.

- Añadir valores a la **tolerancia de parada** no afecta a la precisión, por lo que lo dejamos en su valor predeterminado.

- Añadir valores para ***coef0*** mejora algo los resultados, por lo que establecemos un rango de entre 0 y 4.

- Cambiar el valor de *probability* a *True* empeora el tiempo de ejecución y no mejora los resultados, por lo que lo dejaremos en su predeterminado, *False*.

Como el **tipo de kernel** es muy importante para este algoritmo, hacemos pruebas con las cuatro posibilidades para encontrar la mejor, y obtenemos que es la función sigmoide, que da la mejor precisión (**Tabla 24**).

Kernel	lineal	polinómico	rbf	sigmoide
Prec. media	0.697	0.701	0.704	**0.724**

Tabla 24: Precisión para diferentes kernels en SVM para mujeres

Por tanto en esta ejecución usamos la **función sigmoide** con los valores de gamma que ya empleábamos antes, y los nuevos rangos comentados. Como estamos usando un diccionario tan pequeño, aunque el algoritmo sea de búsqueda aleatoria estamos explorando todas las posibilidades (que son menos de 460). Esta ejecución necesita 15 minutos para completarse, y sus resultados se muestran en la **Tabla 25** y la **Figura 34**.

	D1	D2	D3	D4	D5	Media	Desviación típica
Sin ajustes	0.717	0.617	0.678	0.661	**0.831**	0.701	0.081
Búsq. aleatoria	0.717	0.633	0.746	0.712	**0.814**	0.724	0.065

Tabla 25: Valores de precisión para la ejecución 2 de SVM en mujeres.

Aunque la media de precisión en búsqueda aleatoria ha mejorado (ahora es del **72,4 %**), el mejor modelo sigue siendo el que obtuvimos en la ejecución anterior. Pero sí hemos conseguido reducir la diferencia entre los resultados de unos y otros modelos, es decir, la varianza. El sobreajuste sigue siendo el mismo.

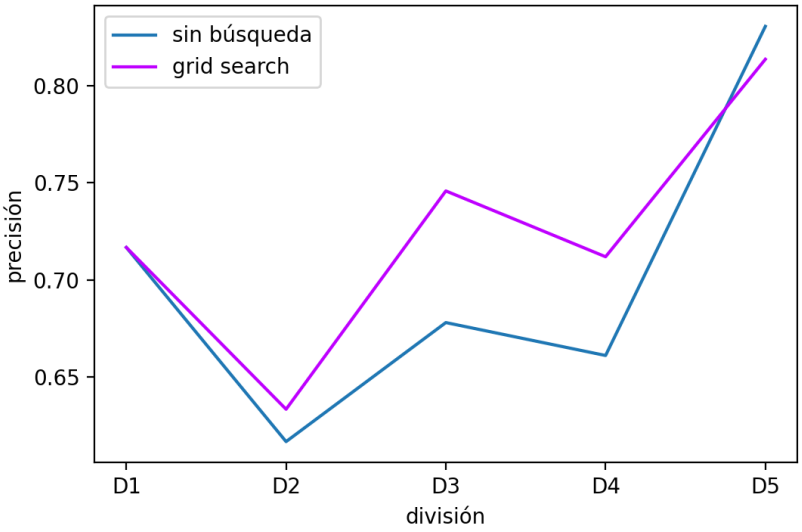

Figura 34: Valores de precisión para la ejecución 1 de SVM en mujeres.

8.2. Conectomas de hombres

8.2.1. Ejecución 1

Como hemos visto que la configuración predeterminada de SVM da buenos resultados, usaremos el diccionario de la segunda ejecución de mujeres, manteniendo los modelos sin búsqueda con sus valores predeterminados. Con esta configuración, la ejecución requiere 8 horas y 45 minutos, ilustrada en la **Tabla 26** y la **Figura 35**.

	D1	D2	D3	D4	D5	Media	Desviación típica
Sin ajustes	0.536	0.566	**0.641**	0.531	0.560	0.567	0.044
Búsq. aleatoria	0.542	0.554	**0.653**	0.522	0.566	0.567	0.051

Tabla 26: Valores de precisión para la ejecución 1 de SVM en hombres.

Las medias de precisión son iguales con ambos métodos, **56,7 %**, pero el **mejor modelo** se consigue con búsqueda, con un **65,3 %** de precisión que usa kernel sigmoide, gamma con valor *scale*, *coef0* igual a 0 y 2,5 como valor de C.

En valores de entrenamiento, tenemos mucho sobreajuste en modelos sin búsqueda, pues tienen un **92,7 %** de precisión media; pero en modelos con búsqueda la media es del **75,8 %**. No realizaremos más ejecuciones por el tiempo que requieren y la baja probabilidad que existe de

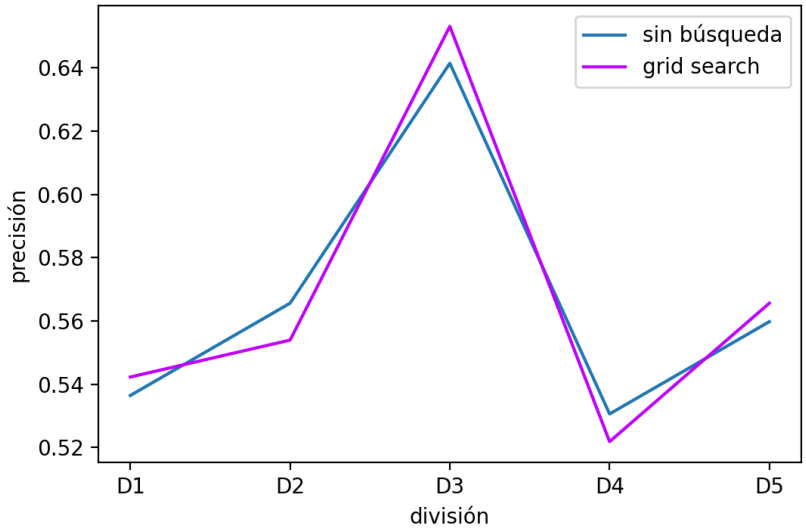

Figura 35: Valores de precisión para la ejecución 1 de SVM en hombres.

que den mejores resultados.

8.3. Conectomas de ambos sexos

8.3.1. Ejecución 1

De nuevo, usamos el diccionario de la segunda ejecución de mujeres en búsqueda y mantenemos los valores predeterminados para la ejecución sin búsqueda. La ejecución cuyos resultados se ilustran en la **Tabla 27** y la **Figura 36** tarda 12 horas.

	D1	D2	D3	D4	D5	Media	Desviación típica
Sin ajustes	0.514	0.526	**0.634**	0.597	0.634	0.581	0.058
Búsq. aleatoria	0.506	0.524	0.637	0.600	**0.642**	0.582	0.063

Tabla 27: Valores de precisión para la ejecución 1 de SVM en ambos sexos.

Las medias para ambos métodos son prácticamente iguales, un **58,1 %** de precisión para el primero y un **58,2 %** para el segundo. Pero el **mejor modelo** lo encontramos en la quinta división, creado por búsqueda. Este nos da una precisión del **64,2 %**, con kernel sigmoide, gamma con valor *scale*, *coef0* igual a 0, y valor C de 2.

Para entrenamiento tenemos un valor de precisión del **92,4 %** para modelos sin ajuste y

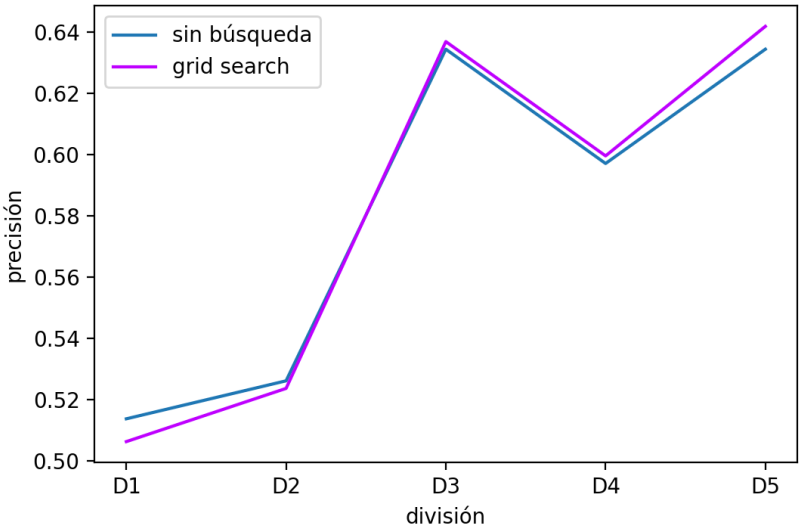

Figura 36: Valores de precisión para la ejecución 1 de SVM en ambos sexos.

del **73,4 %** con búsqueda, con lo que tenemos un sobreajuste parecido al que encontrábamos antes. De nuevo, no realizaremos más ejecuciones.

8.4. Conclusiones finales

Los **mejores modelos de máquinas de vectores de soporte** se han conseguido con ajustes parecidos en todos los conjuntos de datos. Estos son:

- **En mujeres**, conseguimos un **83,1 %** de precisión conseguido por un kernel sigmoide, gamma con valor *scale*, y valor C 0,8.
- **En hombres**, un **65,3 %** de precisión que usa kernel sigmoide, gamma con valor *scale*, *coef0* igual a 0 y 2,5 como valor de C.
- **En ambos sexos**, un **64,2 %**, con kernel sigmoide, gamma con valor *scale*, *coef0* igual a 0, y valor C de 2.

En este último método también hemos obtenido la **mejor precisión en los datos de mujeres**, muy por encima de la obtenida con los demás. **No hemos conseguido eliminar el sobreajuste**, ni mejorar demasiado los resultados obtenidos con los parámetros predeterminados usando búsqueda.

9

Perceptrones multicapa

Para el **perceptrón multicapa** probamos una vez más el diccionario primero en mujeres y luego lo usamos en hombres y ambos sexos. Usamos los valores predeterminados de número de neuronas en capa oculta (*hidden_layer_sizes*), *alpha, shuffle, verbose, beta_1, beta_2, epsilon, n_iter_no_change, max_fun, momentum, power_t* y *nesterovs_momentum*. Muchos de ellos sirven solamente para uno de los valores de *solver*, y otros tienen valores predeterminados que suelen dar resultados buenos.

9.1. Conectomas de mujeres

9.1.1. Ejecución 1

Para esta primera ejecución daremos valores a los parámetros que no son numéricos:

- La **función de activación** (*activation*) podrá ser las funciones $f(x) = x$ (*identity*), $f(x) = \frac{1}{1+exp(-x)}$ (*logistic*), $f(x) = tanh(x$ (*tanh*) o $f(x) = max(0, x)$ (*relu*).
- La **manera de optimizar los pesos** (*solver*) será el optimizador lbfgs (*lbfgs*) o el optimizador adam (*adam*), ay que la opción sgd tarda mucho en converger y da los mismos resultados.
- La **tasa de aprendizaje** será constante (*constant*), decreciente (*invscaling*) o constante mientras los resultados mejoren, luego se adaptará (*adaptative*).

Con esta configuración, tardamos 25 minutos en completar la ejecución, y obtenemos los resultados de la **Tabla 29** y la **Figura 38**.

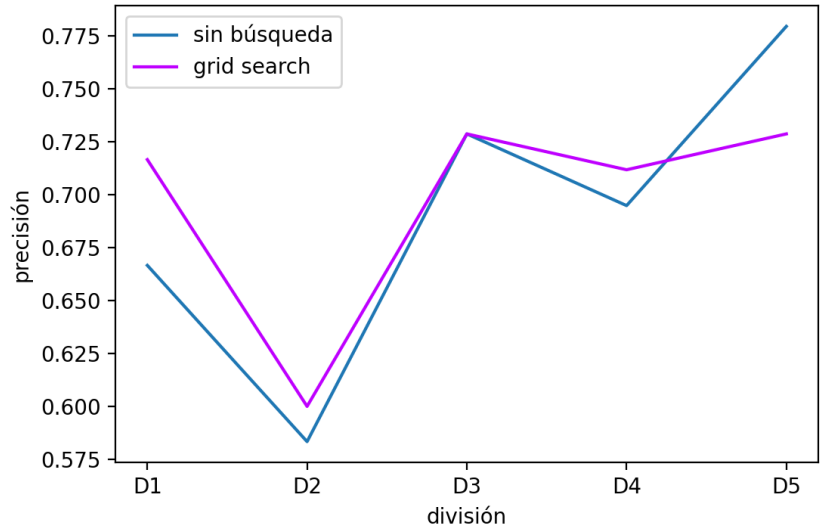

Figura 37: Valores de precisión para la ejecución 1 de MLP en mujeres.

	D1	D2	D3	D4	D5	Media	Desviación típica
Sin ajustes	0.667	0.583	0.729	0.695	**0.780**	0.691	0.073
Búsq. aleatoria	0.717	0.600	**0.729**	0.712	**0.729**	0.697	0.055

Tabla 28: Valores de precisión para la ejecución 1 de MLP en mujeres.

Tenemos medias del **69,1 %** y el **69,7 %** de precisión para modelos sin ajustes y con búsqueda, obteniéndose el mejor modelo en la última división y sin búsqueda, con activación *relu*, tasa de aprendizaje constante y adam como optimizador. este modelo da una precisión del **78 %**.

En cuanto al sobreajuste, las medias en entrenamiento son del **100 %** para ambos métodos, por lo que este es muy alto.

9.1.2. Ejecución 2

En esta segunda ejecución, vamos a usar una estrategia parecida a la que usamos ya en los árboles de decisión, comprobar parámetro por parámetro qué valores son mejores para él. Tras estas pruebas se llega a las siguientes conclusiones:

- La función de activación permanece como antes, con sus cuatro valores.
- El optimizador será el predeterminado, adam, porque da mejores resultados.

- La tasa de aprendizaje será constante, pues siempre se elige como mejor.
- El valor de *tol* será el predeterminado, 0.0001, otros empeoran el tiempo.
- Se necesitan unas 500 iteraciones máximas para que los modelos converjan.
- La fracción de validación se dejará en su valor predeterminado, 0,1, porque este es el que elige siempre como mejor la búsqueda.
- No usaremos *early_stopping* porque aunque reduzca el sobreajuste también se reduce la precisión en datos de pruebas.

Con esta nueva configuración, vemos que solamente daremos a elegir solamente el valor de función de activación. Esta ejecución requiere 6 minutos, y da los resultados de la **Tabla 29** y la **Figura 38**.

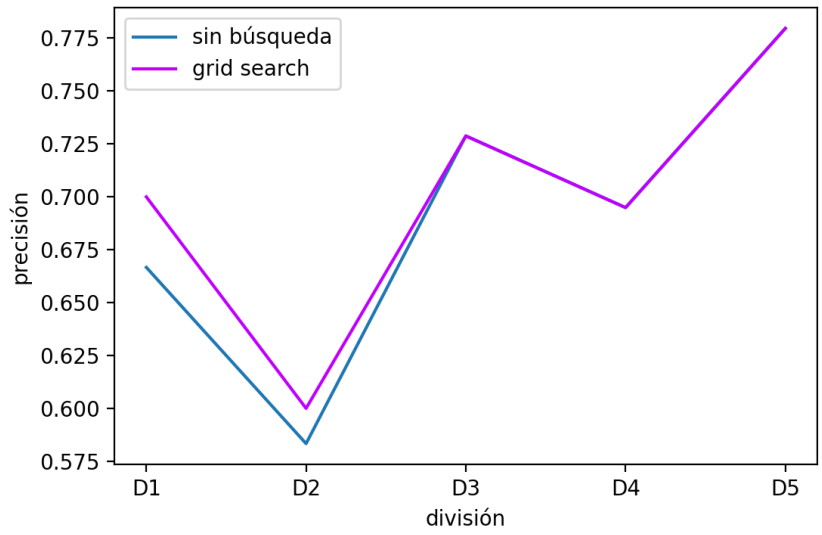

Figura 38: Valores de precisión para la ejecución 2 de MLP en mujeres.

	D1	D2	D3	D4	D5	Media	Desviación típica
Sin ajustes	0.667	0.583	0.729	0.695	**0.780**	0.691	0.073
Búsq. aleatoria	0.667	0.583	0.729	0.695	**0.780**	0.701	0.066

Tabla 29: Valores de precisión para la ejecución 2 de MLP en mujeres.

La media de búsqueda mejora, siendo ahora del **70,1 %**, y su mejor modelo alcanza la misma precisión que el mejor de la creación sin ajustes, un **78 %**, pero no consigue superarlo.

El sobreajuste se ha mantenido, sin conseguirse ninguna mejora que no costara precisión en datos de prueba.

9.2. Conectomas de hombres

9.2.1. Ejecución 1

En esta ejecución tomaremos los mismos valores que la segunda ejecución de conectomas de mujeres. La ejecución se completa en 40 minutos, dando los resultados de la **Tabla 30** y la **Figura 39**.

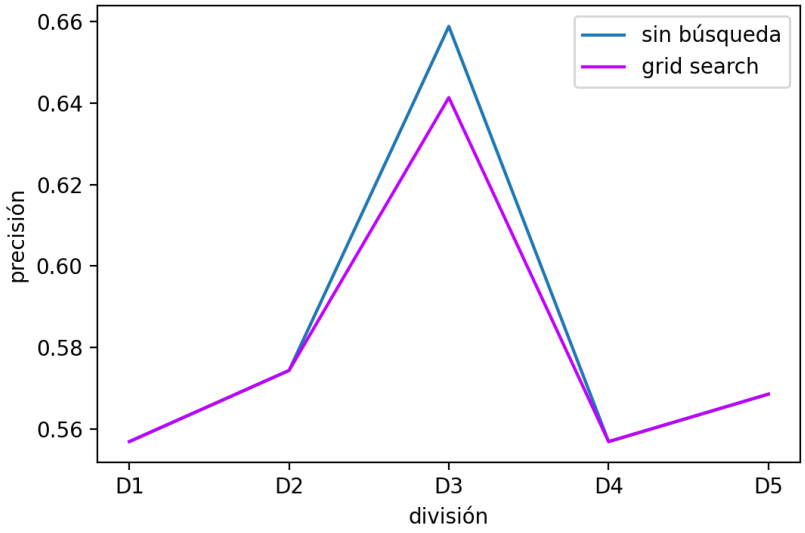

Figura 39: Valores de precisión para la ejecución 1 de MLP en hombres.

	D1	D2	D3	D4	D5	Media	Desviación típica
Sin ajustes	0.557	0.574	**0.659**	0.557	0.569	0.583	0.053
Búsq. aleatoria	0.557	0.574	**0.641**	0.557	0.569	0.580	0.035

Tabla 30: Valores de precisión para la ejecución 1 de MLP en hombres.

Las medias de precisión son del **58,3 %** y del **58 %**, muy similares. El mejor modelo lo obtenemos una vez más sin realizar búsqueda, con un **65,9 %** de precisión dados por activación

relu, tasa de aprendizaje constante y adam como optimizador.

El sobreajuste es, una vez más, muy grande, pues en ambos casos se llega al **100 %** de precisión para datos de entrenamiento.

9.3. Conectomas de ambos sexos

9.3.1. Ejecución 1

Volvemos a usar los mismos parámetros, lo que requiere 81 minutos de ejecución. Con ella obtenemos los resultados ilustrados en la **Tabla 31** y la **Figura 40**.

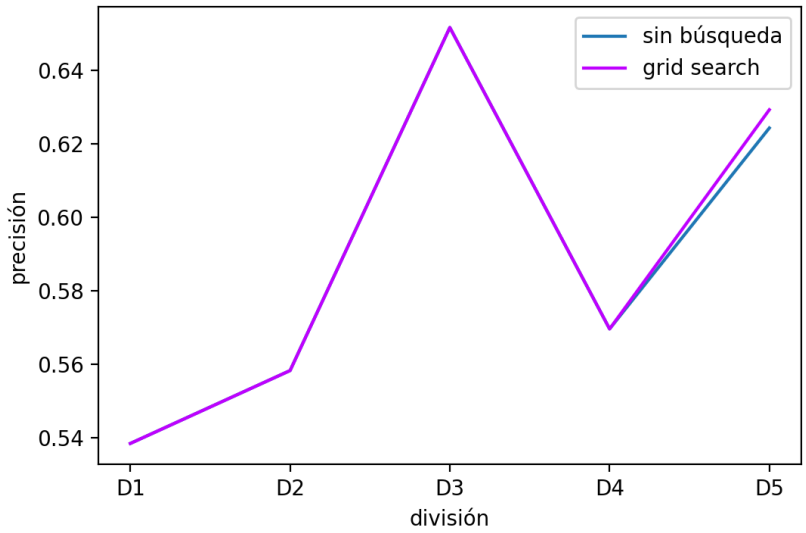

Figura 40: Valores de precisión para la ejecución 1 de MLP en ambos sexos.

	D1	D2	D3	D4	D5	Media	Desviación típica
Sin ajustes	0.538	0.558	**0.652**	0.570	0.624	0.589	0.078
Búsq. aleatoria	0.538	0.558	0.652	0.570	**0.629**	0.580	0.049

Tabla 31: Valores de precisión para la ejecución 1 de MLP en ambos sexos.

Las medias son de un **58,9 %** y un **58 %** de precisión, pero el mejor modelo de ambos métodos tiene la misma precisión, un **65,2 %**, conseguido con los valores predeterminados: activación *relu*, tasa de aprendizaje constante y adam como optimizador.

En ambos métodos la media en entrenamiento es del **100 %**.

9.4. Conclusiones finales

Los mejores modelos de perceptrones multicapa se han conseguido en los tres casos con los valores predeterminados: activación *relu*, tasa de aprendizaje constante y adam como optimizador. Con ellos se ha llegado a un **78 %** de precisión en mujeres, un **65,9 %** para hombres y un **65,2 %** para ambos sexos.

En este último método también hemos obtenido la mejor precisión en los datos de mujeres, muy por encima de la obtenida con los demás. No hemos conseguido eliminar el sobreajuste, ni mejorar los resultados obtenidos con los parámetros predeterminados usando búsqueda.

10 Modelos elegidos

En este capítulo recopilaremos los modelos con mayor precisión de cada algoritmo para cada conjunto de datos. En nuestra interfaz mostraremos la predicción con todos, pero indicando las precisiones sobre datos de prueba de cada uno.

10.1. Mujeres

Modelo	KNN	DT	RF	SVM	MLP
Precisión	74,6 %	75,0 %	78,0 %	**83,1 %**	78,0 %

Tabla 32: Valores de precisión para los mejores modelos en mujeres.

En mujeres, el mejor modelo lo encontramos con máquinas de soporte vectorial, con función *kernel* sigmoidea con gamma $\frac{1}{num\_caractersticas \times num\_individuos}$ y valor de regularización (C) igual a 0,8. La precisión es muy buena, de un **83,1 %**.

10.2. Hombres

Modelo	KNN	DT	RF	SVM	MLP
Precisión	58,3 %	57,1 %	64,1 %	65,3 %	**65,9 %**

Tabla 33: Valores de precisión para los mejores modelos en hombres.

En hombres, el mejor modelo viene dado por los valores predeterminados del perceptrón multicapa, función de activación $f(x) = max(0, x)$ (***relu***), tasa de aprendizaje constante y adam como optimizador. Su precisión, un **65,9 %**, es aceptable, pero no demasiado buena.

Modelo	KNN	DT	RF	SVM	MLP
Precisión	60,7 %	58,7 %	63,7 %	64,2 %	**65,2 %**

Tabla 34: Valores de precisión para los mejores modelos en ambos sexos.

10.3. Ambos sexos

En el conjunto de ambos sexos el mejor modelo tiene los mismos parámetros que el de hombres, con una precisión algo menor, del **65,2 %**, que una vez más, no es demasiado buena.

10.4. Recopilación de los modelos

Para poder usar estos modelos en una interfaz, los guardamos en archivos individuales creando un *script* de Python y almacenándolos en la carpeta de nuestro proyecto.

11 Interfaz

Una vez tenemos los modelos con mayor precisión, se crea una herramienta con la que se puedan usar para dar predicciones sobre otros pacientes. Esta interfaz es una aplicación web creada con Flask [34], ya que está disponible dentro del lenguaje Python, que hemos usado durante todo el proyecto, y es también un paquete fácil de usar en el caso de aplicaciones poco complejas como la nuestra, pues podemos emplear programación para mostrar las páginas HTML que queramos, y realizar el proceso de extracción de conectomas y cálculo de predicciones en un mismo *script*. Todas las páginas HTML de la aplicación se han creado a partir de una plantilla creada con ayuda de una hoja de estilos de ***Bootstrap*** [35].

La aplicación se incluirá en el mismo proyecto que el resto de programas creados durante este trabajo, en la carpeta *app*. Para poder ejecutar la *Python/FSL Resting State Pipeline* se ha incluido esta carpeta dentro del entorno virtual, además de las librerías y programas de los que depende. En este proyecto incluiremos también los documentos con los modelos extraídos en el capítulo anterior.

Para poder usar correctamente la aplicación, esta se debe de iniciar con permisos de administrador, en caso de Linux de usuario *root*.

El proceso seguido por la aplicación para realizar las predicciones de un archivo se ilustra en la **Figura 41**.

11.1. Página de inicio

La página de inicio *index.html* (**Figura 42**) contiene un breve texto explicativo (**Figura 43**) sobre su objetivo y modo de uso, seguido del formulario con tres desplegables en los que el usuario debe indicar si el archivo que subirá será un MRI funcional, en cuyo caso este tiene que tener el formato *.nii*; o un conectoma normalizado o no, en ambos casos en formato *.csv*.

Los otros dos desplegables permiten el sexo del paciente y el modelo que se quiere usar. Si el

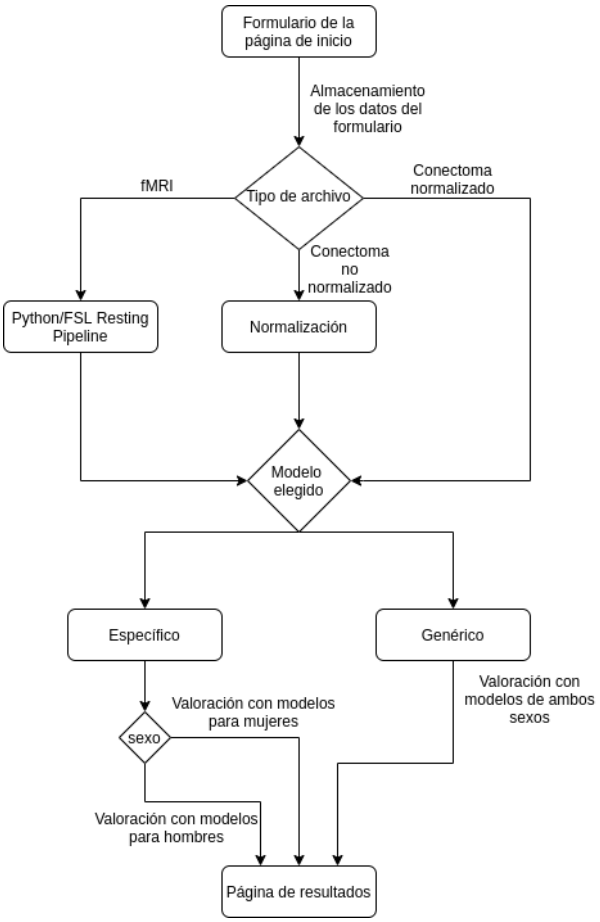

Figura 41: Funcionamiento de la aplicación.

modelo marcado es el específico, entonces se usarán los correspondientes al sexo seleccionado, y si por lo contrario es el general, se usarán los modelos creados con los datos de ambos sexos.

Por último, el formulario tiene disponible un botón de selección de archivo para que el usuario suba el archivo a estudiar.

Cuando el usuario pulsa el botón *Realizar predicción*, se recogen los datos introducidos en los desplegables del formulario, que se guardan como variables de sesión. El archivo se guarda en la carpeta *uploads* dentro del directorio de la aplicación. Dependiendo del tipo de archivo seleccionado en el formulario, habrá tres posibles vías.

- Si se ha seleccionado la opción **fMRI**, después de almacenar el archivo ejecutaremos el proceso de extracción del conectoma llamando a un *script bash* incluido en el proyecto, cuyos resultados se almacenarán en la misma carpeta, *uploads*, tras lo que se leerá el vector triangular superior matriz *zr_matrix.csv*, que contendrá el conectoma ya norma-

Figura 42: Página de inicio.

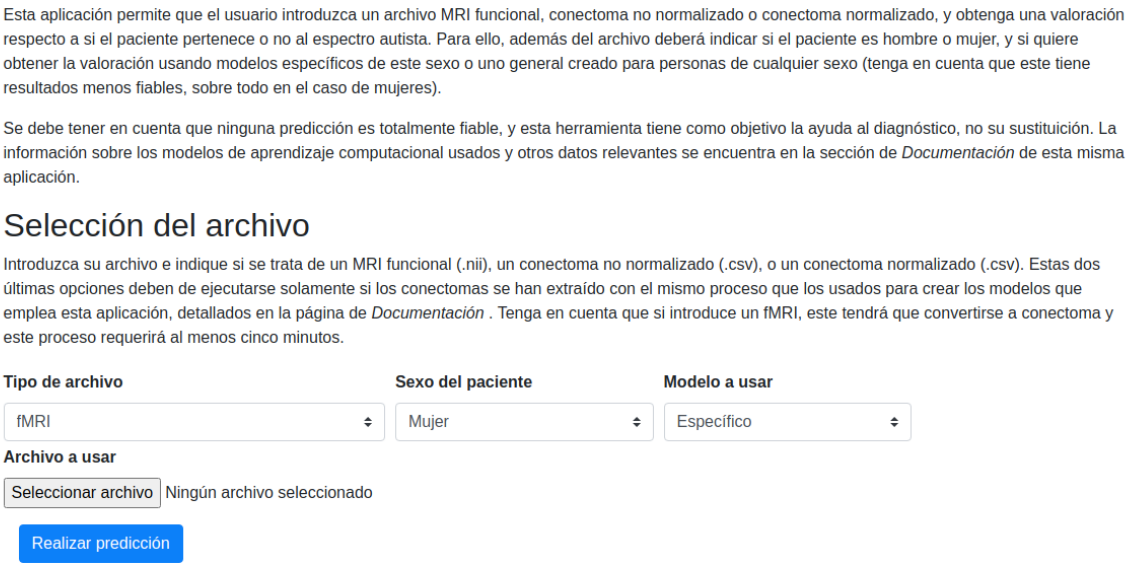

Figura 43: Texto de la página de inicio.

lizado.

- Si, en cambio, hemos introducido un conectoma no normalizado, entonces el archivo se leerá y normalizará con el mismo criterio seguido en la *pipeline* de Duke, y luego se extraerá su matriz triangular superior.

- Finalmente, si el conectoma ya está normalizado, sólo se necesita extraer su matriz trian-

gular superior.

En nuestro programa se crea también un diccionario *confianza* que tiene una clave para las precisiones dadas por cada modelo.

Independientemente del tipo de archivo elegido, los siguientes pasos serán idénticos. Una vez hemos extraído la matriz triangular superior, según se haya elegido el modelo genérico o el específico del sexo del paciente, se pasará este vector a la función correspondiente para calcular las predicciones con los cinco algoritmos. Estas predicciones se almacenan como variable de sesión y se usan en la función *dar_resultados*, que transforma los ceros y unos dados por los algoritmos (correspondientes con que el paciente sea neurotípico o tenga TEA) en las frases *El paciente es neurotípico* y *El paciente tiene TEA*, para almacenarlas como nuevos resultados. Tras ello, se establecerá una variable *sexo*, cuyo valor dependerá de si se ha elegido el modelo genérico, y en caso contrario, del sexo indicado en el formulario.

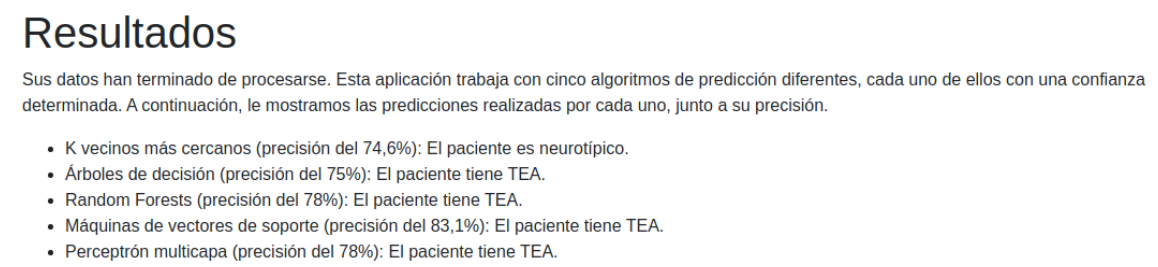

Figura 44: Resultados de la predicción.

Acto seguido, se muestra la página de resultados (*results.html*, **Figura 44**), a la que se le envían las variables *resultados, confianza* y *sexo*, que se usan para escribir las predicciones de cada modelo (**Figura 44**). Estas se incluyen en una lista, donde cada entrada tiene el nombre del modelo, su precisión en datos de pruebas entre paréntesis, y su predicción.

11.2. Página de documentación

La aplicación cuenta también con una sección **Documentación** (*documentacion.html*, **Figura 45**)) que aporta información al usuario sobre el proyecto desarrollado, los datos y métodos empleados, y los modelos usados. Se indica también que la herramienta tiene el objetivo de ser una ayuda al diagnóstico, y nunca un sustituto.

Este proyecto se ha creado con la ayuda de los MRIs funcionales incluidos en la iniciativa ABIDE. Los conectomas se han podido extraer usando la *Python/FSL Resting State Pipeline* del Centro de Análisis e Imagen del cerebro de la Universidad de Duke. Esta herramienta es también la empleada cuando el usuario introduzca un fMRI para obtener resultados. El uso de los algoritmos incluidos en *scikit-learn* ha sido esencial para lograr su completitud.

Los algoritmos usados para crear nuestros modelos de predicción han sido k vecinos más cercanos, árboles de decisión, *random forests*, máquinas de vectores de soporte y perceptrones multicapa. Los modelos se han creado usando validación cruzada para obtener los mejores resultados posibles.

A pesar del extensivo proceso llevado a cabo para crear estos modelos, estos no tienen una fiabilidad total, sino que sus precisiones varían. Es por ello que se incluyen los resultados de cada tipo de algoritmo, para que el usuario pueda valorarlos. Esta aplicación tiene como objetivo ser una ayuda al diagnóstico, nunca un sustituto de los profecsionales médicos.

Esta aplicación es parte del Trabajo de Fin de Grado de Clara Jiménez Valverde, estudiante de Ingeniería de la Salud con mención en Bioinformática en la Universidad de Málaga.

Figura 45: Página de documentación

12

Conclusiones y líneas futuras

12.1. Conclusiones

En este proyecto se ha trabajado con gran cantidad de datos en la mayoría de sus pasos, y esto se ha traducido en la necesidad de emplear mucho tiempo en ellos. En primer lugar, la recopilación de las imágenes de resonancia magnética funcional ha sido lenta por los protocolos de descarga que tenían implantados, y su volumen. El proceso de cálculo de conectomas ha sido también largo, al igual que la creación de modelos para datos de hombres y ambos sexos. Los modelos para mujeres han sido más rápidos de crear, pues los conectomas disponibles eran muchos menos. Esto nos lleva a la importancia de tener equipos de calidad y procesos lo más optimizados posibles a la hora de llevar a cabo proyectos de mayor escala. Aun habiendo usado potencia externa (Google Colab), muchos modelos han requerido horas para crearse, lo que aunque no supone una atención constante, pero sí necesita de cierta supervisión.

Los modelos creados durante el proyecto han dado precisiones muy dispares. Por un lado, los modelos creados para mujeres dan muy buenos resultados, llegando al 83,1 % de precisión. Esto sucede a pesar de que los conectomas de mujeres eran muchos menos que los de hombres, lo que en un principio podría haber inducido a pensar que al tener menos muestras, los modelos serían menos precisos. Pero no ha sido así. De hecho, la máxima precisión conseguida para los datos de hombres es de un 65,9 % y para los de ambos sexos un 65,2 %. En mujeres, el mejor modelo ha sido el obtenido con máquinas de vectores de soporte, y en hombres y ambos sexos se ha obtenido con el perceptrón multicapa. Como los datos de ambos sexos son mayormente de hombres, no es de extrañar que el mejor modelo en ambos conjuntos se consiga con el mismo algoritmo, y la precisión sea similar. Algo que no se ha conseguido resolver

para la mayoría de los modelos es el sobreajuste, pues la diferencia de precisión para datos de entrenamiento y de prueba es en la mayoría de los casos alta, a pesar de haber aplicado medidas para reducirla.

La interfaz creada es una herramienta simple pero funcional que permite al usuario subir un archivo, indicar el sexo del paciente, el modelo a emplear y el tipo de archivo usado. La aplicación devuelve una valoración, que a pesar de no ser totalmente fiable, sí es útil como ayuda al diagnóstico y podría ser empleado en el ambiente sanitario.

Durante el desarrollo de este proyecto se ha conseguido calcular el conectoma de más de dos mil fMRIs, emplear estos para entrenar cinco tipos de modelos de clasificación diferentes para conjuntos de datos de mujeres, hombres y ambos, escoger los que ofrecían una mayor precisión, e integrarlos en una aplicación web que supone un producto final que el usuario podrá usar para introducir otros fMRIs o conectomas y obtener una valoración por parte de los cinco modelos correspondientes. Para lograrlo se han combinado conocimientos de diferentes ámbitos, lo cual es muy frecuente en proyectos de informática, pero aún más en bioinformática. Este proyecto es un claro ejemplo de la multidisciplinariedad actual presente en el campo de la medicina.

12.2. Líneas Futuras

En el futuro, este proyecto podría ampliarse usando otros métodos de clasificación, con el objetivo de crear modelos más precisos y que reduzcan el sobreajuste. Otra posibilidad de mejora de los modelos de hombres y ambos sexos sería intentar reducir la cantidad de conectomas usados, que aunque en un principio no debiera mejorar los resultados, es una prueba simple que nos puede ayudar a, como mínimo, descartar la posibilidad de que los modelos de mujeres sean mejores porque se hayan creado sobre una menor cantidad de datos.

Podría también hacerse otro estudio con los datos del proyecto ABIDE Preprocessed, perteneciente a la Neuro Bureau Preprocessing Initiative [36]. Este reúne series temporales extraídas de ABIDE I, con cuatro *pipelines* diferentes. Podrían crearse modelos con cada una de las cuatro y comparar sus resultados, para comprobar si existe otro proceso de cálculo de conectomas cuyos resultados den modelos de mayor precisión.

La interfaz funciona, pero como se ha comentado, es un entorno simple. Se podría mejorar, dándole una apariencia más visualmente atractiva, y en caso de crear modelos con ABIDE

Preprocessed, se podrían añadir estos modelos y la opción de emplear los creados por cada una de las *pipelines*.

Además, este mismo sistema se puede ampliar y usarlo en fMRIs de otros trastornos en los que la causa sea las conexiones cerebrales, como algunos tipos de depresión, la esquizofrenia, o el trastorno de déficit de atención. Para ello será necesario realizar una búsqueda de bases de datos que contengan estas imágenes.

Referencias

[1] Matthew J. Maenner y col. "Prevalence of autism spectrum disorder among children aged 8 Years-Autism and developmental disabilities monitoring network, 11 Sites, United States, 2016". En: *MMWR Surveillance Summaries* 69.4 (2020), págs. 1-12. ISSN: 15458636. DOI: 10.15585/MMWR.SS6904A1.

[2] Laura Pérez-Crespo y col. "Temporal and Geographical Variability of Prevalence and Incidence of Autism Spectrum Disorder Diagnoses in Children in Catalonia, Spain". En: *Autism Research* 12.11 (2019), págs. 1693-1705. ISSN: 19393806. DOI: 10.1002/aur.2172.

[3] M. Rutter. "Incidence of autism spectrum disorders: Changes over time and their meaning". En: *Acta Paediatrica, International Journal of Paediatrics* 94.1 (2005), págs. 2-15. ISSN: 08035253. DOI: 10.1080/08035250410023124.

[4] Michael C.F. Smith. "Causes and consequences of delayed diagnosis of autism spectrum disorder in forensic practice: a case series". En: *Journal of Intellectual Disabilities and Offending Behaviour* December (2021). ISSN: 20508824. DOI: 10.1108/JIDOB-10-2020-0017.

[5] Suman Raj y Sarfaraz Masood. "Analysis and Detection of Autism Spectrum Disorder Using Machine Learning Techniques". En: *Procedia Computer Science* 167.2019 (2020), págs. 994-1004. ISSN: 18770509. DOI: 10.1016/j.procs.2020.03.399. URL: https://doi.org/10.1016/j.procs.2020.03.399.

[6] Maitha Rashid Alteneiji, Layla Mohammed Alqaydi y Muhammad Usman Tariq. "Autism spectrum disorder diagnosis using optimal machine learning methods". En: *International Journal of Advanced Computer Science and Applications* 11.9 (2020), págs. 252-260. ISSN: 21565570. DOI: 10.14569/IJACSA.2020.0110929.

[7] Seok Jun Hong y col. "Atypical functional connectome hierarchy in autism". En: *Nature Communications* 10.1 (2019), págs. 1-13. ISSN: 20411723. DOI: 10.1038/s41467-019-08944-1.

[8] Xi Nian Zuo. "Editorial: Mapping the Miswired Connectome in Autism Spectrum Disorder". En: *Journal of the American Academy of Child and Adolescent Psychiatry* 59.3 (2020), págs. 348-349. ISSN: 15275418. DOI: 10.1016/j.jaac.2020.01.001. URL: https://doi.org/10.1016/j.jaac.2020.01.001.

[9] Kenia Martínez y col. "Sensory-to-Cognitive Systems Integration Is Associated With Clinical Severity in Autism Spectrum Disorder". En: *Journal of the American Academy of Child and Adolescent Psychiatry* 59.3 (2020), págs. 422-433. ISSN: 15275418. DOI: 10.1016/j.jaac.2019.05.033.

[10] Georgia Lockwood Estrin y col. "Barriers to Autism Spectrum Disorder Diagnosis for Young Women and Girls: a Systematic Review". En: *Review Journal of Autism and Developmental Disorders* (2020). ISSN: 21957185. DOI: 10.1007/s40489-020-00225-8.

[11] Meng Chuan Lai y col. "Sex/Gender Differences and Autism: Setting the Scene for Future Research". En: *Journal of the American Academy of Child and Adolescent Psychiatry* 54.1 (2015), págs. 11-24. ISSN: 15275418. DOI: 10.1016/j.jaac.2014.10.003. URL: http://dx.doi.org/10.1016/j.jaac.2014.10.003.

[12] Leo Kanner. *Autistic Disturbances of Affective Contact*. 1943.

[13] Hans Asperger. "'Autistic psychopathy' in childhood". En: *Autism and Asperger Syndrome*. Ed. por UtaEditor Frith. Cambridge University Press, 1991, págs. 37-92. DOI: 10.1017/CBO9780511526770.002.

[14] Dr. G.e. Ssucharewa. "Die schizoiden Psychopathien im Kindesalter. (Part 1 of 2)". En: *European Neurology* 60.3-4 (1926), págs. 235-247. DOI: 10.1159/000190478.

[15] Irina Manouilenko y Susanne Bejerot. "Sukhareva - Prior to Asperger and Kanner". En: *Nordic Journal of Psychiatry* 69.6 (2015), págs. 1761-1764. ISSN: 15024725. DOI: 10.3109/08039488.2015.1005022.

[16] S Wolff. "The first account of the syndrome Asperger described?" En: *European Child & Adolescent Psychiatry* 5.3 (1996), págs. 119-132. URL: https://doi.org/10.1007/BF00571671.

[17] Donald E Greydanus y Luis H Toledo-pereyra. "Historical Perspectives on Autism: It's Past Record of Discovery and It's Present State of Solipsism , Skepticism , and Sorrowful Suspicion". En: 59 (2012), págs. 1-11. DOI: 10.1016/j.pcl.2011.10.004.

[18] Roberto Chaskel y María Fernanda Bonilla. "Trastorno del espectro autista". En: *CCAP* 15.1 (), págs. 19-29.

[19] American Psychiatric Association. *Diagnostic and statistical manual of mental disorders: DSM-5*. 5th ed. Washington, DC: Autor, 2013.

[20] *Navegador CIE-11*. URL: https://icd.who.int/browse11/l-m/es#/http%5C%3a%5C%2f%5C%2fid.who.int%5C%2ficd%5C%2fentity%5C%2f437815624.

[21] *Magnetic Resonance Imaging (MRI)*. URL: https://www.nibib.nih.gov/science-education/science-topics/magnetic-resonance-imaging-mri (visitado 20-06-2021).

[22] Artem S. Belyaev y col. "Clinical Applications of Functional MR Imaging". En: *Magnetic Resonance Imaging Clinics of North America* 21.2 (2013), págs. 269-278. ISSN: 10649689. DOI: 10.1016/j.mric.2012.12.001. URL: http://dx.doi.org/10.1016/j.mric.2012.12.001.

[23] Kimberly Stigler y col. "Structural and functional magnetic resonance imaging of autism spectrum disorders". En: *Brain Res* 1380 (2011), págs. 146-161. DOI: 10.1016/j.brainres.2010.11.076.

[24] Gabriel S. Dichter. "Functional magnetic resonance imaging of autism spectrum disorders". En: *Dialogues in Clinical Neuroscience* 14.3 (2012), págs. 319-351. ISSN: 12948322. DOI: 10.31887/dcns.2012.14.3/gdichter.

[25] Olaf Sporns, Giulio Tononi y Rolf Kötter. "The human connectome: A structural description of the human brain". En: *PLoS Computational Biology* 1.4 (2005), págs. 0245-0251. ISSN: 15537358. DOI: 10.1371/journal.pcbi.0010042.

[26] Santiago Ramon y Cajal. "The structure and connexions of neurons. In Nobel Lectures Physiology or Medicine 1901-1921". En: (1906), págs. 220-253. URL: https://www.nobelprize.org/uploads/2018/06/cajal-lecture.pdf.

[27] Kamalaker Dadi y col. “Benchmarking functional connectome-based predictive models for resting-state fMRI”. En: *NeuroImage* 192.February (2019), págs. 115-134. ISSN: 10959572. DOI: 10.1016/j.neuroimage.2019.02.062.

[28] *1000 Functional Connectomes Project*. 2009. URL: https://www.nitrc.org/projects/fcon_1000.

[29] *IDA LONI*. 2003. URL: https://ida.loni.usc.edu/.

[30] Adriana Di Martino y Michael P. Milham. *ABIDE II*. 2016. URL: http://fcon_1000.projects.nitrc.org/indi/abide/abide_II.html (visitado 26-02-2021).

[31] *biac:analysis:resting*$_{p}ipeline[BrainImaging\&AnalysisCenter]$. URL: https://wiki.biac.duke.edu/biac:analysis:resting_pipeline (visitado 25-02-2021).

[32] F. Pedregosa y col. “Scikit-learn: Machine Learning in Python”. En: *Journal of Machine Learning Research* 12 (2011), págs. 2825-2830.

[33] Sun Weiran. *Hyper parameter Tuning with Grid Search for Deep Learning*. 2019. URL: https://towardsdatascience.com/hyper-parameter-tuning-with-randomised-grid-search-54f865d27926.

[34] *Welcome to Flask — Flask Documentation (2.0.x)*. URL: https://flask.palletsprojects.com/en/2.0.x/ (visitado 26-06-2021).

[35] *Bootstrap · The most popular HTML, CSS, and JS library in the world.* URL: https://getbootstrap.com/ (visitado 26-06-2021).

[36] C Craddock y col. “The Neuro Bureau Preprocessing Initiative: open sharing of preprocessed neuroimaging data and derivatives”. En: (2013).

Apéndice A
Entidades participantes en ABIDE

Entidad	MRIs en ABIDE I	MRIs en ABIDE II
Barrow Neurological Institute	-	58
California Institute of Technology	38	-
Carnegie Mellon University	27	-
Erasmus University Medical Center Rotterdam	-	54
ETH Zürich	-	37
Georgetown University	-	106
Indiana University	-	40
Institut Pasteur and Robert Debré Hospital	-	56
Katholieke Universiteit Leuven	-	28
Kennedy Krieger Institute	55	211
Ludwig Maximilians University Munich	57	-
NYU Langone Medical Center	184	105
Olin, Institute of Living at Hartford Hospital	36	59
Oregon Health and Science University	28	93
San Diego State University	36	58
Social Brain Lab BCN NIC UMC Groningen and Netherlands Institute for Neuroscience	30	-
Stanford University	40	42
Trinity Centre for Health Sciences	49	42
University of California Los Angeles	109	32
University of Leuven	64	-

University of Utah School of Medicine	101	33
University of Michigan	145	-
University of Pittsburgh School of Medicine	57	-
Entidad	MRIs en ABIDE I	MRIs en ABIDE II
University of California Davis	-	32
University of Miami	-	28
Yale Child Study Center	56	-

Apéndice B
Uso de la aplicación

En este apéndice se describirán los pasos necesarios para lanzar la aplicación web creada desde el ordenador de un usuario.

Antes de usar la aplicación, se debe tener en cuenta que, en caso de querer introducir un fMRI en ella, el sistema operativo con el que se esté trabajando debe de ser alguna distribución Linux o MacOS, pues las herramientas de FSL no son funcionales en Windows.

La aplicación se encuentra dentro del archivo comprimido *prTFG.zip*, que al ser descomprimido da lugar a un proyecto de Python. Este proyecto puede ser abierto con cualquier programa compatible, pero siempre desde el modo adminsitrador o *root*. Dentro de este proyecto encontramos un entorno virtual con una serie de carpetas. En concreto, la correspondiente a nuestra interfaz será la llamada *app*. Para poder ejecutar el archivo *pr_web.py* y que este monte correctamente la aplicación se necesitarán una serie de instalaciones.

B.1. Instalaciones

- Paquete *networkx*. Este paquete suele estar incluido en la instalación típica de Python, pero necesitaremos su versión 1.6.

- Librería *lsb-core* de Linux. Esta librería es necesaria para ejecutar la *pipeline*.

- Paquete *fslpy*. Este paquete debe instalarse para el correcto funcionamiento de la *pipeline*.

- Paquete *scikit-learn*. Es necesario para realizar las predicciones. En concreto, se necesita la versión *0.22.2.post1*.

El resto de componentes están ya incluidos en el proyecto.

B.2. Inicio de la aplicación

Para lanzar la aplicación, se debe ejecutar el archivo *pr_web.py*, lo que debe de devolver un mensaje que indique el puerto en el que se ha creado la aplicación. Introduciendo este puerto en nuestro navegador, podremos acceder a la página de inicio de la interfaz.

Una vez en esta página, tenemos un formulario que rellenar con los datos de nuestro paciente y su archivo a analizar. Es muy importante tener en cuenta que los modelos se han creado a partir de conectomas calculados por la *Python/FSL Resting State Pipeline*, y por tanto si se opta por introducir algún tipo de conectoma este debe de haberse calculado con la misma estrategia.

Teniendo esto en cuenta, la aplicación ya está lista para ser usada en el diagnóstico de TEA.

UNIVERSIDAD DE MÁLAGA | uma.es

E.T.S de Ingeniería Informática
Bulevar Louis Pasteur, 35
Campus de Teatinos
29071 Málaga

E.T.S. DE INGENIERÍA INFORMÁTICA