



Universidad Carlos III de Madrid  
Escuela Politécnica Superior

TRABAJO FIN DE GRADO

Detección de TOC en escenarios  
multicéntricos mediante el uso de PRoNTTo.

**Patricia Muñoz Pérez**

GRADO EN INGENIERÍA DE SISTEMAS AUDIOVISUALES

**Tutor: Vanessa Gómez Verdejo**

Junio 2015



## Agradecimientos

---

Este proyecto no habría sido posible sin el trabajo de mi tutora, Vanessa Gómez Verdejo. Es de agradecer su implicación y su disponibilidad durante todo el desarrollo del mismo.

Me gustaría agradecer a mis padres, Julián e Isabel, todo el apoyo que me han brindado durante estos duros años de estudio y su esfuerzo diario para que yo pueda haber estudiado lo que deseaba. Gracias por esta excelente herencia.

Y por último a mis compañeros que, tras estos años, muchos se han convertido en amigos. Gracias por todo.

## Resumen

---

Este Trabajo Fin de Grado pretende detectar, a través de imágenes de resonancia magnética estructural, la enfermedad neurológica del Trastorno Obsesivo Compulsivo (TOC). Para ello, se utiliza la herramienta de aprendizaje máquina PRoNTto (v.1.1), basada en Matlab.

Se trata de un problema de clasificación binaria donde tenemos datos de pacientes enfermos de TOC y datos de controles sanos. Como el objetivo es encontrar la mejor solución para nuestro problema, se han realizado diversos experimentos. En primer lugar, tratando de encontrar el modelo de clasificación más adecuado para nuestro caso. Los experimentos se han realizado con dos modelos de aprendizaje máquina: la máquina de vectores de soporte y el proceso gaussiano. Además, se ha tratado de evaluar las consecuencias de incluir algunos preprocesados de datos, ofrecidos por PRoNTto, y alguna máscara de selección de vóxeles que delimite las regiones de interés.

En este documento se describen todos los experimentos mencionados anteriormente, con base teórica, junto a sus resultados correspondientes.

**Palabras clave:** TOC, PRoNTto, imagen por resonancia magnética estructural, aprendizaje máquina, clasificación binaria, máquinas de vectores de soporte, proceso gaussiano.

# Índice general

---

1. Introducción.....	9
1.1. Motivación.....	9
1.2. Objetivos.....	10
1.3. Estructura del documento.....	11
2. Introduction.....	12
2.1. Motivations.....	12
2.2. Goals.....	13
2.3. The structure of this thesis.....	13
3. Estado del arte.....	15
3.1. Planteamiento del problema.....	15
3.1.1. Imágenes de resonancia magnética estructural.....	16
3.2. El aprendizaje máquina.....	17
3.2.1. Generalización y sobreajuste.....	18
3.2.2. Validación.....	19
3.3. Algoritmos de aprendizaje máquina.....	21
3.3.1. Las máquinas de vectores de soporte.....	21
3.3.2. Proceso Gaussiano.....	25
3.4. Evaluación de las prestaciones de clasificadores.....	28
3.4.1. Tasa de acierto.....	29
3.4.2. Predicciones.....	30
3.4.3. Curva ROC.....	30
3.4.4. Histograma de la salida del clasificador.....	31
3.4.5. Matriz de confusión.....	32
4. Diseño de la solución técnica.....	34
4.1. Descripción inicial de los datos.....	34
4.2. PRoNTTo v.1.1.....	34
4.2.1. Fases principales de PRoNTTo.....	35
4.2.2. Fases de verificación.....	38
4.2.3. <i>Batch</i> .....	40
4.3. Experimentos.....	41
4.3.1. Análisis de los modelos estadísticos.....	41
4.3.2. Análisis de métodos de preprocesado.....	46
4.3.3. Análisis de máscaras de selección.....	48
4.3.4. Estudio multicentro.....	49
5. Resultados finales.....	51
5.1. Evaluación de modelos de clasificación.....	51

5.2. Evaluación de preprocesados de datos.....	52
5.3. Evaluación de máscaras de selección.....	54
5.4. Evaluación multicentro.....	55
6. Planificación y presupuesto.....	58
6.1. Planificación del proyecto.....	58
6.1.1. Conceptualización.....	58
6.1.2. Planificación.....	58
6.1.3. Desarrollo y ejecución.....	59
6.1.4. Evaluación y cierre del proyecto.....	60
6.2. Presupuesto.....	61
6.2.1. Costes personales.....	61
6.2.2. Costes materiales.....	62
6.2.3. Presupuesto total.....	62
7. Conclusiones.....	63
8. Conclusions.....	64
Anexos.....	65
Anexo I. Instalación de PRoNTTo.....	65
Anexo II. Resumen en inglés del documento.....	66
Bibliografía.....	72

## Índice de figuras

---

- Figura 3.1.: Ejemplo de clasificación ajustada (parábola) y con sobreajuste (línea verde). Gráfica del curso *Machine Learning* de Andrew Ng.
- Figura 3.2.: Ejemplo de validación cruzada K-veces.
- Figura 3.3.: Ejemplo de validación cruzada “Leave One Out”.
- Figura 3.4.: Ejemplo de validación a través del muestreo aleatorio.
- Figura 3.5.: Múltiples hiperplanos de clasificación.
- Figura 3.6.: Hiperplano de clasificación SVM óptimo.
- Figura 3.7.: Ejemplo de las prestaciones del modelo evaluadas por PRoNTto.
- Figura 3.8.: Ejemplo gráfica con las salidas del clasificador para cada muestra de prueba en un caso binario.
- Figura 3.9.: Ejemplo de curva ROC cuya AUC es igual a 0.74.
- Figura 3.10.: Ejemplo de histograma de salida del clasificador de un problema binario.
- Figura 3.11.: Ejemplo de matriz de confusión para una clasificación binaria.
- Figura 4.1.: Ventana principal de la toolbox PRoNTto.
- Figura 4.2.: Ejemplo de imagen de mapa de pesos.
- Figura 4.3.: Ejemplo de estructura creada en el primer módulo principal de la herramienta.
- Figura 4.4.: Ejemplo de modelo creado con validación cruzada tipo LOO.
- Figura 4.5.: Ejemplo de ventana del módulo “Display Results”.
- Figura 4.6.: Ventana de la fase *Data & Design* de la toolbox PRoNTto.
- Figura 4.7.: Ventana principal de la segunda fase principal de PRoNTto.
- Figura 4.8.: Segunda ventana de la segunda fase principal de PRoNTto.
- Figura 4.9.: Ventana de la tercera fase principal de la herramienta.
- Figura 4.10.: Ventana donde se definen las clases del modelo.
- Figura 4.11.: Sección de validación cruzada y preprocesados de datos en PRoNTto.
- Figura 4.12.: Búsqueda del archivo correspondiente a la segunda máscara.
- Figura 5.1.: Tabla estadística resultado de experimentar con el centro 2 y el segundo preprocesado en la *toolbox* PRoNTto.

## Índice de tablas

---

- Tabla 4.1.: Conjunto de datos de tipo sMRI de los que se disponen.
- Tabla 5.1.: Tasas de acierto obtenidas con cada máquina de clasificación para cada centro.
- Tabla 5.2.: Tasas de acierto obtenidas con las distintas operaciones para cada centro.
- Tabla 5.3.: Resultados obtenidos al utilizar distintas segundas máscaras de selección de vóxeles.
- Tabla 5.4.: Resultados obtenidos utilizando los datos de un centro para entrenar y los de otro para testear.
- Tabla 6.1.: Duración en semanas de cada tarea de los primeros 5 meses de trabajo.
- Tabla 6.2.: Duración en semanas de cada tarea de los últimos 5 meses de trabajo.
- Tabla 6.3.: Costes personales del proyecto.
- Tabla 6.4.: Costes materiales del proyecto.
- Tabla 6.5.: Presupuesto total del proyecto (sin IVA).



# Capítulo 1

## Introducción

---

### 1.1. Motivación

El trastorno obsesivo compulsivo (TOC) es una enfermedad neurológica que afecta a un gran número de personas en todo el mundo. Actualmente existen diversos tratamientos para aliviar a los enfermos que lo padecen, como terapias cognitivo-conductuales o medicación antidepresiva [22]. Sin embargo, las investigaciones sobre nuevos fármacos que produzcan mejores resultados y sobre las causas cerebrales que provocan esta enfermedad van en aumento.

El análisis de enfermedades como el TOC se lleva a cabo a través de técnicas de neuroimagen, como imágenes por resonancia magnética funcional (*functional Magnetic Resonance Imaging*, fMRI) o imágenes por resonancia magnética estructural (*structural Magnetic Resonance Imaging*, sMRI). FMRI es una técnica para medir la actividad cerebral que funciona mediante la detección de cambios en el flujo y la oxigenación de la sangre que se producen en respuesta a la actividad neuronal [23]. SMRI, en cambio, es una técnica para examinar la estructura física del cerebro.

El estudio de estas imágenes por parte de un médico o investigador de forma visual resulta costoso en cuanto a tiempo y esfuerzo. Para simplificar este trabajo son necesarias técnicas capaces de procesar dichas imágenes de manera automática.

Por otro lado, las técnicas de aprendizaje máquina (*Machine Learning*, ML) han sufrido un gran desarrollo en los últimos años. Estas técnicas consisten, principalmente, en una variedad de métodos estadísticos y de optimización que permiten a los ordenadores "aprender" de ejemplos pasados. Su principal utilidad es detectar patrones difíciles de discernir a partir de grandes conjuntos de datos, ruidosos o complejos.

Si obtenemos datos de pacientes de TOC, como imágenes por resonancia magnética estructural, podemos utilizarlos en herramientas de ML para categorizar los patrones cerebrales de dicha enfermedad al compararlos con otros datos de personas sanas, o controles. Las técnicas de ML son una buena opción para procesar las imágenes y cubrir las necesidades que se comentaban anteriormente. En la actualidad ya se utilizan estas técnicas y, cada vez, con más frecuencia.

PRoNTTo es una herramienta que utiliza técnicas de ML. Es una unión de ambas ciencias, la medicina y la ingeniería, muy interesante y que puede aportar grandes beneficios. Por este motivo, se ha considerado la realización del estudio de esta herramienta con datos sMRI de enfermos de TOC y controles de hasta 6 centros diferentes en este proyecto.

La *toolbox* permite experimentar con diferentes modelos de clasificación, preprocesados de datos y máscaras de selección de vóxeles para realizar una categorización de los datos introducidos que puede llegar a ser muy satisfactoria. Esto puede facilitar al médico el diagnóstico de este tipo de enfermedades neurológicas y consiguiendo que el paciente obtenga antes el tratamiento correspondiente.

El cerebro es uno de los órganos más complejos y, por ello, más desconocido. Con técnicas y herramientas como las empleadas en este proyecto se pueden obtener conclusiones de gran valor para la comunidad médica y científica que lo investigan.

## 1.2. Objetivos

En base a las necesidades anteriores, este Trabajo de Fin de Grado (TFG) trata de dar soluciones cubriendo los siguientes objetivos:

El primero de ellos es aprender a utilizar la herramienta PRoNTTo, y conocer al detalle todas sus utilidades. Conseguir un buen manejo de ésta que permita manejarse por ella con total seguridad y firmeza, conociendo los pasos a seguir según la máquina a utilizar, y ser capaz de localizar, consultando el código MATLAB si es necesario, los errores que puedan surgir durante su uso.

Otro de los objetivos a lograr es el análisis de las prestaciones de los métodos de clasificación que ofrece la herramienta, así como conocer en profundidad la selección de vóxeles que permite y que facilitará el análisis de las áreas relevantes para la clasificación y, por tanto, para analizar las causas del diagnóstico. Además de analizar el uso de preprocesados de datos, comprobar la viabilidad de incluir una segunda máscara de selección de vóxeles y saber valorar los resultados.

Como objetivo principal, cabe destacar la detección de trastorno obsesivo compulsivo (TOC) a través de estudios multicentro. Para ello será necesario realizar diversas pruebas con los datos sMRI, tanto de pacientes como de controles, de los que se disponen, que en este caso son de hasta 6 centros distintos.

## 1.3. Estructura del documento

Con objeto de mejorar la lectura de la memoria, se incluye a continuación un breve resumen de cada capítulo.

- **Capítulo 1:** en el presente capítulo se expone la motivación del proyecto y se definen los objetivos que se pretenden conseguir, así como la estructura de este documento.
- **Capítulo 2:** este capítulo contiene la misma información que el anterior pero, esta vez, en inglés.
- **Capítulo 3:** se ofrece una visión general que ayuda al lector a situarse y conocer detalles del proyecto. Se presentan los tipos de datos de los que se disponen y se explica detalladamente que es ML y su relación con este estudio. A continuación, se exponen teóricamente dos modelos estadísticos utilizados en problemas de ML y que, además, forman parte de los experimentos de este proyecto. Para finalizar esta sección se describen una serie de medidas de evaluación típicas y en las que, posteriormente, se apoyará la evaluación de los experimentos realizados.
- **Capítulo 4:** contiene una presentación de PRoNTTo, la herramienta utilizada en el desarrollo del proyecto, así como una explicación de todas sus funcionalidades. El capítulo continúa con varias secciones donde se describen de forma detallada los pasos a seguir para realizar los distintos experimentos llevados a cabo.
- **Capítulo 5:** es en el que se exponen y analizan todos los experimentos reales efectuados junto con sus resultados.
- **Capítulo 6:** muestra la planificación seguida durante el proyecto. También incluye una descripción del presupuesto del proyecto.
- **Capítulo 7:** se ofrecen unas conclusiones globales y una valoración de los objetivos cumplidos.
- **Capítulo 8:** este capítulo contiene la misma información que el anterior pero, esta vez, en inglés.

Para finalizar la memoria se incluyen una serie de **Anexos** que pueden ser de utilidad para el lector. Estos anexos ofrecen una guía para instalar la herramienta e incluyen un resumen de todo el documento.

# Capítulo 2

## Introduction

---

### 2.1. Motivations

Obsessive compulsive disorder (OCD) is a neurological disease that affects a large number of people around the world. Currently, there are various treatments for patients who suffer it, as cognitive behavioral therapy or antidepressant medication [22]. However, research on new medicines, which produce better results, and studies of the brain causes of this disease are increasing.

The analysis of diseases, like OCD, is performed through neuroimaging techniques such as functional magnetic resonance imaging (fMRI) or structural Magnetic Resonance Imaging (sMRI). fMRI, is a technique for measuring brain activity. It works by detecting the changes in blood oxygenation and flow that occur in response to neural activity [23]. However, sMRI is a technique to examine the physical structure of the brain.

The study visual of these images by a doctor or researcher is costly in terms of time and effort. To simplify this work techniques capable to process these images automatically are necessary.

On the other hand, machine learning (ML) techniques have had a great development in recent years. These techniques mainly consist in a variety of statistical and optimization methods that allows computers to “learn” from past examples and to detect hard-to-discern patterns from large, noisy or complex data sets.

If you get a set data from OCD patients, as structural magnetic resonance imaging (sMRI), we can use it in the tools of machine learning, and thus, categorize brain patterns of the disease when compared with other data or healthy controls. Techniques of ML are a good option for processing images and for meet the needs that were discussed above. At present, and increasingly more often, these techniques are used.

PRoNTo is a tool that uses machine learning techniques. It is a union of both sciences, medicine and engineering, very interesting and can bring great benefits. For

this reason, it was considered the study of this toolbox, with a set of sMRI data of OCD sufferers and controls. This set of data is from 6 different clinical centers.

The toolbox allows you to experiment with different classification models, data preprocessing and voxel selection masks. Thanks to that, the results of this categorization can be very satisfactory. This can provide an effective way to automate the doctor diagnose of this type of neurological diseases and getting sooner the appropriate treatment for the patient.

The brain is one of the most complex and unknown organs. Through techniques and tools, like those used in this project, can be obtained valuable conclusions for the medical and scientific community.

## **2.2. Goals**

Based on the above needs, this project aims to give them solution covering the following objectives:

The first one is to learn to use the toolbox PRoNTo, and know in detail all their profits. Get a good handle on it that allows use with confidence and firmness. Being able to locate errors that may arise during use, taking a look at the MATLAB code if necessary.

Another goal to achieve is to analyze the performance of classification methods provided by the toolbox, as well as learn more about the selection of voxels allowed. This selection of voxels will facilitate the analysis of the relevant areas for classification and, therefore, to analyze the causes of diagnosis. In addition to analyzing the use of operations to apply to the data, test the feasibility of including a second mask selection of voxels and know how to evaluate the results.

Main objective included the detection of obsessive compulsive disorder (OCD) through multicenter studies. This will require performing various tests with the sMRI (Structural Magnetic Resonance Imaging) data, both patients and controls, which are provided, which in this case are up to 6 different centers. To check the quality of the classification of such images different methods are used and preprocessed offered by PRoNTo and finally, the results are validated.

## **2.3. The structure of this thesis**

In order to improve the reading of this bachelor thesis, a brief summary of each chapter is included.

- **Chapter 1:** in this chapter the motivation of the project is exposed and the objectives to be achieved. In addition the structure of this document it is presented.
- **Chapter 2:** has the same content as the previous chapter, but this time in English.
- **Chapter 3:** provides an overview to help the reader to know details of the project. It gives information about the data used and it spells out what is ML and its relationship to this study. The next section talks about two statistical models used in ML problems and also they are part of the experiments of this project. To end this chapter, a typical series of “evaluation measures in ML are described. Subsequently these measures will support the evaluation of the experiments.
- **Chapter 4:** contains a description about PRoNTTo, the toolbox used in this project, and an explanation of all its features. The chapter continues with several sections which describe the steps to perform the experiments of this work.
- The document continues with **chapter 5** in which all real experiments performed are disclosed and analyzed together with their results.
- **Chapter 6:** shows the complete project planning. It also includes a description of the project budget.
- In **Chapter 7:** some overall conclusions and an evaluation of the goals accomplished are offered.
- **Chapter 8:** has the same content as the previous chapter, but this time in English.

At the end of the document it includes a series of **annexes** which may be useful to the reader. These annexes provide a guide to install the toolbox and a summary of the entire document.

## Capítulo 3

# Estado del arte

---

Este capítulo revisa, desde un punto de vista teórico, las técnicas y algoritmos que han sido empleados a lo largo de este TFG. Además, describe como las técnicas de ML pueden emplearse en problemas de neuroimagen y, concretamente, en la detección del trastorno obsesivo compulsivo.

En la primera sección se ofrece una visión general del problema objeto de estudio, la detección de TOC, y de las técnicas que existen actualmente para su resolución. En la segunda sección se trata en detalle el ML, qué es y que utilidades tiene. En la tercera sección se describen dos algoritmos de clasificación utilizados típicamente en herramientas de ML, la máquina de vectores de soporte y el proceso Gaussiano. Por último, se explicarán las medidas de evaluación habituales en ML y que PRoNTo ofrece, las cuales utilizaremos posteriormente en el análisis de los experimentos realizados.

### 3.1. Planteamiento del problema

Desde que la tecnología MRI está disponible para la psiquiatría y neurología, las imágenes de resonancias magnéticas han tenido un impacto considerable en la comprensión de las alteraciones en la función y estructura del cerebro [24]. Las técnicas de neuroimagen permiten el estudio del cerebro humano en vivo. Los procedimientos originarios, como la arteriografía carotídea y la neumoencefalografía eran complicadas y no exentas de riesgo para el paciente [27]. La neumoencefalografía se practicaba mediante la sustitución del líquido cefalorraquídeo por la inyección de aire, al interior de los ventrículos cerebrales y permitía visualizar las cavidades normalmente bañadas por dicho fluido corporal. Las técnicas actuales de neuroimagen miden o valoran un aspecto o característica del cerebro y traducen esa información en una imagen útil para el médico o investigador. Debido a esto, podemos afirmar que estas nuevas técnicas, además de no poner en riesgo al paciente, ofrecen unos datos más claros y útiles de la estructura y actividad cerebral.

En un principio, se empezó a utilizar la técnica de análisis de las regiones de interés (ROIs). El objetivo principal de las ROIs es aislar un volumen de datos concreto, del volumen total, para centrar el estudio en dicha área. Sin embargo, la aplicación universal de este método implica un conocimiento completo de todas las regiones del cerebro implicadas en las funciones cerebrales normales, y de cuando la función o estructura del cerebro es anormal. Teniendo en cuenta que aún estamos lejos de un

estado de conocimiento tal, eran, y son, necesarios métodos capaces de explorar los cambios de actividad a nivel de vóxel.

A principios de 1990, Karl Friston y su equipo comenzaron a desarrollar métodos para el análisis de los cambios en la activación cerebral sobre todo el cerebro, lo que llevó al desarrollo del paquete conocido como Mapeo Paramétrico Estadístico (*Statistical Parametric Mapping*, SPM).

Con la aparición de la imagen por resonancia magnética funcional (*functional Magnetic Resonance Imaging*, fMRI), el paquete SPM fue adaptado rápidamente para hacer frente a las diferentes características de los nuevos conjuntos de datos.

Algo más tarde, la posibilidad de analizar de manera similar los cambios estructurales vóxel por vóxel llevó al desarrollo de lo que hoy se conoce como la morfometría basada en vóxel (*Voxel-Based Morphometry*, VBM). SPM se aplicó rápidamente a un gran número de proyectos de imágenes cerebrales estructurales y funcionales.

Recientemente el interés se ha dirigido hacia un amplio grupo de técnicas conocidas como ML. La idea de estos métodos es que, en lugar de analizar el cerebro vóxel por vóxel, los datos de grupos de vóxeles (ROIs) o incluso de todo el cerebro, se utilizan para entrenar un programa de ordenador. Este programa debe ser capaz de generalizar comportamientos a partir de los datos suministrados en forma de ejemplos.

El problema mental que se trata de detectar en este proyecto es el trastorno obsesivo-compulsivo (TOC). Es un trastorno de ansiedad caracterizado por pensamientos intrusivos y angustiosos, así como comportamientos repetitivos dirigidos a disminuir las molestias causadas por esos pensamientos obsesivos [1]. Este problema mental puede presentarse a cualquier edad, pudiendo producir una importante discapacidad por lo que su diagnóstico y tratamiento son muy importantes. Las herramientas de ML pueden ser de gran ayuda para detectar dicha enfermedad a través de imágenes por resonancia magnética estructural.

### **3.1.1. Imágenes de resonancia magnética estructural**

Las imágenes de resonancia magnética estructural (*Structural Magnetic Resonance Image*, sMRI) son una técnica para el examen de la estructura física del cerebro (en contraposición a la utilización de imágenes de resonancia magnética funcional, fMRI, para examinar la actividad cerebral). Se trata de una técnica no invasiva ya que no utiliza radiaciones ionizantes convirtiéndola en una técnica segura para el paciente.



Muchas secuencias de escaneo sMRI son volumétricas, lo que significa que se pueden hacer medidas de estructuras cerebrales específicas para calcular los volúmenes de tejido [19]. Concretamente, los datos sMRI proporcionan información para describir cualitativa y cuantitativamente la forma y el tamaño de las estructuras de las materias blanca y gris en el cerebro. En términos generales, la señal de MRI varía a través de tipos de tejidos porque la materia gris contiene más cuerpos celulares (por ejemplo, neuronas) que la materia blanca, que se compone principalmente de fibras nerviosas de largo alcance, junto con el apoyo de células gliales [20].

Estos datos obtenidos tras la realización al paciente de la resonancia magnética no son más que imágenes en 3 dimensiones compuestas por vóxeles, o píxeles volumétricos. Un vóxel es el elemento distinguible más pequeño de una imagen 3D, siendo el equivalente del píxel en 2D.

Existen evidencias considerables de que algunas enfermedades cerebrales están directamente relacionadas con una variación anormal del volumen de ciertas regiones cerebrales por lo que este tipo de datos resultan de gran utilidad, tanto para el diagnóstico como para el tratamiento de esas enfermedades, y son objeto de diversas investigaciones.

## **3.2. El aprendizaje máquina**

El campo de ML se ocupa de cómo construir programas informáticos empleando técnicas estadísticas, probabilísticas y de optimización que permiten a los ordenadores “aprender” de ejemplos pasados y detectar patrones difíciles de discernir a partir de grandes conjuntos de datos, ruidosos o complejos. Ese aprendizaje implica la adquisición de conceptos generales a partir de ejemplos específicos de entrenamiento [2].

En los últimos años se han desarrollado muchas aplicaciones de ML que han resultado exitosas, desde programas de minería de datos que aprenden a detectar transacciones fraudulentas de tarjetas de crédito hasta sistemas de filtrado de información que aprenden las preferencias de lectura de los usuarios. Al mismo tiempo, se han producido importantes avances en la teoría y algoritmos que forman los cimientos de estos métodos.

Dependiendo de la utilización de etiquetas durante el entrenamiento, los algoritmos de ML se pueden clasificar en tres tipos: aprendizaje supervisado, aprendizaje semi-supervisado, y sin supervisión.

En el aprendizaje supervisado, cada muestra contiene dos partes: una es las observaciones de entrada o características y el otro es observaciones o etiquetas de salida [7]. Por lo general, las observaciones de entrada son las causas y las observaciones de salida son efectos, o la asociación de cada dato, o una categoría en problemas de clasificación. El propósito del aprendizaje supervisado es deducir una relación funcional a partir de datos de entrada que infiera las observaciones de salida. La forma de la relación es un conjunto de ecuaciones y coeficientes numéricos o pesos. Ejemplos de aprendizaje supervisado incluyen clasificación, regresión y aprendizaje por refuerzo.

En el aprendizaje no supervisado, sólo tenemos un conjunto de observaciones y no hay información de la etiqueta para cada muestra [8]. Por lo general, estas observaciones o características se deben a un conjunto de variables no observadas o latentes. El propósito principal del aprendizaje no supervisado es descubrir las relaciones entre las muestras o revelar las variables latentes detrás de las observaciones. Ejemplos de aprendizaje no supervisado incluyen agrupación, estimación de la densidad, y la separación ciega de fuentes.

El aprendizaje semi-supervisado se sitúa entre el aprendizaje supervisado y el no supervisado [9]. Se utilizan tanto los datos etiquetados (por lo general unos pocos) como los datos sin etiqueta (por lo general muchos) durante el proceso de entrenamiento. Este aprendizaje se da en aplicaciones en las que el etiquetado de los datos es muy costoso. Ejemplos de aprendizaje semi-supervisado incluyen clasificación semi-supervisada y sistemas de recomendación de información [10].

### **3.2.1. Generalización y sobreajuste**

Cuando se evalúa la calidad de un modelo de clasificación, es importante medir el error en el conjunto de entrenamiento y en la predicción. Utilizar exclusivamente el error del conjunto de entrenamiento puede conducir a resultados engañosos.

Idealmente, para probar una hipótesis se utilizarían todas las variaciones posibles [28], es decir, si tomamos como ejemplo el caso objeto de estudio de este TFG lo ideal sería obtener imágenes de resonancia magnética estructural focalizadas en el cerebro de toda la población. Al no ser posible, debido al gran volumen de datos que supondría, se elige un grupo representativo. En cualquier experimento, la generalización provoca ciertos problemas [29]. Uno de ellos radica en el hecho de que el grupo seleccionado no sea totalmente representativo del conjunto total. Siguiendo con el ejemplo, si las sMRI de las que se disponen corresponden a gente de una misma edad las estadísticas resultantes pueden no ser representativas de toda la población. Otro punto importante a tener en cuenta es el tamaño del grupo seleccionado. Debe ser suficiente para que las estadísticas resultantes puedan ser extrapoladas a toda la población. Teniendo en cuenta

estos problemas que puede conllevar la generalización, y construyendo un buen conjunto de datos representativo se puede obtener un buen modelo de clasificación.

Existe otro problema asociado a perder la capacidad de generalizar. Se produce un sobreajuste cuando el modelo estadístico “memoriza” los datos en lugar de “aprender” de ellos. En ese caso, el algoritmo de aprendizaje automático queda ajustado a unas características muy concretas de los datos de entrenamiento. Puede dar lugar a error y pensar que la clasificación está siendo buena ya que la tasa de acierto al clasificar datos con los que ha entrenado es muy alta. El modelo se ajusta muy bien a dicho datos. Sin embargo, con nuevas muestras es el error el que aumenta. El rendimiento para clasificar nuevos datos es muy pobre.

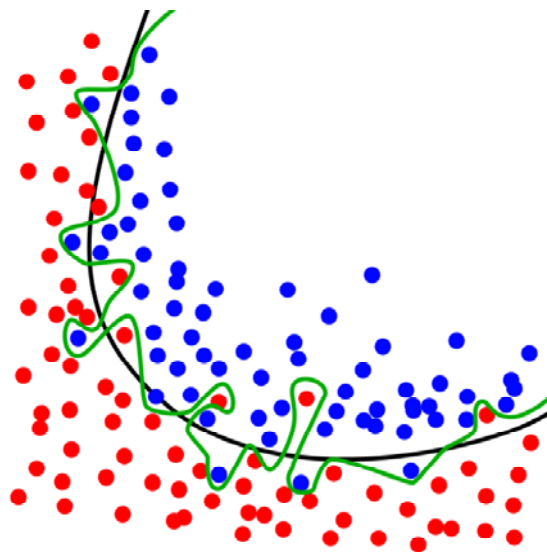


Fig. 3.1. Ejemplo de clasificación ajustada (parábola) y con sobreajuste (línea verde). Gráfica del curso *Machine Learning* de Andrew Ng.

### 3.2.2. Validación

Un algoritmo de aprendizaje automático, por lo general, será entrenado en un conjunto de datos de entrenamiento y testado en un conjunto de datos distinto, sobre los que debe generalizar. Para evaluar los algoritmos de aprendizaje automático en un conjunto de datos en particular y analizar su generalización, a menudo se utiliza el método de validación cruzada. Este método consiste en repetir y calcular la media aritmética obtenida de las medidas de evaluación sobre diferentes particiones del conjunto de datos. Entre las estrategias de partición más populares se encuentran la validación cruzada K-veces, “Leave-one-out” y el muestreo aleatorio.

En la validación K-veces, todo el conjunto de datos disponible se divide en K subconjuntos. El algoritmo de aprendizaje a evaluar es entrenado usando K-1 de los subconjuntos y evaluado en el restante. Este procedimiento se lleva a cabo K veces hasta que todos los subconjuntos de K han sido empleados como conjunto de test (véase Fig. 3.2). [26].

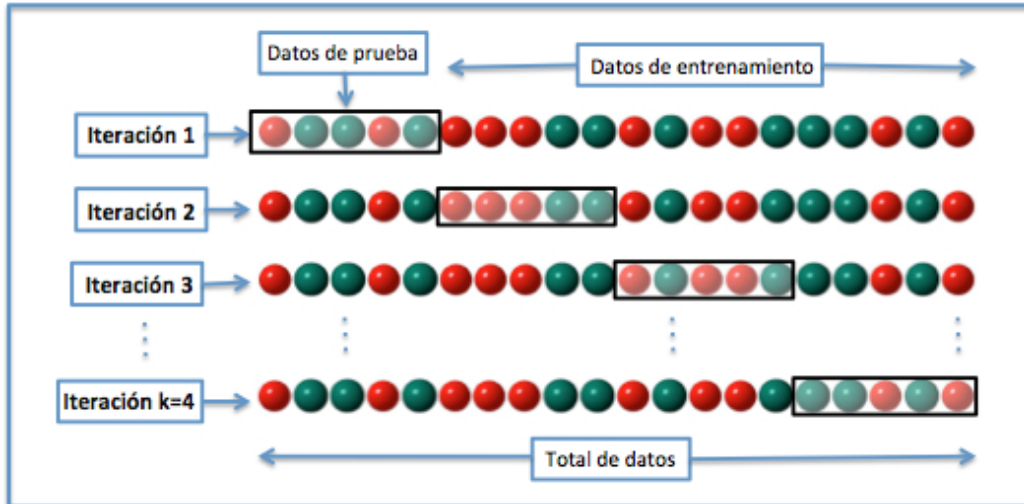


Fig. 3.2. Ejemplo de validación cruzada K-veces.

El procedimiento de “Leave-One-Out” (LOO) es similar a la validación anterior pero, en este caso, cada subconjunto se compone solo de una muestra del conjunto total de datos. Es decir, en cada iteración tenemos solo una muestra para los datos de test y el resto forman los datos de entrenamiento. Este proceso se repite N veces siendo N el número de muestras en el conjunto de datos (véase Figura 3.3). Este procedimiento es muy utilizado cuando el conjunto de entrenamiento dispone es de pocos datos (centenas o menos) ya que, de esta manera, se entrenan tantos ejemplos como sean posibles. Así, el aprendizaje de la máquina será mayor.

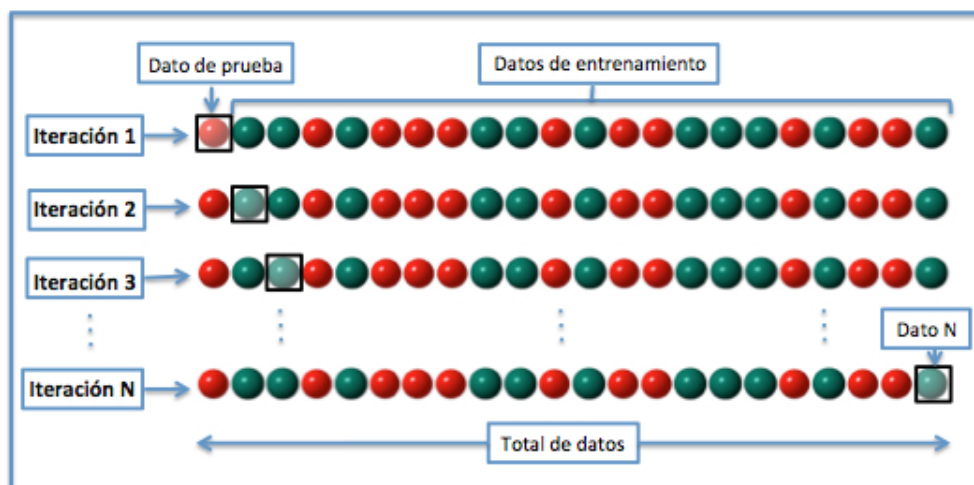


Fig. 3.3. Ejemplo de validación cruzada “Leave One Out”.

Para el muestreo aleatorio, los subconjuntos de datos están formados por un muestreo aleatorio de los datos sin sustitución como muestras de entrenamiento y los datos restantes sirven como datos de prueba [25]. Con este método existe el riesgo de que algunas muestras nunca formen parte del conjunto de datos de prueba, y se queden sin evaluar, y que otras muestras se evalúen más de una vez (véase Figura 3.4.). Es posible que se desperdicien datos.

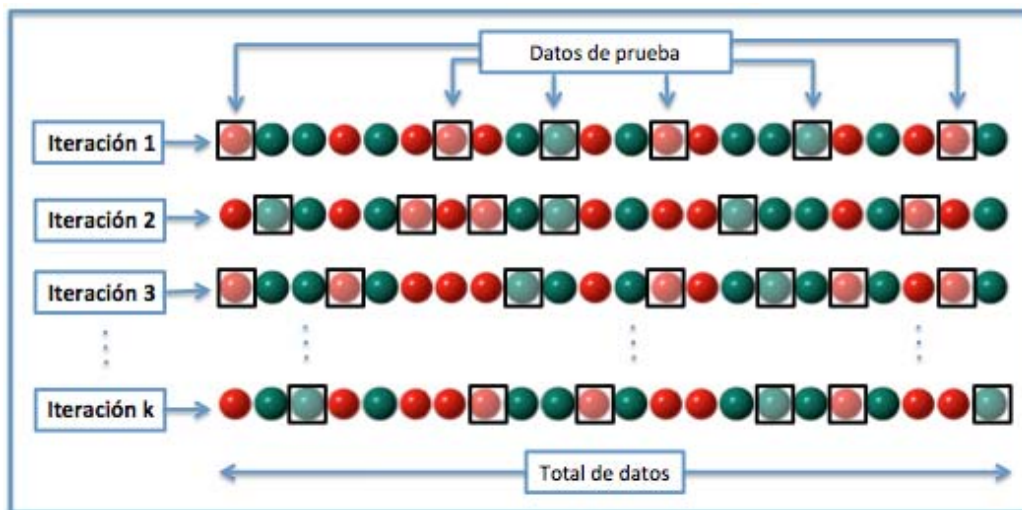


Fig. 3.4. Ejemplo de validación a través del muestreo aleatorio.

### 3.3. Algoritmos de aprendizaje máquina

Los algoritmos óptimos de ML pueden ser muy útiles y gracias a ellos podemos obtener una clasificación de datos de calidad. En los próximos apartados se explican dos de los principales algoritmos de aprendizaje máquina que, además, se han utilizado en los experimentos para detectar TOC realizados en este proyecto.

#### 3.3.1. Las máquinas de vectores de soporte

En ML, las máquinas de vectores de soporte (*Support Vector Machines*, SVM) son un conjunto de algoritmos de aprendizaje supervisado utilizados para resolver problemas de clasificación y regresión [4]. Nos centraremos en las SVM para problemas de clasificación, ya que el problema objeto de estudio en este TFG pertenece a este grupo.

El objetivo de las SVM es igual al de todos los métodos de ML y consiste en generar un modelo basado en los datos de entrenamiento que sea capaz de predecir la

clase a la que pertenecen los datos de un conjunto a partir de sus atributos. Para lograrlo, este algoritmo construye un hiperplano óptimo, o función de decisión óptima, buscando la máxima distancia entre éste y el punto de datos de entrenamiento más cercano de cualquiera de las dos clases ya que cuanto mayor sea el margen menor será el error de generalización del clasificador.

A continuación, se detallarán teóricamente el principio de funcionamiento de estas máquinas para problemas de clasificación binarios. Comenzaremos con el caso lineal separable por sencillez para después generalizar al caso lineal no separable. Hay extensiones a casos no lineales que no se describen debido a que no se utilizan en este TFG.

### Caso linealmente separable

La notación que seguiremos será principalmente la seguida en [Vapnik, 1995]. Supongamos que se nos dan  $l$  observaciones. Cada observación se compone de: un vector  $\mathbf{x}_i \in R^n, i = 1, \dots, l$  y la “verdad” asociada  $y_i$ , dada por una fuente de confianza. Siendo los datos de entrenamiento:

$$\{\mathbf{x}_i, y_i\}, i = 1, \dots, y_i \in \{-1, 1\}, \mathbf{x}_i \in R^d \quad (1)$$

Si queremos diseñar un clasificador lineal, nos encontramos con muchos hiperplanos que podrían clasificar los datos, como se puede observar en la Figura 3.5.

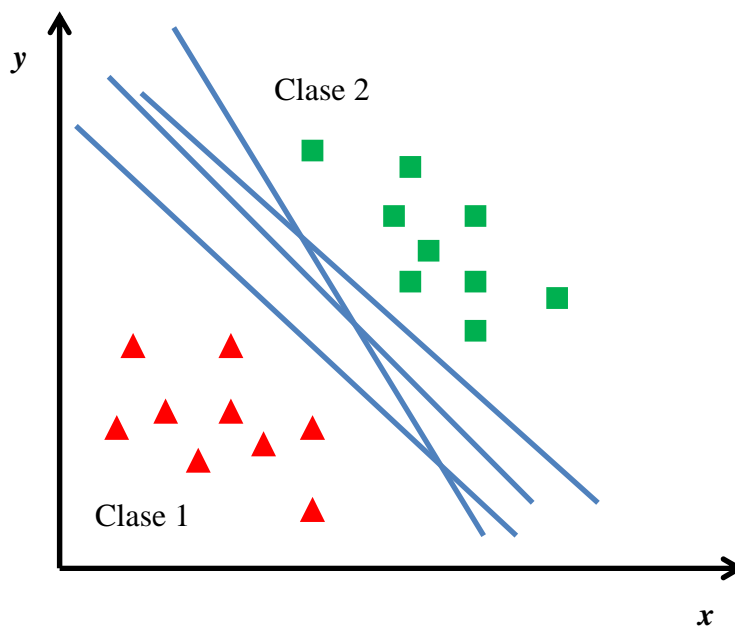


Fig. 3.5. Múltiples hiperplanos de clasificación.

La función de decisión se define con la siguiente ecuación:

$$\mathbf{w}^T \mathbf{x}_i + b = 0 \quad (2)$$

donde  $\mathbf{w}$  es normal al hiperplano,  $|b|/\|\mathbf{w}\|^2$  la distancia perpendicular desde el hiperplano hasta el origen, y  $\|\mathbf{w}\|$  la norma euclídea de  $\mathbf{w}$ . El entrenamiento del clasificador busca estimar los parámetros  $\mathbf{w}$  y  $b$ .

Las máquinas de vectores de soporte, para el caso linealmente separable, simplemente buscan el hiperplano de separación con mayor margen, lo que se puede formular como sigue a continuación, y asumiendo que todos los datos de entrenamiento satisfacen las siguientes restricciones:

$$\mathbf{x}_i \mathbf{w} + b \geq +1 \text{ para } y_i = +1 \quad (3)$$

$$\mathbf{x}_i \mathbf{w} + b \leq -1 \text{ para } y_i = -1 \quad (4)$$

Si combinamos estas dos desigualdades obtenemos el siguiente conjunto:

$$y_i(\mathbf{x}_i \mathbf{w} + b) \geq 1 \quad \forall_i \quad (5)$$

Optimizar el margen geométrico, es decir, hacer que el margen sea maximizado, implicará minimizar  $\|\mathbf{w}\|^2$ . La SVM en este caso se formula como sigue a continuación:

$$\begin{aligned} & \min \|\mathbf{w}\|^2 \\ & \text{s. t. } y_i(\mathbf{w}^T \mathbf{x} + b) \geq 1 \quad \forall_i \end{aligned} \quad (6)$$

Para ello se buscan dos hiperplanos paralelos (H1 y H2), con la misma normal, al hiperplano óptimo (H), sin que los puntos que representan los datos de entrenamiento caigan entre ellos. Puede que algunos de estos puntos se sitúen sobre H1 o H2. A esos puntos se les llama vectores de soporte. En la Figura 3.6 se puede observar una representación en dos dimensiones.

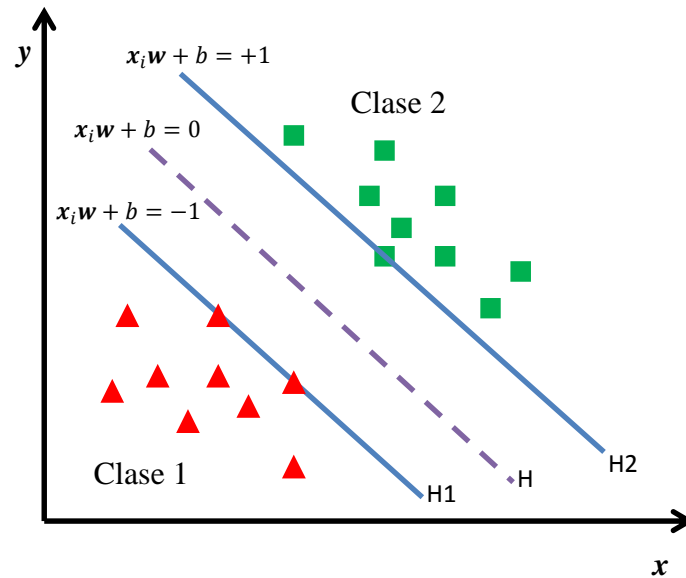


Fig. 3.6. Hiperplano de clasificación SVM óptimo.

### Caso linealmente no separable

En el caso de tener datos no separables linealmente, es decir, cuando no se puede trazar una frontera lineal que separe ambas clases, la formulación anterior no es aplicable. Debemos conseguir relajar las restricciones anteriores (3) y (4) cuando sea necesario, lo que se consigue introduciendo unas variables de holgura  $\xi$ ,  $i=1, \dots, l$  que permiten que algunos de los datos de entrenamiento queden mal clasificados. Las desigualdades anteriores quedan como sigue a continuación:

$$x_i \mathbf{w} + b \geq +1 - \xi_i \text{ para } y_i = +1 \quad (7)$$

$$x_i \mathbf{w} + b \leq -1 + \xi_i \text{ para } y_i = -1 \quad (8)$$

$$\xi_i \geq 0 \quad \forall_i$$

Cuando un dato queda mal clasificado, el correspondiente  $\xi_i$  toma un valor superior a la unidad, por lo que  $\sum_i \xi_i$  es un límite superior sobre el número de errores de entrenamiento. Por esto, una forma de asignar un coste extra para los errores es cambiar la función objetivo, quedando de la siguiente manera:

$$\begin{aligned}
 & \min \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_i \xi_i \\
 & \text{s. t. } y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i \quad \forall \xi_i > 0 \quad (9)
 \end{aligned}$$



donde  $C$  es un parámetro a elegir por el usuario. Un valor de  $C$  muy elevado asigna una alta penalización a los errores.

### 3.3.2. Proceso Gaussiano

El proceso de Gauss (*Gaussian Process*, GP) es una técnica de aprendizaje máquina no paramétrica para la construcción de modelos probabilísticos de problemas reales. Los procesos Gaussianos se basan en distribuciones gaussianas multivariantes. Pueden ser aplicados a problemas de regresión y de clasificación. Mientras que la salida para la clasificación son etiquetas de datos discretos, la regresión se centra en la predicción de variables continuas [6]. A continuación se explicará el desarrollo para problemas de regresión ya que, de esta forma, la posterior explicación de problemas de clasificación binarios será más sencilla.

#### 1) Regresión

Cuando muestreamos a partir de un GP, lo que obtenemos son funciones que siguen una distribución gaussiana [31].

La función  $f(x)$  se considera una variable aleatoria con distribución, a priori, gaussiana. Esta función queda completamente descrita por una media  $m$  y una función de covarianza (*kernel*), de tal manera que:

$$m(x) = \mathbb{E}[f(x)] \quad (10)$$

$$k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x')))] \quad (11)$$

A menudo los datos son ruidosos, o se producen errores de medición. Cada observación  $y$  se puede relacionar con  $f(x)$  a través de un modelo de ruido gaussiano:

$$y = f(x) + \varepsilon \quad (12)$$

donde  $\varepsilon$  es un ruido modelado por una gaussiana de media nula y varianza  $\sigma_n^2$ :  $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$ .

Sabemos que para  $f(x)$  la máxima covarianza admisible es definida como  $\sigma_f^2$ , que  $l$  es la longitud y que la función de covarianza es:

$$k(x, x') = \sigma_f^2 \exp\left[\frac{-(x-x')^2}{2l^2}\right] + \sigma_n^2 \delta(x, x') \quad (13)$$

Para preparar el proceso de Gauss es necesario calcular la covarianza entre todas las posibles combinaciones de puntos, lo que queda resumido en las siguientes matrices:

$$K = \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \cdots & k(x_n, x_n) \end{bmatrix}$$

$$K_* = [k(x_*, x_1) \quad \cdots \quad k(x_*, x_n)] \quad K_{**} = k(x_*, x_*)$$

y definir los valores de salida como  $y = (y_1, \dots, y_n)^T$ .

Del modelo anterior, los valores de salida se distribuyen de acuerdo a:

$$\begin{pmatrix} y \\ f(x_*) \end{pmatrix} \varepsilon \sim \mathcal{N} \left( 0, \begin{pmatrix} K + \sigma_n^2 & k_* \\ k_*^T & k_{**} \end{pmatrix} \right)$$

Para predecir, el GP se obtiene mediante el cálculo de la distribución condicional

$$f(x_*) | y, \{x_1, \dots, x_n\}; x_*$$

que se puede escribir como una distribución Gaussiana de media

$$\mathbb{E}[f(x_*)] = k_*^T (K + \sigma_n^2)^{-1} y \quad (14)$$

y varianza

$$\mathbb{V}[f(x_*)] = k_{**} - k_*^T (K + \sigma_n^2)^{-1} k_* \quad (15)$$

## 2) Clasificación

El proceso Gaussiano para problemas de clasificación está estrechamente relacionado con el proceso de Gauss para problemas de regresión. La gran diferencia entre GPR y GPC es cómo los datos de salida,  $y$ , están vinculados a las salidas de la función fundamental,  $\mathbf{f}$ . Esos datos ya no están conectados simplemente a través de un proceso de ruido como en (12), sino que ahora son discretos:  $y = +1$  para una clase, e  $y = -1$  para la otra. En principio, podríamos intentar ajustar un GP que produzca una salida de aproximadamente

1, para algunos valores de  $x$ , y aproximadamente -1 para otros, simulando esta discretización. En lugar de ello, interponemos el GP entre los datos y una función *squashing*.

Para clasificar un nuevo punto de datos  $x_*$  son necesarios dos pasos:

- a) Evaluar una ‘función latente’  $f$  que modele cualitativamente cómo la probabilidad de una clase frente a otra cambia a lo largo del eje  $x$ . Este es el proceso de Gauss.
- b) Hacer *squash* la salida de esta función latente sobre  $[0,1]$  usando una función sigmoidea como:

$$\pi(f) = \text{prob}(y = +1|f) \quad (16)$$

Una función sigmoidea es una función monótonamente creciente de  $\mathbb{R}$  a  $[0,1]$  que debe su nombre a que gráficamente posee la forma de la letra S.

Una vez introducido un nuevo dato en el clasificador, es necesario determinar su probabilidad de pertenencia a una u otra clase.

$$p(x_*|\mathbf{f}) = \mathcal{N}(K_*K^{-1}\hat{\mathbf{f}}, K_{**} - K_*(K')^{-1}K_*^T) \quad (17)$$

Nuestro objetivo es encontrar  $\hat{\mathbf{f}}$  y  $K'$  para conocerlo todo acerca de la producción del GP (17), el primer paso del clasificador. El siguiente paso no requiere entrenamiento, ya que es una función sigmoidea fija.

Por supuesto, nos gustaría comparar la utilidad cuantitativamente de los muchos GP que podrían estar asociados a nuestro conjunto de datos. Teniendo en cuenta las salidas  $\mathbf{f}$  de un determinado GP, la probabilidad que existe de ser apropiado para los datos de entrenamiento se pueden descomponer usando el teorema de Bayes:

$$p(\mathbf{f}|x, y) = \frac{p(y|\mathbf{f})p(\mathbf{f}|x)}{p(y|x)} \quad (18)$$

Ahora nos centraremos en los dos factores del numerador, ya que el denominador no depende de  $\mathbf{f}$ . Fijando nuestra atención en el primer factor y suponiendo que el conjunto de datos es independiente e idénticamente distribuido,

$$p(y|\mathbf{f}) = \prod_{i=1}^n p(y_i|f_i) \quad (19)$$

$p(y_i|f_i)$  es informada por nuestra función sigmoidea,  $\pi(f)$ . Concretamente,  $p(y = 1|f)$  es, por definición, igual a  $\pi(f)$ , así como  $p(y = -1|f) = 1 - \pi(f)$ .

El segundo factor del numerador  $p(\mathbf{f}|x)$  está relacionado con el primer bloque de nuestro esquema (figura esquema) pero primero nos interesa el valor de  $p(\mathbf{f}|x)$  que maximiza  $p(\mathbf{f}|x, y)$ . Esto ocurre cuando la derivada de (18) con respecto a  $\mathbf{f}$  es cero o cuando la derivada de su logaritmo es cero. Así obtenemos  $\hat{\mathbf{f}}$ .

$$\hat{\mathbf{f}} = K \nabla \log p(y|\hat{\mathbf{f}}), \quad (20)$$

donde  $\hat{\mathbf{f}}$  es la mejor  $f$  de nuestro problema. Desafortunadamente,  $\hat{\mathbf{f}}$  aparece a ambos lados de la ecuación por lo que haremos una estimación inicial. Entonces la ecuación (20) se puede utilizar en la (16). Así hemos encontrado una de las cantidades que necesitábamos.

La varianza de  $\mathbf{f}$  viene dada por la segunda derivada negativa del logaritmo de la ecuación (18), que resulta ser  $(K^{-1} + W)^{-1}$ , con  $W = -\nabla \nabla \log p(y|\mathbf{f})$ . Pretendemos tener una distribución gaussiana de  $p(\mathbf{f}|x, y)$  a través de una aproximación de Laplace:

$$p(\mathbf{f}|x, y) \sim q(\mathbf{f}|x, y) = N(\hat{\mathbf{f}}, (K^{-1} + W)^{-1}). \quad (21)$$

Ahora podemos confirmar que  $K' = K + W^{-1}$ , por lo que ya tenemos el GP completamente especificado.

### 3.4. Evaluación de las prestaciones de clasificadores

En este apartado se van a detallar las distintas medidas de evaluación que se utilizan típicamente en herramientas de ML para analizar los resultados [21], concretamente en problemas de clasificación binarios. Son medidas que ofrece la *toolbox* estudiada en este TFG, por lo que, además de describirlas teóricamente, se explica cómo PRoNTo facilita esta información. Algunas de estas medidas se utilizarán en capítulos posteriores para evaluar los experimentos realizados.

Las figuras que aparecen en cada apartado pertenecen a medidas obtenidas durante la ejecución de algunos de los experimentos realizados a lo largo del proyecto.

### 3.4.1. Tasa de acierto

La **precisión** (p) del modelo o tasa de acierto se presenta en una tabla de estadísticas junto con otros datos de interés como la precisión ponderada o la precisión de clases.

La **tasa de acierto** se calcula como el número total de muestras de prueba clasificadas correctamente dividido por el número total de muestras de ensayo (N), independientemente de la clase.

La **precisión ponderada** tiene en cuenta el número de muestras que hay en cada clase, y da la misma importancia a las precisiones obtenidas en las muestras de prueba de cada clase. Es la medida preferida cuando hay desequilibrio de clases (una clase, llamada la clase mayoritaria, tiene muchos más datos que las demás).

La tabla también da la **precisión de clases** (Class acc), útil para comprobar si el modelo favorece a unas clases sobre otras.

Todos los resultados son multiplicados por 100 para obtener los porcentajes correspondientes y facilitar así la comprensión visual rápida (ver Figura 3.7).

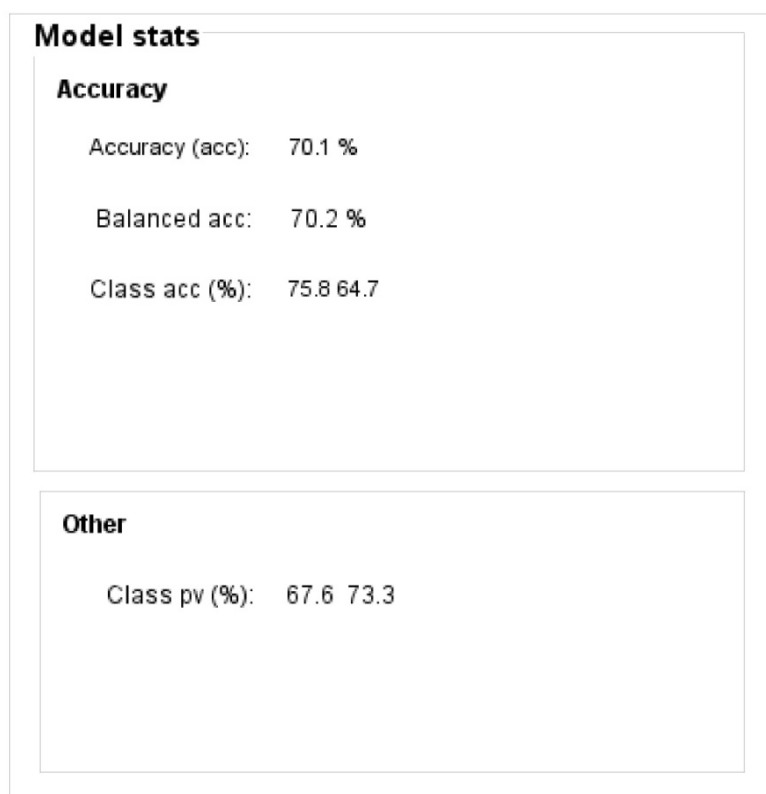


Fig. 3.7. Ejemplo de las prestaciones del modelo evaluadas por PRoNTTo.

### 3.4.2. Predicciones

Muestra el valor de salida de la función de decisión de la máquina para cada muestra de entrada. En el centro de la gráfica aparece una línea vertical que indica el umbral de decisión. Tendremos un clasificador de buen rendimiento si la mayoría de las muestras de una clase caen a un lado del umbral y las muestras de la otra clase al otro lado.

Gráficamente, cada clase se representa mediante un marcador diferente, y se indica en la leyenda. En la Fig. 3.8 se muestra el resultado de una clasificación para un problema binario. Como podemos observar, la mayoría de las muestras de la clase controles aparecen a la izquierda del umbral, así como la mayoría de las muestras de la clase pacientes aparecen a la derecha del mismo. Esto indica que la clasificación, principalmente, está siendo correcta ya que la máquina diferencia las muestras de una clase frente a las muestras de la otra clase.

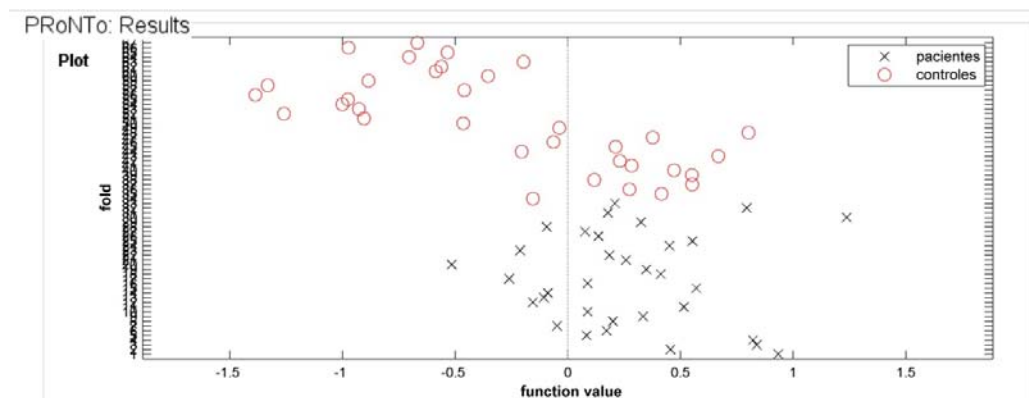


Fig. 3.8. Ejemplo gráfica con las salidas del clasificador para cada muestra de prueba en un caso binario.

### 3.4.3. Curva ROC

En la clasificación de dos clases, siempre hay un intercambio entre los errores de la clase +1 y de la clase -1. De hecho, la predicción de un clasificador para la clase 1, independientemente de la entrada, tendría una excelente precisión en la clase 1 pero bastante mala en la clase 2. Esto también se conoce como sensibilidad/ especificidad (Sensibilidad = 1- especificidad). Concretamente, podemos definir sensibilidad como la capacidad de nuestro clasificador para dar como casos positivos aquellos que tienen la condición buscada (verdaderos positivos). La especificidad, en cambio, se refiere a la capacidad que tiene el clasificador para detectar aquellos casos que no tienen la condición buscada (verdaderos negativos) [30].

La curva ROC es una representación gráfica de ese intercambio (ver Figura 3.9). Si el problema objeto de estudio es separable, podríamos diseñar un clasificador con una ROC ideal que pasase por la esquina superior izquierda.

El área bajo la curva (AUC) es una medida resumen del rendimiento del clasificador. Cuanto mayor es, mejor es el clasificador (1 representa el rendimiento perfecto, 0.5 el rendimiento aleatorio). Igual que con todas las medidas resumen, el AUC no es sino una manera de comparar el rendimiento de las máquinas, y no puede ser utilizada sola para declarar una máquina estadísticamente significativamente superior a otra sobre un determinado conjunto de datos.

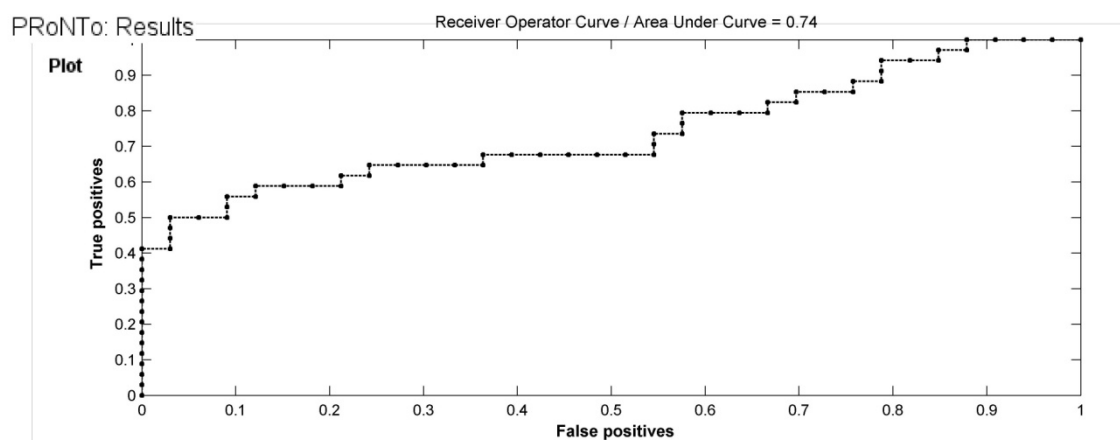


Fig. 3.9. Ejemplo de curva ROC cuya AUC es igual a 0.74.

### 3.4.4. Histograma de la salida del clasificador

El histograma de la salida del clasificador es una versión suavizada de la densidad del apartado de predicción (ver 3.4.1.), que muestra cómo se distribuyen los valores de la función de clasificación sobre cada clase (ver Figura 3.10). Un buen clasificador tiene solapamiento mínimo o nulo entre las densidades de cada clase. La tasa de error del clasificador es proporcional al área del solapamiento.

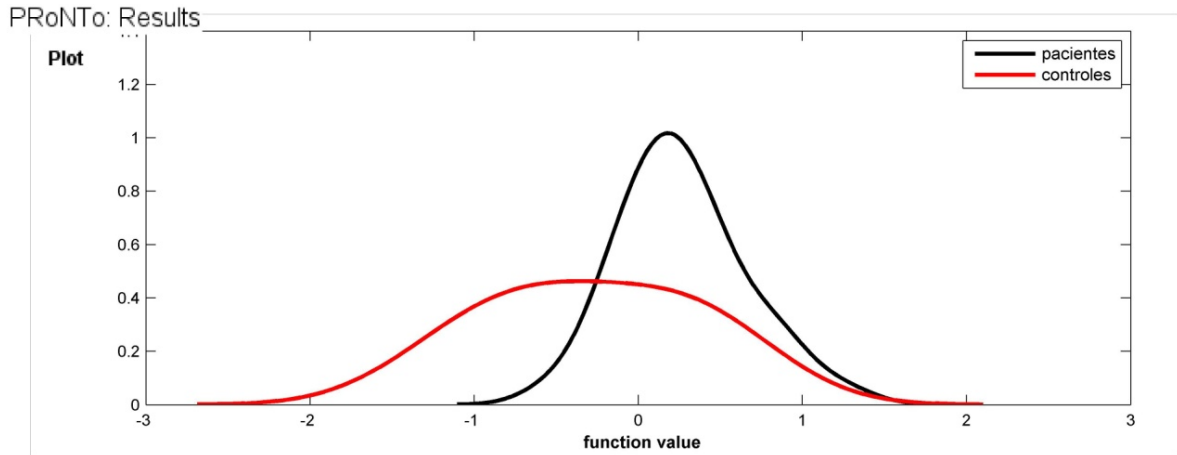


Fig. 3.10. Ejemplo de histograma de salida del clasificador de un problema binario.

### 3.4.5. Matriz de confusión

La matriz de confusión muestra el recuento de etiquetas de clases predichas (filas) frente a etiquetas de clases reales (columnas). Es de gran utilidad ya que puedes observar gráficamente la calidad de un clasificador (ver Figura 3.11). Una matriz de confusión ideal es diagonal, donde todas las etiquetas de las clases predichas se corresponden con las reales. Los elementos fuera de la diagonal se corresponden con errores.

Gracias a la matriz de confusión también podemos ver si las clases resultan confundibles, es decir, si las muestras se están etiquetando de forma incorrecta. Es importante comprobar que ninguna de las clases se sacrifica para ganar precisión en otras clases, en otras palabras, si todas las clases son igualmente importantes para la clasificación, ninguna clase debería tener más puntos fuera de la diagonal que dentro.

Muchas estadísticas de resumen, incluyendo la precisión de clases, la exactitud total, la sensibilidad y especificidad, se pueden calcular a partir de la matriz de confusión.



PRoNTTo: Results

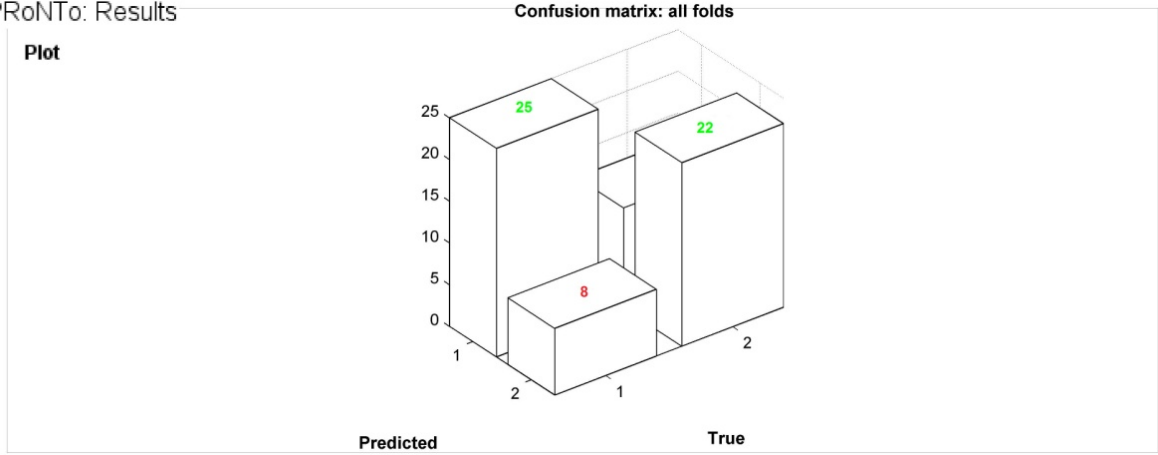


Fig. 3.11. Ejemplo de matriz de confusión para una clasificación binaria.

## Capítulo 4

# Diseño de la solución técnica

---

En este capítulo se va a presentar la herramienta utilizada en el proyecto, PRoNTTo, describiendo su objetivo principal así como sus distintas funcionalidades.

Además, se explicarán detalladamente los experimentos realizados y los pasos que se han seguido en PRoNTTo para su obtención.

### 4.1. Descripción inicial de los datos

Para la realización de este TFG hemos contado con una serie de datos de tipo sMRI. Estos datos, que en total suman 645, pertenecen tanto a enfermos de TOC como a controles sanos, obtenidos en hasta 6 centros distintos. En la siguiente Tabla se ofrece un desglose de los datos de cada uno de los centros.

	Datos de controles	Datos de pacientes	Datos totales
Centro 1	37	36	73
Centro 2	102	86	188
Centro 3	33	44	77
Centro 4	33	34	67
Centro 5	91	53	91
Centro 6	73	76	149
			<b>645 datos</b>

Tabla 4.1. Conjunto de datos de tipo sMRI de los que se disponen.

### 4.2. PRoNTTo v.1.1

PRoNTTo (*Pattern Recognition Neuroimaging Toolbox*) es una *toolbox* sobre Matlab que implementa diferentes técnicas de reconocimiento de patrones orientados al análisis de datos de neuroimagen, y desarrollada por la *University College* de Londres [21]. El reconocimiento estadístico de patrones es un campo en el área del aprendizaje máquina que se ocupa de la detección automática de patrones en los datos mediante el uso de algoritmos y de procesamiento de datos.

En PRoNTo, los escáneres cerebrales se tratan como patrones espaciales, las etiquetas corresponden, en los modelos de clasificación a diferentes estados mentales (por ejemplo, mirando casas vs. no mirando casas) o tipos de sujetos (por ejemplo, los pacientes de una determinada enfermedad frente a los controles sanos) o en el caso de los modelos de regresión, cualquier medida continua relacionada con las exploraciones del cerebro (por ejemplo, edad, o el grado de la enfermedad, medida por una escala clínica). Se utilizan modelos de aprendizaje estadístico para identificar las propiedades estadísticas de los datos que, a su vez, se pueden utilizar para discriminar entre las condiciones o grupos de sujetos experimentales (modelos de clasificación) o para predecir una medida continua (modelos de regresión).

#### 4.2.1. Fases principales de PRoNTo

La *toolbox*, basada en Matlab, incluye cinco pasos principales a seguir para solucionar un problema de regresión o, como en nuestro caso, de clasificación (véase Figura 4.1.).

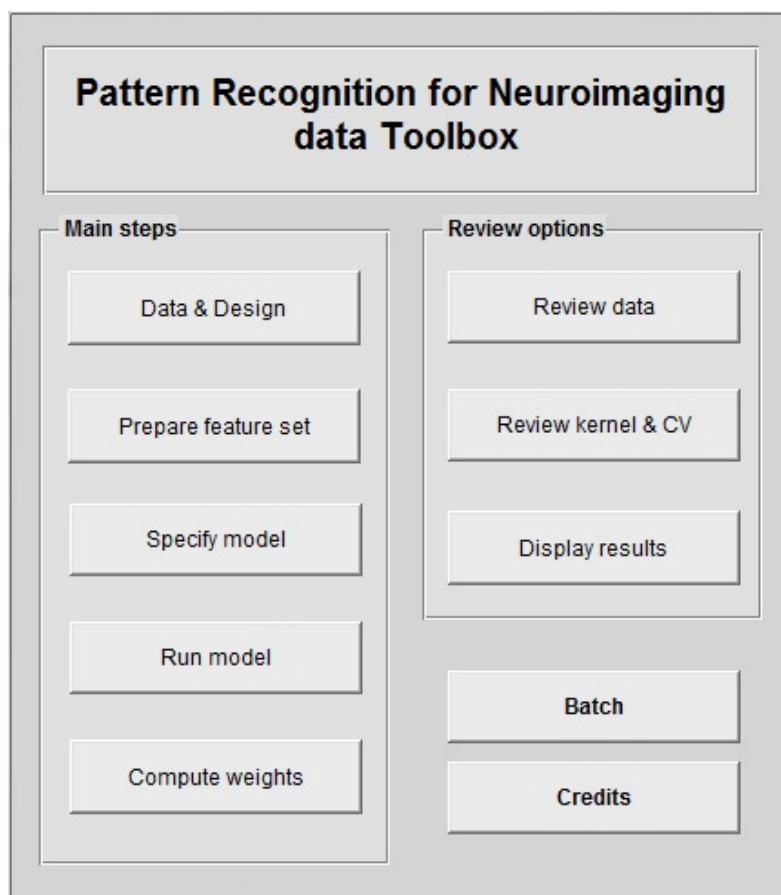


Fig. 4.1. Ventana principal de la toolbox PRoNTo.

La explicación de cada fase se desarrolla a continuación:

### 1) *Data & Design*

En esta fase se introducen todos los datos necesarios para comenzar el análisis estadístico. El usuario debe introducir los archivos de tipo imagen. PRoNTto soporta conjuntos de datos de varias modalidades, es decir, permite al usuario introducir datos de más de una modalidad de datos, como fMRI (*functional Magnetic Resonance Imaging*), sMRI (*structural Magnetic Resonance Imaging*) y PET (*Positron Emission Tomography*). Además, la herramienta permite hacer grupos de sujetos que pueden resultar útiles en algunas ocasiones, como veremos posteriormente. En esta fase es necesario introducir una máscara de selección de vóxeles para cada modalidad. Esta es una máscara de primer nivel que se utiliza simplemente para optimizar el conjunto de características descartando aquellas que son poco interesantes, tales como vóxeles fuera del cerebro.

Al terminar la introducción de todos los datos necesarios y ejecutar esta fase se genera un archivo PRT.mat que contiene todos los datos necesarios para continuar con la resolución del problema de clasificación.

### 2) *Prepare feature set*

Es en esta fase la herramienta accede a cada imagen, es decir, consigue el valor de los vóxeles comprendidos en la primera máscara de selección incluida en la fase anterior. A partir del conjunto de características, se crea una matriz de dimensiones  $N$  muestras por  $N$  características. Esta matriz será la que se le pase al modelo estadístico como entrada.

También, existe la posibilidad de incluir una máscara de selección de vóxeles de segundo nivel. El interés de esto, por ejemplo, es definir regiones de interés (ROIs) que contengan la información necesaria para la clasificación.

### 3) *Specify model*

En esta fase es donde se selecciona el conjunto de características, es decir, los datos que se utilizarán como entrada, el tipo de problema que se pretende resolver, si es de clasificación o de regresión, que algoritmo de ML se va a

emplear, la estrategia de validación cruzada a manejar y las operaciones que se aplicarán antes de entrenar al algoritmo.

#### 4) *Run model*

Tras especificar el modelo en la fase anterior es el momento de aprenderlo, es decir, en esta fase se utilizan los datos de entrenamiento y de test para estimar los parámetros de los clasificadores y comprobar cómo es de bueno el modelo haciendo predicciones.

#### 5) *Compute weights*

Para los modelos lineales, Pronto proporciona la opción de recuperar los pesos del modelo en el espacio de características originales (vóxel), y transformar el vector de pesos en una imagen, o un mapa (véase Figura 4.2). Estos mapas contienen en cada vóxel el peso correspondiente del modelo lineal, es decir, se refiere a lo mucho que este vóxel particular, contribuyó a la tarea de clasificación / regresión en cuestión. Tras elegir el archivo del mapa de pesos en la visualización de los resultados, el mapa de ponderación se visualizará en forma de cruz y con una barra de colores. La barra de colores indica la importancia relativa del vóxel en la función de decisión de la máquina.

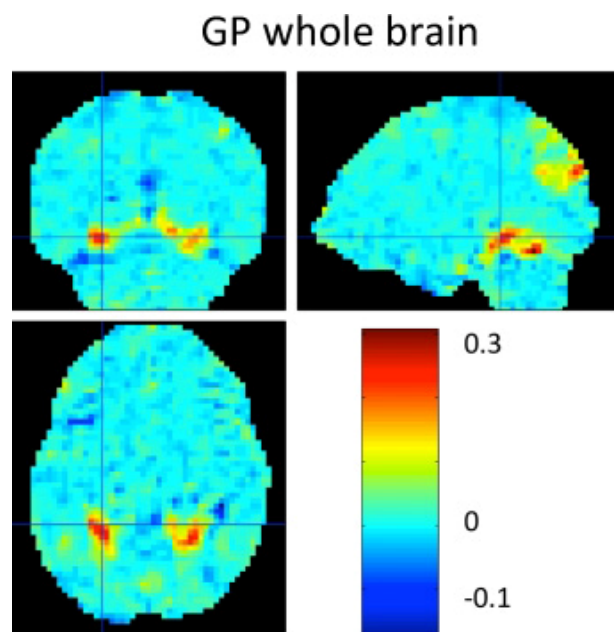


Fig. 4.2. Ejemplo de imagen de mapa de pesos

## 4.2.2. Fases de verificación

Además, PRoNTo incluye tres fases donde examinar tanto los datos introducidos al crear una máquina, como el procedimiento de validación cruzada seleccionado, o estadísticas de los resultados de la clasificación.

### 1) *Review data*

Sirve para comprobar la estructura creada en la primera fase principal. Por ejemplo, puedes ver el número de grupos de sujetos o modalidades creadas, o el número de sujetos introducidos (véase Figura 4.3).

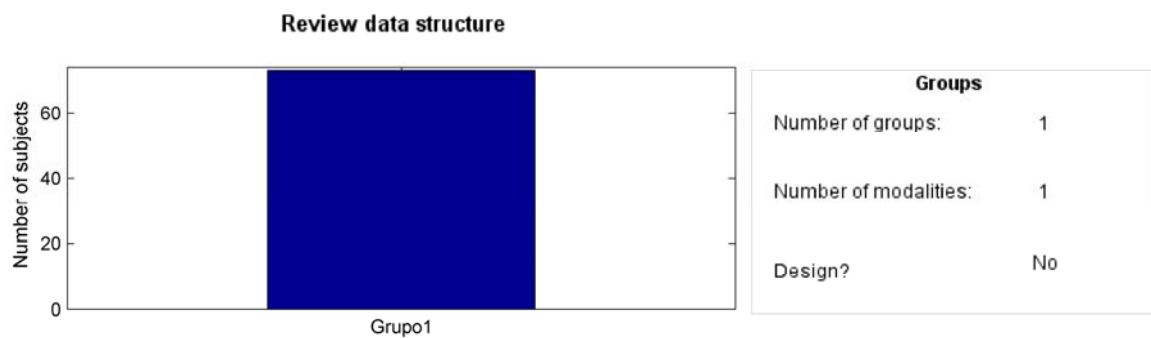


Fig. 4.3. Ejemplo de estructura creada en la primera fase principal de la herramienta.

### 2) *Review Kernel & CV*

Es el lugar donde revisar el modelo creado, la validación cruzada y el *kernel*. Al seleccionar esta opción, y tras elegir el archivo PRT.mat correspondiente, aparece una primera ventana donde seleccionar el modelo que desees analizar y tres botones para visualizar lo que sea necesario, como la definición de clases realizada, el procedimiento de validación cruzada utilizado (véase Figura 4.4) o el *kernel*.

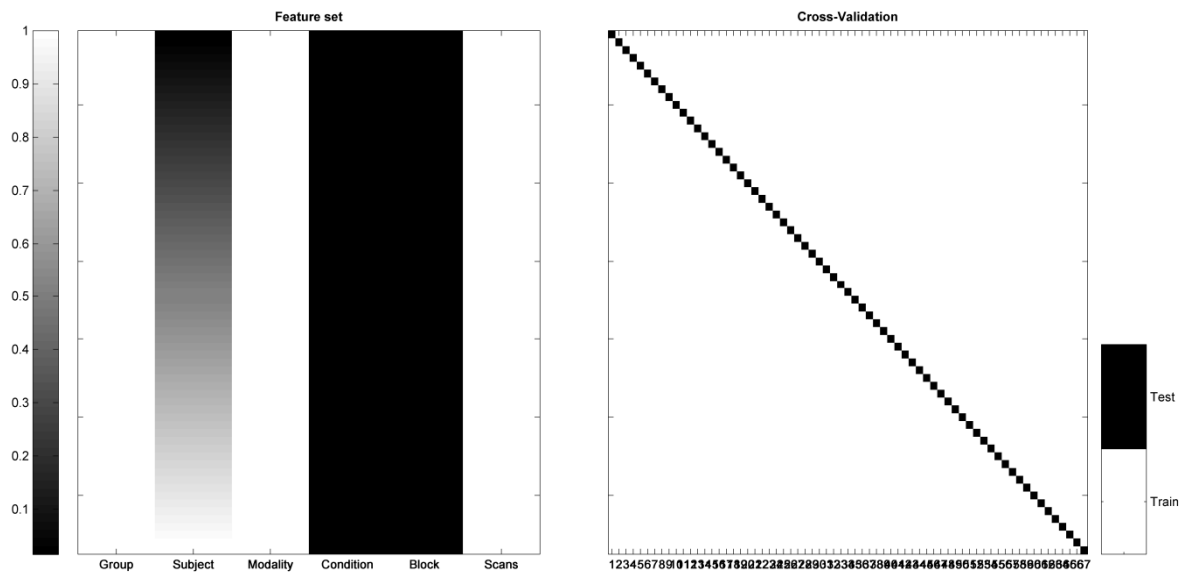


Figura 4.4. Ejemplo de modelo creado con validación cruzada tipo LOO.

### 3) *Display Results*

Muy útil para analizar las medidas de evaluación según los experimentos realizados (véase Figura 4.5), así como comprobar la tasa de acierto (ver 3.4.1). Además, puedes incluir el mapa de pesos, si lo has creado con anterioridad, y añadir una imagen anatómica cerebral en el caso de disponer de ella. La imagen anatómica se relaciona con el mapa de pesos, de esta manera se puede ver alrededor de que áreas del cerebro se encuentran los vóxeles con mayor peso en la clasificación.

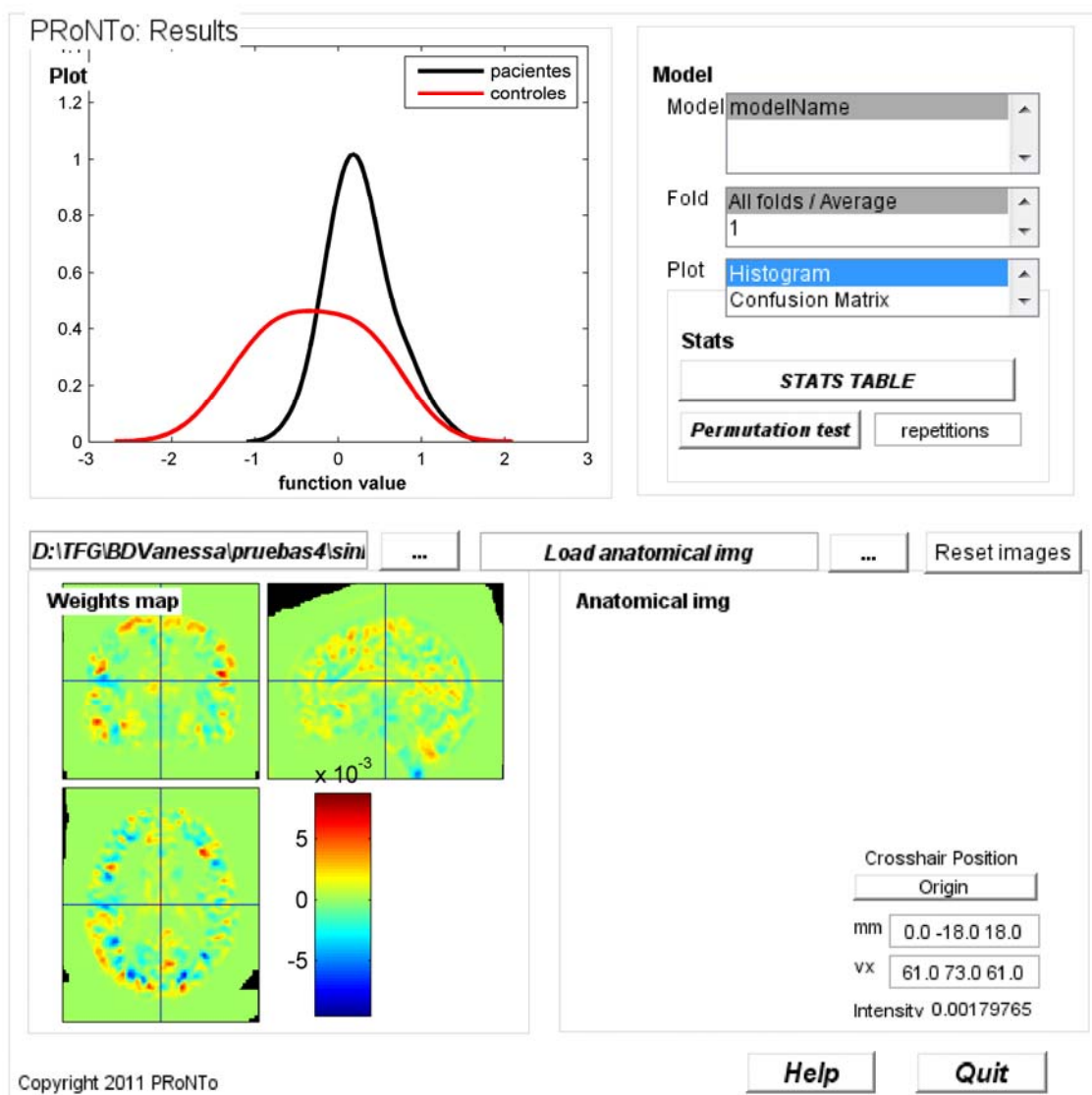


Fig. 4.5. Ejemplo de ventana del módulo "Display Results".

### 4.2.3 Batch

La pestaña *Batch*, que encontramos en la ventana principal de PRoNTTo, permite hacer lo mismo que con la interfaz descrita hasta ahora salvo que, en algunas ocasiones, se puede personalizar más el experimento. Debes introducir cada dato paso a paso. Lo más importante es que la salida siempre es la misma, es decir, no varían los resultados dependiendo de si utilizas la interfaz gráfica de usuario o el sistema *Batch*.

Esta opción resulta menos intuitiva que la interfaz gráfica de usuario pero permite guardar el trabajo como un archivo .mat, y cargarlo después para continuar el trabajo, o como un archivo .m, permitiendo así abrirlo con el editor de Matlab. Esto la



convierte en una buena herramienta que puede hacer que la especificación de los datos y el diseño sean mucho más fáciles y rápidos.

## 4.3. Experimentos

En los próximos apartados se tratará de explicar los experimentos realizados con la herramienta de ML PRoNTo para entender la finalidad de los mismos y así, en el capítulo posterior, facilitar la comprensión de los resultados obtenidos y como se ha llegado a ellos.

### 4.3.1. Análisis de los modelos estadísticos

Algunos métodos de aprendizaje máquina son mejores que otros dependiendo del tipo de problema. Por este motivo, consideramos necesario hacer un estudio para descubrir que algoritmo de clasificación de los que ofrece PRoNTo es el más adecuado para detectar TOC con los datos disponibles.

Los pasos a seguir para realizar dicho experimento son los que siguen a continuación.

Tras abrir la herramienta (basta con escribir “pronto” o “prt” en la ventana de comandos de Matlab) y encontrarnos con la pantalla principal de PRoNTo, debemos seleccionar la primera fase principal. La herramienta es bastante intuitiva, como podemos observar en la Figura 4.6. Lo primero que debemos hacer es seleccionar un directorio, donde vamos a guardar el archivo de salida PRT.mat. Este archivo contiene campos con toda la información y datos necesarios para los pasos posteriores del análisis y se crea al completar la primera fase principal de la *toolbox*. En las siguientes fases de PRoNTo este archivo se actualizará con los datos que se vayan introduciendo.

A continuación, se crean los grupos de sujetos que se consideren oportunos, que en nuestro caso bastará con uno. Estos grupos pueden resultar útiles en el caso de querer aplicar una modalidad a una parte del total de los datos, y otra modalidad distinta a la parte restante. En ese caso, habría que hacer dos grupos con los datos que correspondan. Se debe elegir también si la introducción del resto de los datos se hará por sujetos, adecuada en el caso de disponer de datos fMRI al tener unos pocos sujetos con muchas imágenes por cada uno, o por escáneres, adecuada para datos sMRI, como los nuestros, donde tenemos muchos sujetos con una imagen por cada uno. Además, es obligatorio seleccionar una primera máscara, asociada a la modalidad creada, que descarte características irrelevantes, como vóxeles fuera del cerebro.

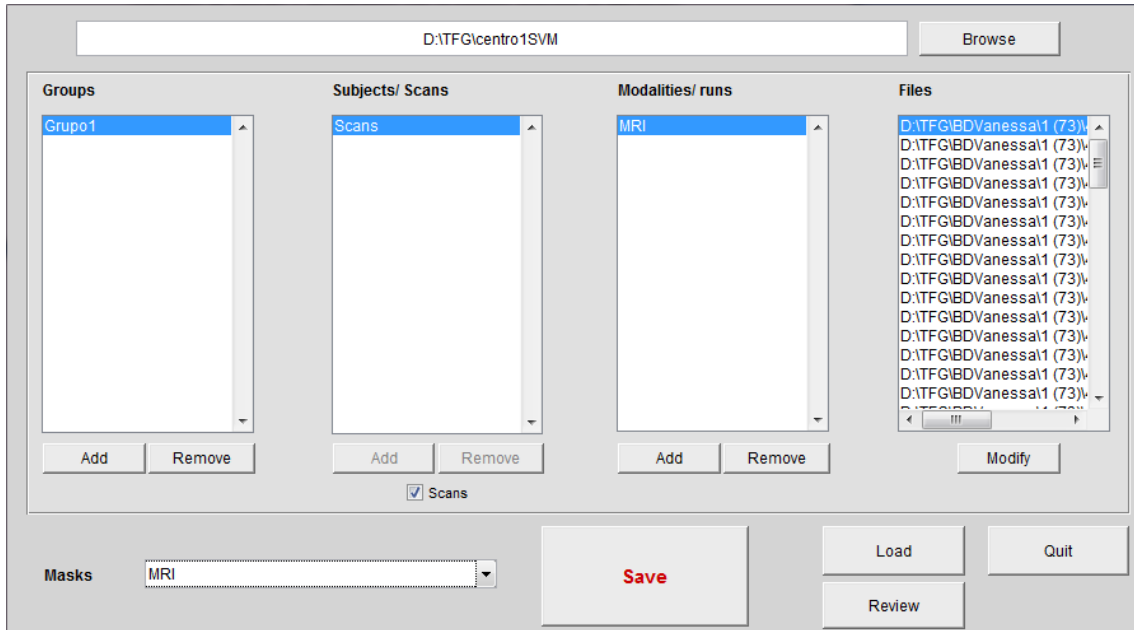


Fig. 4.6. Ventana de la fase *Data & Design* de la toolbox PRoNTTo.

Antes de pasar a la siguiente fase debemos guardar los datos (pulsando en “Save”) y salir (pulsando en “Quit”), lo que generará el archivo PRT.mat.

Una vez que se han introducido los datos necesarios para comenzar el análisis estadístico podemos pasar a la segunda fase principal. En la ventana que aparece (ver Figura 4.7), lo primero que debemos hacer es seleccionar el archivo PRT.mat creado en el paso anterior.

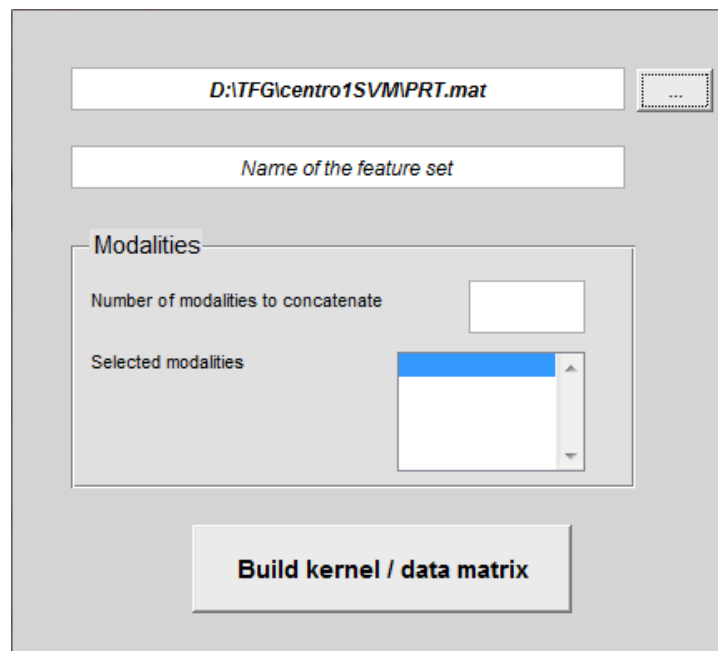
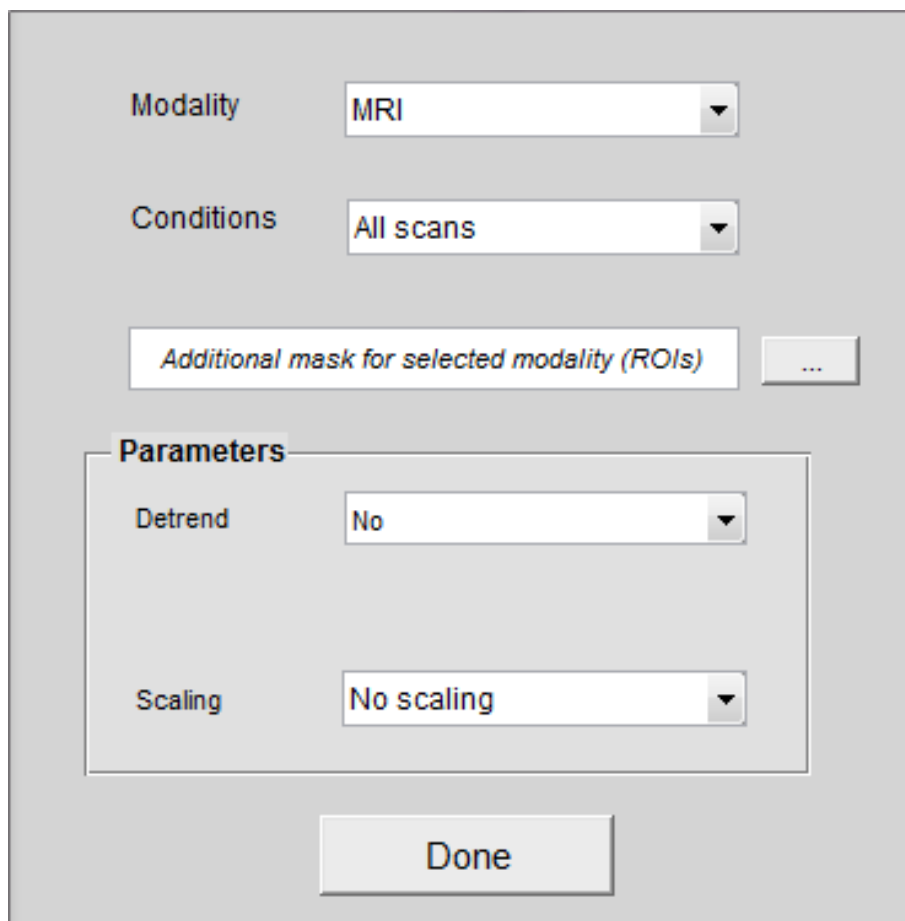


Fig. 4.7. Ventana principal de la segunda fase principal de PRoNTTo.

Seguidamente aparecerá una segunda ventana, como la de la Figura 4.8, en la que debemos seleccionar la modalidad creada en la primera fase. Esto es posible ya que, para otros experimentos, puede resultar útil definir varias modalidades. En esta ventana encontramos la opción de añadir una segunda máscara de selección de vóxeles que utilizaremos en experimentos posteriores pero no por ahora. Los parámetros que aparecen al final de la ventana no son necesarios en nuestro caso ya que el primero de ellos es para datos fMRI y el segundo es útil en el caso de tener datos PET. Podemos finalizar en esta ventana (pulsando “Done”) y solo nos faltaría volver a la primera ventana de esta fase y poner un nombre al conjunto de características.



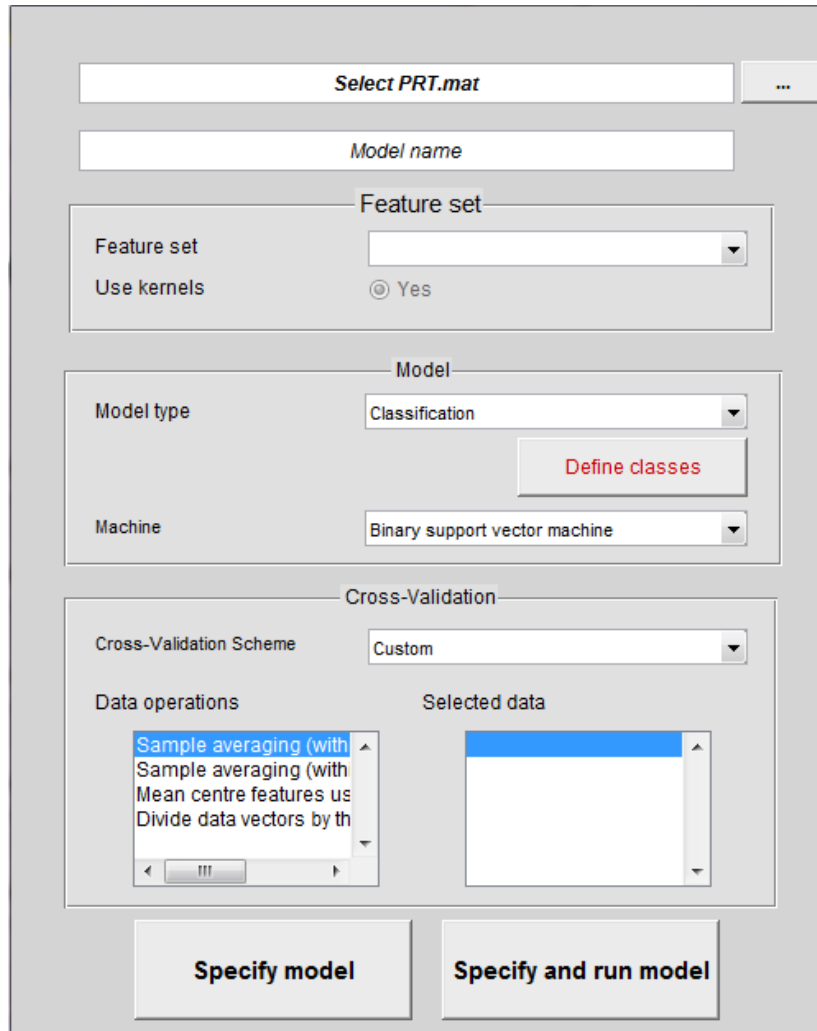
The screenshot shows a software interface with the following elements:

- Modality:** A dropdown menu currently showing 'MRI'.
- Conditions:** A dropdown menu currently showing 'All scans'.
- Additional mask for selected modality (ROIs):** A text input field followed by a button with three dots (a browse button).
- Parameters:** A sub-section containing:
  - Detrend:** A dropdown menu currently showing 'No'.
  - Scaling:** A dropdown menu currently showing 'No scaling'.
- Done:** A large button at the bottom center of the window.

Fig. 4.8. Segunda ventana de la segunda fase principal de PRoNTTo.

Volviendo a la pantalla principal de PRoNTTo continuamos el experimento seleccionando la tercera fase, *Specify model*. Aparecerá una ventana similar a la de la Figura 4.9. Como podemos ver, lo primero que debemos hacer, como en el paso anterior, es elegir el archivo PRT.mat y darle un nombre al modelo. Después, seleccionaremos el conjunto de características creado en la fase anterior. En esta

ventana se elige el modelo de clasificación, el procedimiento de validación cruzada que se va seguir y los preprocesados de datos que se quieran incluir.



The screenshot shows a software interface for configuring a machine learning model. It is divided into several sections:

- Select PRT.mat**: A text input field with a browse button (...).
- Model name**: A text input field.
- Feature set**: A dropdown menu for selecting the feature set, and a radio button labeled "Use kernels" with "Yes" selected.
- Model**: A dropdown menu for "Model type" set to "Classification", a "Define classes" button, and a dropdown menu for "Machine" set to "Binary support vector machine".
- Cross-Validation**: A dropdown menu for "Cross-Validation Scheme" set to "Custom", and two lists for "Data operations" and "Selected data".

At the bottom of the window are two buttons: "Specify model" and "Specify and run model".

Fig. 4.9. Ventana de la tercera fase principal de la herramienta.

Tras seleccionar el archivo PRT.mat, asignar un nombre al modelo y seleccionar el conjunto de características, el siguiente paso será seleccionar si se trata de un problema de clasificación o de regresión. Como en nuestro caso se trata de un problema de clasificación, a continuación tendremos que definir las clases. Aparecerá una nueva ventana (ver Figura 4.10) en la que hay que indicar el número de clases y, una vez indicado esto, podrás ir clase por clase asignando un nombre y los sujetos que incluye, uno por uno. En nuestro experimento debemos tener dos clases, pacientes, con los sujetos sanos, y controles, con los sujetos que padecen TOC.

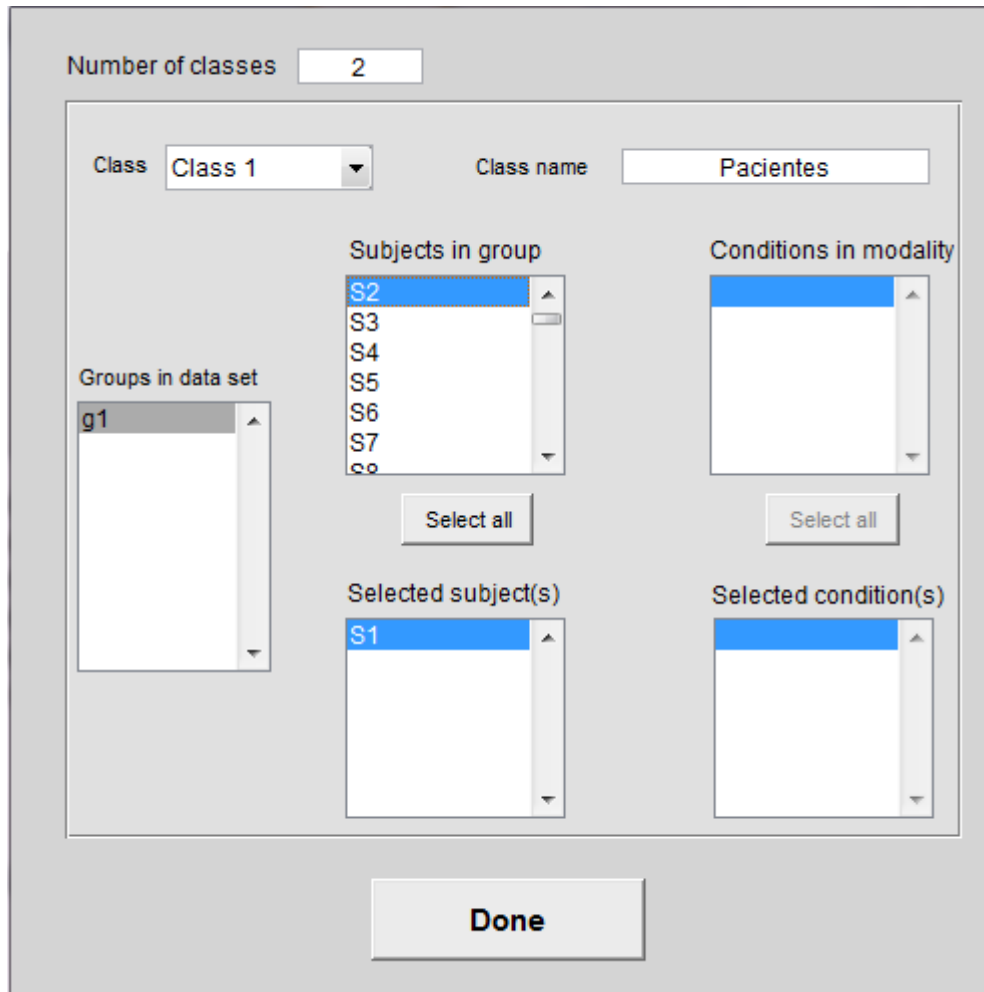


Fig. 4.10. Ventana donde se definen las clases del modelo.

Una vez definidas las clases, debemos continuar en la ventana de la Figura 4.9 y seleccionar la máquina objeto de estudio. Como queremos comprobar con cuál de ellas obtenemos la mejor tasa de acierto habrá que realizar el experimento seleccionando cada vez una de ellas. No sería necesario realizar los pasos de las fases anteriores cada experimento sino que bastaría con hacerlos para el primero y en los siguientes empezar por la fase tres, *Specify model*, variando la máquina.

En esta fase, también se encuentra el apartado de validación cruzada (véase Figura 4.11). Vamos a utilizar el procedimiento de validación cruzada LOO (ver 3.2.2) pero con algunas particularidades. PRoNTo ofrece varios tipos apropiados según el diseño experimental.

- LOSO (*Leave One Subject Out*) es adecuado para los diseños de varios sujetos. En cada iteración se utiliza un sujeto para testear y el resto para entrenar.

- LORO (*Leave One Run Out*) este tipo es adecuado para diseños de un único sujeto, adaptándose mejor a los diseños que tienen múltiples ensayos experimentales. En cada iteración deja fuera una ejecución (modalidad) de cada sujeto.
- LOBO (*Leave One Block Out*) también es adecuado para diseños de un único sujeto, pero es apropiado cuando sólo hay una única prueba. En un diseño por bloques, cada bloque tendrá una duración de un cierto número de exploraciones de fMRI. Con este tipo de validación en cada iteración se deja fuera uno de esos bloques, y se entrena con el resto.

La versión actual de PRoNTTo soporta cada uno de ellos. En el caso de este proyecto, al utilizar datos sMRI, el más adecuado es el método LOSO.

Además, la herramienta ofrece la posibilidad de seleccionar en esta tercera fase uno o más preprocesados para aplicar a los datos que serán descritos en el próximo apartado. Estas operaciones permiten hacer selección de vóxeles lo que, si se aplica correctamente, ayuda a obtener mejores resultados. En estos primeros experimentos, donde el objetivo es analizar las máquinas de clasificación, no vamos a utilizar ninguna de estas operaciones.

Ahora tan solo nos falta comprobar los resultados obtenidos en la fase *Display Results* de la ventana principal de PRoNTTo, en el que podemos ver todas las medidas de evaluación habituales en ML.

### 4.3.2. Análisis de métodos de preprocesado

En este apartado vamos a explicar los diferentes tipos de preprocesado que ofrece la toolbox para datos sMRI. PRoNTTo da la posibilidad de aplicar hasta cuatro operaciones a los datos (véase Figura 4.11). Sin embargo, las dos primeras solo son de utilidad en el caso de tener datos de tipo fMRI, lo que significa que para nosotros son solo útiles las dos últimas operaciones.

Los pasos a seguir en las dos primeras fases principales de PRoNTTo son los mismos que los explicados en el apartado anterior. Tras añadir todos los datos necesarios en la primera fase y crear la matriz de características, que se le pasará al clasificador, en la segunda fase podemos pasar a la tercera, donde esta vez analizaremos qué ocurre al seleccionar preprocesados de datos.

Las dos operaciones que ofrece PRoNTTo útiles en el caso de tener datos sMRI son las que se describen a continuación:

- *MEAN CENTRE FEATURES USING TRAINING DATA:*

El algoritmo *Mean Centre* identifica el centro de concentración (o el centro geográfico) de un conjunto de características, en este caso del conjunto de datos de entrenamiento [12].

Es útil para el seguimiento de los cambios en la distribución o para la comparación de las distribuciones de los diferentes tipos de características.

Analíticamente [14], el algoritmo *Mean centre* es el valor medio de las coordenadas X, Y, Z de todas las características de área de estudio, y viene dado por:

$$\bar{X} = \frac{\sum_{i=1}^n X_i}{n}, \bar{Y} = \frac{\sum_{i=1}^n Y_i}{n}, \bar{Z} = \frac{\sum_{i=1}^n Z_i}{n}$$

donde  $X_i$ ,  $Y_i$  y  $Z_i$  son las coordenadas en tres dimensiones de todas las características, y n es el número total de características.

Esta operación lo que hace es restar la media, obtenida con el método *Mean Centre*, vóxel a vóxel.

- *DIVIDE DATA VECTORS BY THEIR NORM:*

Escala cada vector de datos apoyado sobre la hipersfera de radio unidad dividiéndolo por su norma euclídea.

En un espacio euclídeo los vectores se pueden representar como segmentos orientados entre puntos de dicho espacio. Dado un vector de un espacio vectorial euclídeo, la norma de un vector se define como la distancia euclídea (en línea recta) entre dos puntos A y B que delimitan dicho vector [13].

Para un espacio euclídeo de 3 dimensiones:

$$\|\overrightarrow{AB}\| = \sqrt{(b_1 - a_1)^2 + (b_2 - a_2)^2 + (b_3 - a_3)^2}$$

siendo  $\overrightarrow{OA} = (a_1, a_2, a_3)$ ,  $\overrightarrow{OB} = (b_1, b_2, b_3)$  y O el origen de coordenadas de dicho espacio.

Cabe destacar el hecho de que en un espacio euclídeo la norma de un vector coincide con el módulo de dicho vector. Esto implica que la operación realizada

normaliza el vector en cuestión, obteniendo como resultado un vector unitario, de la misma dirección y sentido que el vector original.

Estos dos preprocesados pueden aplicarse tanto por separado como en conjunto. Es necesario tener en cuenta que el orden en el que son seleccionados es de gran importancia ya que será el mismo orden en el que se apliquen. Basta con pulsar sobre la operación u operaciones que desees y automáticamente aparecerán como seleccionadas, como se puede observar en la Figura 4.11 donde se ha seleccionado la operación *Mean centre features using training data*.

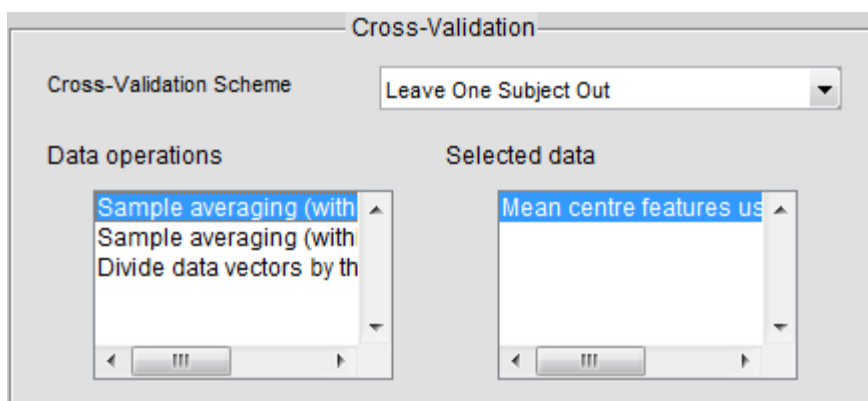


Fig. 4.11. Sección de validación cruzada y preprocesados de datos en PRoNT.

El experimento continuaría de la misma manera que en el caso del apartado anterior, pudiendo revisar los resultados obtenidos seleccionando la fase correspondiente en la ventana principal de la herramienta.

### 4.3.3. Análisis de máscaras de selección

En este apartado, vamos a hablar acerca de las ventajas de utilizar una segunda máscara de selección de vóxeles ya que, como se mencionó con anterioridad, PRoNT exige incluir una primera máscara y da la opción de incluir una segunda.

Incluir esta máscara adicional puede ser de utilidad si queremos definir regiones de interés en las que realizar la clasificación. Los vóxeles que no se incluyan en la máscara quedarán descartados.

Para ello, debemos situarnos en la segunda fase principal de la herramienta. Recordemos que en ella, tras seleccionar el archivo PRT.mat correspondiente, aparece una segunda ventana. Es ahí donde podemos incluir la segunda máscara. Tan solo hay que seleccionar el archivo que queramos. Este debe ser un archivo con extensión .img



pero también debemos tener uno con extensión .hdr en la misma carpeta. El archivo .img contiene los número que componen la información de la imagen. El archivo .hdr contiene información acerca del archivo .img, como el volumen representado por cada número de la imagen (tamaño de vóxel) o el número de píxeles en las coordenadas X, Y y Z [32].

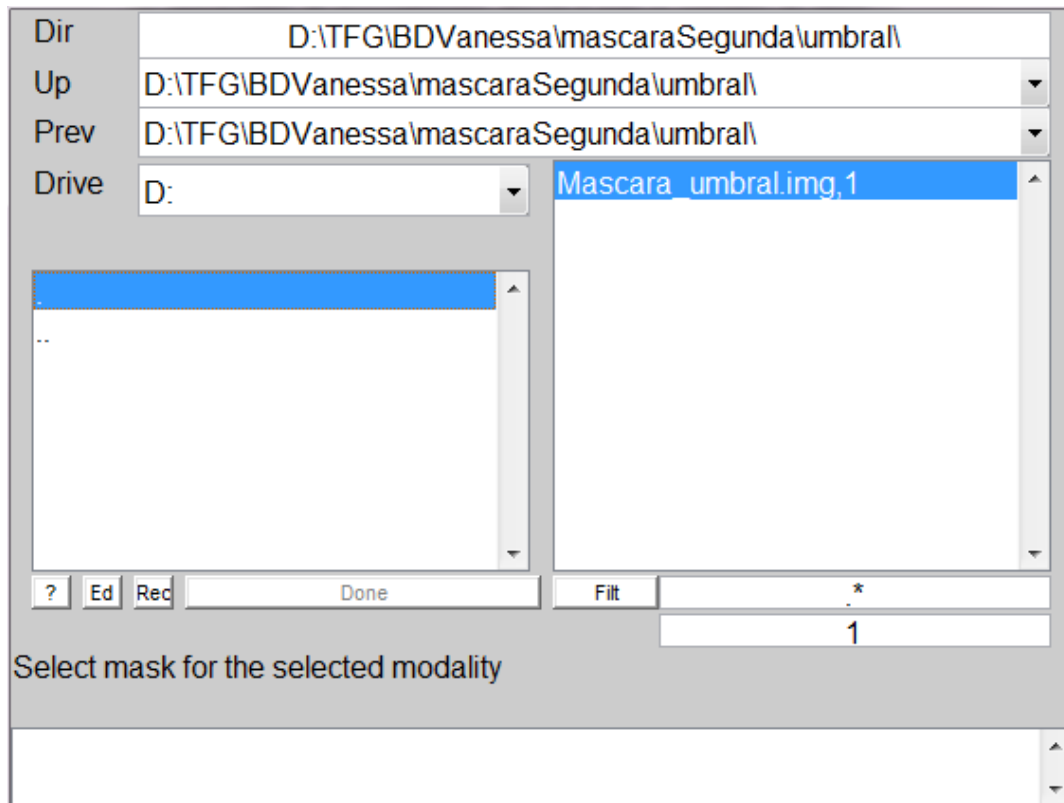


Fig. 4.12. Búsqueda del archivo correspondiente a la segunda máscara.

Como vemos en la Figura 4.12, tras localizar el archivo solo tenemos que seleccionarlo y aparecerá en el recuadro de abajo, como máscara seleccionada.

#### 4.3.4. Estudio multicentro

Otro de los objetivos de este proyecto es comprobar qué ocurre si se realiza el experimento utilizando los datos de todos los centros de los que disponemos en conjunto. Para ello, además de introducir todos los sujetos en la primera fase, se deben crear las clases, pacientes y controles, cuidadosamente en la tercera fase. Es necesario incluir todos los sujetos de control de todos los centros en una clase, y los pacientes en otra.

Tras este estudio, se ha querido experimentar utilizando los datos de un centro como datos de entrenamiento y los de otro como datos de test. Además de introducir los

sujetos de ambos centros al comenzar el experimento, el gran cambio, respecto a los experimentos simples anteriores, se encuentra en el momento de realizar la validación. En este caso es necesario utilizar la opción *custom*. Se debe crear un archivo *.mat* en el que se indiquen los sujetos de entrenamiento y los sujetos de test. Este archivo debe contener una matriz con los siguientes valores: 0, para los sujetos que no quieras incluir; 1, para los sujetos de entrenamiento; y 2, para los sujetos de test [21]. Aprovecharemos para crear el archivo con dos vectores, uno en el que los datos de entrenamiento se correspondan con el primer centro y los de test con el segundo, y viceversa. Los dos vectores deben concatenarse, es decir, debe formarse una matriz de dimensión igual al número total de datos por dos.

Para que el archivo sea correcto debe cumplir una serie de requisitos impuestos por el código de la herramienta. Concretamente, si nos fijamos en el archivo *prt\_model.m* encontramos una serie de condiciones para nuestro archivo *custom*:

- Es necesario que la matriz creada reciba el nombre de CV, ya que si no es así el código de la herramienta no la reconocerá.
- Los vectores creados deben ser vectores columna.
- Debe haber sujetos de test en todos los vectores.
- Debe haber sujetos de entrenamiento en todos los vectores.

Una vez creado el archivo bastará con seleccionarlo en el momento de definir el procedimiento de validación.

A la hora de visualizar las medidas de evaluación en la fase de PRoNTto que corresponde, en el apartado “Fold” podremos ver los resultados propios a un vector u otro seleccionando 1 o 2. Es decir, podremos ver los resultados de utilizar el primer centro como entrenamiento y el segundo como test si seleccionamos “Fold” igual a 1 y viceversa si seleccionamos “Fold” igual a 2.

También se han realizado algunos experimentos con más de 2 centros, utilizando los datos de un centro como entrenamiento y los del resto como test. El mecanismo a seguir es el mismo salvo que la matriz tendrá una dimensión igual al número total de datos por el número total de centros.

## Capítulo 5

# Resultados finales

---

En los próximos apartados se muestran los resultados obtenidos a través de diferentes experimentos realizados, cubriendo los siguientes objetivos:

- Encontrar el modelo de clasificación más adecuado para nuestro problema
- Los preprocesados de datos convenientes que nos ayuden a mejorar las tasas de acierto de la clasificación
- Las máscaras de selección de vóxeles necesarias para seleccionar el subconjunto de vóxeles que nos aporte más información

Todos ellos para conseguir una clasificación fiable de datos MRI de cualquier centro que nos asegure si se trata o no de un enfermo de TOC.

También se muestran los resultados obtenidos al realizar experimentos combinando datos de todos los centros disponibles para asegurar que con cualquier nuevo dato la clasificación de la máquina sea correcta.

Al final de cada apartado de pruebas se ofrecen algunas conclusiones del experimento en cuestión.

### 5.1. Evaluación de modelos de clasificación

El objetivo es encontrar el algoritmo de clasificación con el que la tasa de acierto es más elevada. PRoNTo ofrece, como ya se ha mencionado con anterioridad, dos algoritmos: la Máquina de Vectores de Soporte y el Proceso Gaussiano. Se muestran los resultados de las pruebas realizadas con los datos de los 6 centros de los que se dispone, y aplicando una primera máscara que elimina los vóxeles que se sitúan fuera del cerebro (ver Tabla 5.1). No se ha aplicado ningún preprocesado de datos.

	Centro 1 (73 datos)	Centro 2 (188 datos)	Centro 3 (77 datos)	Centro 4 (67 datos)	Centro 5 (91 datos)	Centro 6 (149 datos)
SVM	58.9% AUC= 0.56	58.5% AUC=0.63	40.3% AUC=0.38	70.1% AUC=0.74	61.5% AUC=0.63	96.0% AUC=0.98
GPC	56.2% AUC=0.56	58.5% AUC=0.64	40.3% AUC=0.38	68.7% AUC=0.72	62.6% AUC=0.65	94.6% AUC=0.98

Tabla 5.1. Tasas de acierto obtenidas con cada máquina de clasificación para cada centro.

Si observamos los resultados obtenidos en los experimentos podemos afirmar que las tasas de acierto alcanzadas con cada método de clasificación son similares. Una variación de un 3%, teniendo en cuenta el número de datos que tenemos, puede suponer tan solo un dato menos clasificado de forma correcta. Como es conveniente elegir un método de clasificación para simplificar los experimentos, a partir de ahora se empleará el algoritmo de clasificación SVM, que es el que se utiliza habitualmente en problemas de ML.

Las tasas de acierto obtenidas son bastante malas, pero, más adelante, mejorarán gracias al resto de experimentos.

## 5.2. Evaluación de preprocesados de datos

Ahora nos centramos en comprobar que preprocesado de datos, de los que ofrece PRoNTto, es más adecuado en nuestro caso, es decir, con cual obtenemos mejores resultados para simplificar los experimentos posteriores. Las tasas de acierto obtenidas se muestran en la Tabla 5.2. Se ha empleado el algoritmo de clasificación SVM y una máscara que elimine los vóxeles situados fuera del cerebro.

El primer preprocesado es el denominado por la herramienta como: *Mean centre features using training data* y el segundo el llamado *Divide data vectors by their norm*. El manual de PRoNTto recomienda incluir, al menos, el primer preprocesado ya que resulta beneficioso. Debido a esto, no se incluyen tasas de acierto obtenidas al experimentar sin ningún preprocesado en la Tabla 5.2. Al aplicar únicamente el segundo procesado no solo disminuye la tasa de acierto notablemente si no que, además, si nos fijamos en detalle vemos que los datos de una clase siempre son acertados pero los de la

otra clase nunca lo son. Podemos observar claramente, gracias al dato de precisión de clases que ofrece la tabla de estadísticas de PRoNTto (véase Figura 5.1), que se está favoreciendo a una clase frente a otra. Por este motivo decidimos no aplicar el segundo preprocesado en solitario.

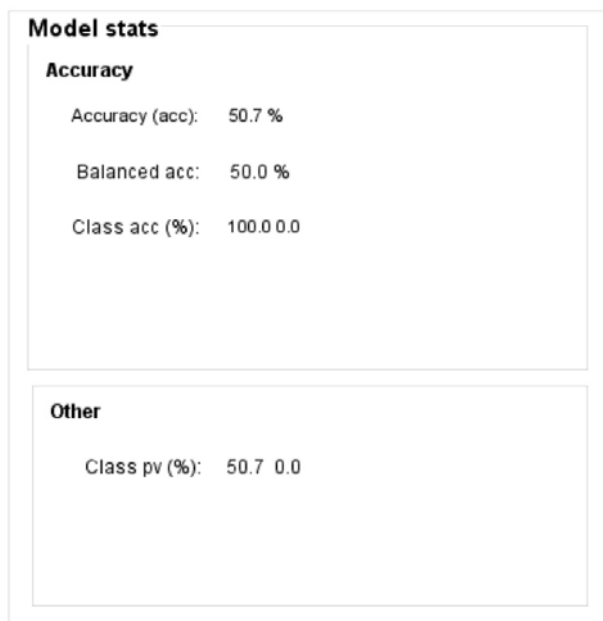


Fig. 5.1. Tabla estadística resultado de experimentar con el centro 2 y el segundo preprocesado en la *toolbox* PRoNTto.

Cuando hablamos de ambas operaciones el orden en el que se utilizan es primero *Mean centre features using training data* y después *Divide data vectors by their norm*. Esto se debe a que el orden en el que se aplican es influyente y los resultados de la segunda operación por sí sola, como acabamos de comprobar, no son buenos. Ésta solo es efectiva si se aplica después de la primera.

	Centro 1 (73 datos)	Centro 2 (188 datos)	Centro 3 (77 datos)	Centro 4 (67 datos)	Centro 5 (91 datos)	Centro 6 (149 datos)
Preprocesado 1	58.9% AUC= 0.56	58.5% AUC=0.63	46.8% AUC=0.47	68.7% AUC=0.7	61.5% AUC=0.63	96.0% AUC=0.98
Preprocesado 2	-	-	-	-	-	-
Ambos	54.8% AUC=0.55	56.4% AUC=0.61	49.4% AUC=0.37	71.6% AUC=0.7	60.4% AUC=0.65	96.0% AUC=0.98

Tabla 5.2. Tasas de acierto obtenidas con las distintas operaciones para cada centro.

Observando los resultados obtenidos llegamos a la conclusión de que, a pesar de que en algunos casos aplicando las dos operaciones no mejoren los resultados, lo más adecuado es incluirlas en los siguientes experimentos.

### 5.3. Evaluación de máscaras de selección

En este apartado se pretende observar el efecto que provoca en la clasificación aplicar una segunda máscara de selección de vóxeles. Esto puede resultar de utilidad en el caso de querer reducir el número de vóxeles que forman parte del modelo estadístico, ya que algunos de ellos pueden no contener información relevante para la clasificación. De esta manera, los vóxeles que queden fuera de la máscara serán descartados. Los experimentos se han realizado con las tres máscaras de las que disponemos, desarrolladas en [15], que son las siguientes:

- **Máscara 1:**

Esta máscara de selección de vóxeles se compone de dos valores, 1 o 0. En el caso de que el vóxel en cuestión entre dentro de la región de interés, este adquirirá el valor 1. En el caso contrario, si se queda fuera de ese subconjunto, tendrá el valor 0.

- **Máscara 2:**

La segunda máscara de la que disponemos es similar a la anterior pero, en este caso, el valor de cada vóxel oscila entre 0 y 172. El vóxel adquiere un valor igual a 0 cuando no se encuentra en la región de interés. Cuando sí se encuentra dentro de esa región, se le asigna un valor entre 1 y 172, según sea el vóxel seleccionado de relevante sobre todos los sujetos. A mayor relevancia, mayor valor.

- **Máscara 3:**

La tercera máscara utilizada es igual que la segunda pero, en este caso, con más vóxeles seleccionados. El valor de los vóxeles también se encuentra entre 0 y 172. La gran diferencia es que esta máscara es menos restrictiva, es decir, la región de interés considerada abarca mayor número de vóxeles.

En la Tabla 5.3 se pueden comprobar las tasas de acierto obtenidas para cada centro con cada máscara. Recordemos que para simplificar los experimentos y encontrar el camino más adecuado para obtener una buena clasificación de datos, y con ello un diagnóstico válido, vamos a utilizar el algoritmo de clasificación SVM y los preprocesados *Mean centre features using training data* y *Divide data vectors by their norm*, en ese orden.

	Centro 1	Centro 2	Centro 3	Centro 4	Centro 5	Centro 6	MEAN
Sin máscara	54.8% AUC=0.55	56.4% AUC=0.61	49.4% AUC=0.37	71.6% AUC=0.7	60.4% AUC=0.65	96.0% AUC=0.98	<b>64.7%</b>
Máscara 1	63.0% AUC= 0.70	64.9% AUC=0.71	40.3% AUC=0.38	67.2% AUC=0.71	48.4% AUC=0.46	92.6% AUC=0.96	<b>62.73%</b>
Máscara 2	63.0% AUC=0.70	56.4% AUC=0.61	40.3% AUC=0.38	67.2% AUC=0.71	48.4% AUC=0.46	92.6% AUC=0.96	<b>61.32%</b>
Máscara 3	60.3% AUC=0.67	67.6% AUC=0.72	40.3% AUC=0.42	71.6% AUC=0.71	59.3% AUC=0.56	92.6% AUC=0.96	<b>65.28%</b>

Tabla 5.3. Resultados obtenidos al utilizar distintas segundas máscaras de selección de vóxeles.

Cabe destacar el resultado obtenido para los centros 3, 5 y 6 que con máscara obtienen peor tasa de acierto que sin utilizarla. Esto puede deberse a que se están excluyendo vóxeles que contienen información relevante para la clasificación.

Salvo casos concretos, como los mencionados anteriormente, vemos que utilizar una segunda máscara de selección de vóxeles, como la máscara 3, puede resultar muy útil para la clasificación. Observando todos los experimentos podemos comprobar que, en la mayoría de los casos, aplicando la tercera máscara la tasa de acierto mejora, aumentando hasta un 9% para el caso de alguno de los centros objeto de estudio. Debido a esto, y como podemos ver si nos fijamos en la tasa de acierto media por máscara, la más adecuada de las tres es la tercera.

## 5.4. Evaluación multicentro

El interés del estudio multicentro radica en el hecho de que cada centro tiene su escáner, y los protocolos de adquisición son distintos. Se pretende comprobar cómo de significativas son estas diferencias.

En primer lugar se presenta la tasa de acierto obtenida tras utilizar los datos de los 6 centros disponibles de forma conjunta. Se han incluido en la clase Pacientes los sujetos correspondientes de los 6 centros y de la misma manera se ha definido la clase Controles. La máquina utilizada es una SVM y la validación en este caso es LOSO, como en los experimentos anteriores.

El resultado ha sido una tasa de acierto del 65.2% y una AUC igual a 0.73. Es un resultado bastante aceptable y demuestra, una vez más, el adecuado funcionamiento de la máquina.

A partir de ahora se ofrecen tasas de acierto obtenidas realizando diversos experimentos combinando los datos de todos los centros para asegurar el funcionamiento de la máquina ante cualquier escáner sMRI de entrada. Se analizan las diferencias entre centros y la influencia de esto al usar uno para entrenar y otro para testear (ver Tabla 5.4).

Siguiendo con la mecánica anterior, estos experimentos se realizarán utilizando un clasificador SVM, incluyendo los dos preprocesados disponibles para el caso de datos sMRI y utilizando una segunda máscara de selección de vóxeles. Sin embargo, la validación cambiará. Para comparar entre centros no tiene sentido seleccionar una validación cruzada tipo LOSO como hasta ahora. Será necesario seleccionar la opción *custom*, e introducir el archivo .mat que crearemos indicando que datos son de test y que datos son de entrenamiento.

		Training						MEAN
		Centro 1	Centro 2	Centro 3	Centro 4	Centro 5	Centro 6	
Test	Centro 1	-	52.1% AUC=0.61	41.1% AUC=0.36	54.8% AUC=0.48	54.8% AUC=0.62	52.1% AUC=0.60	<b>50.98%</b>
	Centro 2	48.9% AUC=0.50	-	53.2% AUC=0.55	47.3% AUC=0.56	52.1% AUC=0.57	55.3% AUC=0.52	<b>51.36%</b>
	Centro 3	46.8% AUC=0.41	54.5% AUC=0.60	-	58.4% AUC=0.60	57.1% AUC=0.57	46.8% AUC=0.52	<b>52.72%</b>
	Centro 4	44.8% AUC=0.45	44.8% AUC=0.48	64.2% AUC=0.61	-	47.8% AUC=0.46	68.7% AUC=0.42	<b>54.06%</b>
	Centro 5	60.4% AUC=0.59	53.8% AUC=0.60	58.2% AUC=0.56	47.3% AUC=0.55	-	58.2% AUC=0.49	<b>55.58%</b>
	Centro 6	57.7% AUC=0.64	51.0% AUC=0.46	43.6% AUC=0.42	49.0% AUC=0.40	49.0% AUC=0.53	-	<b>50.06%</b>

Tabla 5.4. Resultados obtenidos utilizando los datos de un centro para entrenar y los de otro para testear.



En general, podemos afirmar que los resultados obtenidos (ver Tabla 5.4) no son los mejores. Las tasas de error han aumentado respecto al método de validación anterior aunque era de esperar. No es lo mismo entrenar con uno grupo de datos y testear con otro grupo distinto, como en este caso, que entrenar y testear con el mismo grupo de datos, como en la validación LOSO aplicada anteriormente. Existe una dependencia con el centro. Al entrenar con los datos de todos los centros las tasas de acierto eran mejores debido a que la máquina era capaz de aprender las particularidades de cada centro.

Volviendo a los experimentos entrenando con un centro y testeando con otro, destacan las combinaciones de algunos centros que llegan a tener alrededor de un 65% de tasa de acierto, pero en la mayoría de los casos que la clasificación sea correcta depende del azar.

# Capítulo 6

## Planificación y presupuesto

---

En este capítulo se va a describir la planificación del trabajo y detallar el presupuesto necesario para hacerlo posible.

### 6.1. Planificación del proyecto

A la hora de comenzar un nuevo proyecto es muy importante tener definido el objetivo a cumplir y los requisitos necesarios para llevarlo a cabo. Para ello, es necesario llevar una organización adecuada y un seguimiento durante el proyecto de las actividades realizadas y las actividades pendientes.



#### 6.1.1. Conceptualización

La conceptualización es el primer paso a dar si se quiere realizar un nuevo proyecto. Para ello se debe identificar la idea, desarrollarla y comprobar la necesidad del proyecto, es decir, si va a aportar algo nuevo o a ayudar a lo ya existente. Tras valorar la utilidad del proyecto se puede comenzar la definición de los objetivos. Este paso es muy importante ya que si no están claros el posterior desarrollo del proyecto será más complicado.

Si conocemos dependencias de nuestro proyecto con otros anteriores puede resultarnos útil, tanto para obtener información como para tener una visión distinta relacionada con nuestro problema.

#### 6.1.2. Planificación

Tras la conceptualización, podemos avanzar a la siguiente fase, la planificación del proyecto. Es en este punto donde debemos estudiar la duración que debe tener el proyecto. Tienen que fijarse tanto una fecha de inicio como una de fin.

Para facilitar el cumplimiento de las fechas fijadas se organiza el trabajo en hitos, es decir, actividades que deben estar completadas en un momento concreto. Esos hitos pueden ser: búsqueda de información, instalación de los programas informáticos necesarios, aprender a utilizar dichos programas, plantear y realizar los experimentos para el estudio o comenzar a escribir la documentación.

En esta fase debemos contar ya con los recursos necesarios para empezar el desarrollo, ya que es el siguiente paso.

### 6.1.3. Desarrollo y ejecución

La fase de desarrollo es el momento en el que se debe poner en marcha la planificación y cumplir los hitos especificados. En las tablas que siguen a continuación se muestra una consecución de las tareas desarrolladas durante el trabajo y su duración. Seguidamente, se describe cada una de dichas tareas.

Septiembre				Octubre				Noviembre				Diciembre				Enero			
S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4

Tabla 6.1. Duración en semanas de cada tarea de los primeros 5 meses de trabajo.

Febrero				Marzo				Abril				Mayo				Junio			
S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4

Tabla 6.2. Duración en semanas de cada tarea de los últimos 5 meses de trabajo.

- **Tarea 1:** Instalación del software necesario y preparación del equipo. Duración: primera semana de Septiembre.
- **Tarea 2:** Búsqueda de información y tutoriales para conocer la herramienta. Duración: desde la segunda semana de Septiembre hasta la tercera semana de Octubre.
- **Tarea 3:** Definición de experimentos a realizar. Duración: tercera y cuarta semanas de Octubre.
- **Tarea 4:** Realización de experimentos definidos en la tarea anterior. Duración: desde la cuarta semana de Octubre hasta la segunda semana de Marzo.
- **Tarea 5:** Desarrollo de la documentación del trabajo. Duración: desde la primera semana de Febrero hasta la tercera semana de Mayo.
- **Tarea 6:** Revisión de la documentación completa. Duración: desde la cuarta semana de Mayo hasta la segunda semana de Junio.
- **Tarea 7:** Cierre del Proyecto. Duración: tercera semana de junio.

#### 6.1.4. Evaluación y cierre del proyecto

Para terminar el proyecto se debe realizar una evaluación final, pero no solo de los resultados de los experimentos concretos si no también del resultado global del proyecto. Hay que valorar si se han cumplido los primeros objetivos o si se han producido desviaciones. Para ello podemos basarnos en los siguientes KPIs (*Key Performance Indicator*):

- CPI (*Cost Performance Index*):

Este KPI indica la eficiencia en la dedicación de recursos hasta la fecha. Un valor menor del 100% muestra que se gastan más recursos de los planeados

$$CPI = \frac{\text{Valor ganado}}{\text{Coste actual}}$$

siendo el valor ganado las horas planificadas de tareas finalizadas hasta la fecha, y el coste actual la suma de las horas incurridas en las tareas finalizadas hasta la fecha.

- SPI (*Schedule Performance Index*):

Indica si un proyecto se está realizando en tiempo o se está retrasando

$$SPI = \frac{\text{Valor ganado}}{\text{Valor planeado}}$$

siendo el valor ganado las horas planificadas de tareas finalizadas hasta la fecha, y el valor planeado las tareas planificadas para cerrar.

Es conveniente realizar una evaluación del proyecto en el cierre del mismo pero también a la mitad aproximadamente. Por esta razón, se calcularon estos KPIs a los 5 meses de trabajo. El resultado indicó que el desarrollo del proceso iba con cierto retraso según lo planificado y que se consumían más horas en la realización de cada tarea de las planeadas al comienzo del trabajo.

Debido a ese retraso, el cierre del proyecto se ha producido alrededor de 3 meses más tarde de lo previsto pero se han logrado cumplir todos los objetivos iniciales.

## 6.2. Presupuesto

En esta sección se ofrece un presupuesto detallado con los recursos necesarios para la realización del proyecto. Estos recursos son tanto personales como materiales, por lo que la sección se va a dividir a su vez en dos sub-secciones más el presupuesto total.

### 6.2.1. Costes personales

Para valorar los costes personales es necesario tener en cuenta las personas que han trabajado en el desarrollo del proyecto. En este caso han sido dos: un responsable del proyecto, o tutor, y un estudiante.

Según las horas trabajadas, el presupuesto del personal concluye como sigue a continuación:

Personal	Horas Trabajadas	Honorario (€/horas)	Honorario Total (€)
<b>Responsable/ tutor</b>	144	40	5.760
<b>Estudiante (10 meses, a una media de 3 horas al día)</b>	720	20	14.400
<b>Total costes personales</b>			<b>20.160 €</b>

Tabla 6.3. Costes personales del proyecto.

### 6.2.2. Costes materiales

Durante la realización del proyecto se han necesitado diversos recursos materiales y tecnológicos cuyos costes se describen en la tabla 5.4 y su descripción es la siguiente:

- Ordenador portátil.
- Licencia Matlab.
- Licencia de Microsoft Word.
- Licencia de Adobe Acrobat.
- Toolbox PRoNTto.
- Imágenes por Resonancia Magnética de 6 centros distintos.

Concepto	Cantidad	Coste/U. (€)	Amortización	Coste (€)
<i>Ordenador portátil</i>	<i>1</i>	<i>500</i>	<i>40%</i>	<i>200</i>
<i>Licencia Matlab</i>	<i>1</i>	<i>105</i>	<i>100%</i>	<i>105</i>
<i>Licencia Microsoft Word</i>	<i>1</i>	<i>135</i>	<i>100%</i>	<i>135</i>
<i>Licencia de Adobe Acrobat</i>	<i>1</i>	<i>99</i>	<i>100%</i>	<i>99</i>
<i>Toolbox PRoNTto</i>	<i>1</i>	<i>0</i>	<i>100%</i>	<i>0</i>
<i>Datos sMRI de 6 centros</i>	<i>645</i>	<i>150</i>	<i>50%</i>	<i>48.375</i>
<b>Total costes materiales</b>				<b>48.914 €</b>

Tabla 6.4. Costes materiales del proyecto. La amortización es anual.

### 6.2.3. Presupuesto total

En este apartado se presenta el presupuesto completo del proyecto incluyendo tanto los costes de personal como los costes de material. Estos costes totales aparecen en la siguiente tabla:

Concepto	Coste (€)
<i>Costes personales</i>	<i>20.160</i>
<i>Costes materiales</i>	<i>48.914</i>
<b>Total costes PROYECTO</b>	<b>69.074 €</b>

Tabla 6.5. Presupuesto total del proyecto (sin IVA).

## Capítulo 7

# Conclusiones

---

En este trabajo, hemos hecho una presentación del Trastorno Obsesivo Compulsivo y de como el aprendizaje máquina y herramientas como PRoNTTo pueden ayudar a su diagnóstico.

El objetivo de conocer el funcionamiento de la herramienta para problemas de clasificación con datos sMRI se ha cumplido. Los experimentos realizados para el problema en cuestión han sido muy variados y completos, pudiendo conocer varias partes diferentes de la herramienta.

Asimismo, hemos comprobado que aplicar una segunda máscara de selección de vóxeles, que defina la región cerebral que aporta más información para detectar TOC, puede suponer una gran ayuda para la clasificación. De la misma manera que aplicar de forma adecuada ciertos preprocesados de datos puede resultar muy útil. En el estudio multicentro hemos analizado las diferencias entre centros y la influencia de esto al usar uno para entrenar y otro para testear. Hemos probado que la clasificación funciona con cualquier conjunto de datos sMRI de enfermos de TOC, no solo con los de un centro concreto, lo que aporta mayor valor al estudio.

Con tasas de acierto como las que hemos obtenido, podemos afirmar que este tipo de herramientas pueden ser de gran utilidad para los médicos e investigadores. Pero también son favorables para los pacientes. Un diagnóstico rápido supone aplicar antes al enfermo un tratamiento adecuado, lo que en algunos casos es de vital importancia.

La incursión del aprendizaje máquina en la medicina es cada vez mayor y con estudios como este queda demostrada su eficiencia. No solo es útil para trastornos neurológicos como este, cada vez son más las especialidades a las que puede contribuir, como son el caso de la oncología o la dermatología. Son necesarios más estudios e investigaciones para que aumente el progreso y la calidad de estas herramientas. De esta manera el uso de clasificadores de aprendizaje máquina será mucho más común en entornos clínicos. El mundo de la ingeniería y la medicina están unidos por este tipo de herramientas que, en el futuro, aumentarán su valor y desarrollo para que ambas ciencias avancen conjuntamente.

## Capítulo 8

# Conclusions

---

In this paper, we have presented a short introduction about Obsessive-Compulsive Disorder (OCD) and how machine learning (ML) techniques and toolboxes like PRoNTTo can help diagnosis.

The first objective of this thesis, to explain the operation of the toolbox for data classification problems sMRI, has been fulfilled. Experiments to the problem in question have been varied and complete. Because of this, we have studied several parts of this toolbox.

We have also found that applying a second mask of selection of voxels, which define the brain region that contributes more to detect OCD, can make a great help for classification. Also, that properly implementing certain data preprocessing can be very useful. In the multicenter study, we have analyzed the differences between centers and the influence of this if we use one of them for training and another for testing. The study has also shown that the classification works with any set of sMRI data of OCD patients. This provides greater value to the classification and, therefore, to the thesis.

We have obtained very low error rates. Because of this, we can say that these toolboxes can be very useful for clinicians and researchers. But these toolboxes are also an advantage for the patient. A quick diagnosis is good because the patient receives adequate treatment sooner and this is very important in a lot of cases.

The incursion of machine learning in medical jobs is increasing and, with studies like this, is demonstrated its efficiency. But ML is not only useful for neurological disorders like this. There are other medical specialties who work with classification tools, such as oncology and dermatology.

It is true that more studies and research are needed to increase the progress and quality of these tools. Thus the use of machine learning classifiers will be much more usual in clinical settings. The world of engineering and medicine are united by these tools which will increase its value and development for a simultaneous advance of both sciences.



# Anexos

---

## Anexo I. Instalación de PRoNTo

Para poder comenzar a experimentar con la herramienta, es necesario instalarla en el ordenador que se quiera utilizar durante el proyecto. Se puede descargar en (<http://www.mnl.cs.ucl.ac.uk/pronto/>), donde además te indican los pasos a seguir para su correcta instalación. La descarga es gratuita, tan solo hay que registrarse y adjuntar algunos datos.

Para que funcione correctamente se requiere tener ya instalado en el ordenador el siguiente software:

- MATLAB: Versión 7.5 o superior. MATLAB es un lenguaje de computación técnica de alto nivel y un entorno interactivo para desarrollo de algoritmos, visualización de datos, análisis de datos y cálculo numérico.
- SPM8, SPM12 o SPM12b: representa la aplicación de los conceptos teóricos de SPM (*Statistical Parametric Mapping*) en un paquete de análisis completo. Es necesario ya que PRoNTo se basa en funciones SPM. En el enlace anterior, donde se descarga la herramienta, también se incluye el enlace correspondiente para la descarga del paquete SPM.

Después de cumplir todos los requisitos la herramienta se puede comenzar a utilizar con normalidad.

## Anexo II. Resumen en inglés del documento

The new techniques of neuroimage, as the functional magnetic resonance imaging (fMRI) or the structural magnetic resonance imaging (sMRI), make the study and the characterization of the brain be more advanced every day. FMRI is a technique that measures the brain activity and that works by detecting changes in the flow and the oxygenation of the blood that are produced as a response of the neuronal activity. SMRI is a technique for the exam of the physical structure of the brain. These techniques don't offer more than images in 3 dimensions composed by voxels, or volumetric pixels. Thanks to the neuroimage techniques, the diagnosis of some neurological illnesses can be made easier, as the obsessive compulsive disorder (OCD). OCD is a neurological disease characterized for distressing thoughts and repetitive behaviors. In this project we count with 645 sets of data of sMRI type, either from healthy controls or OCD patients. This data has been obtained from 6 different centers.

However, analyzing this kind of images one by one means a work that gets too extended in time and requires a lot of effort by researchers and doctor. It is necessary to find techniques capable of processing these images in an automatic way.

Nowadays, an option to treat this kind of images is the Machine Learning techniques (ML). These techniques are based in a variety of statistical methods that let computers "learn" from past examples and detecting patterns hard to discern from big sets of data.

In this Bachelor Thesis is proposed a concrete tool of ML: PRoNTo (Pattern Recognition Neuroimaging Toolbox). This toolbox based on Matlab implements different techniques for recognizing patterns oriented to the analysis of neuroimage data. The toolbox allows you to experiment with different classification models, data preprocessing and voxel selection masks. Thanks to that, the results of this categorization can be very satisfactory. This can provide an effective way to automate the doctor diagnose of this type of neurological diseases and getting sooner the appropriate treatment for the patient.

The main aim of this project is to detect OCD, thanks to the sMRI data available, by using PRoNTo. To achieve the main objective, this has been divided into smaller objectives:

- 1) To know the performance and structure of the tool to use it without any problem. PRoNTo is divided into 6 main phases: "Data & Design", "Prepare Feature Set", "Specify Model", "Run Model" and "Compute Weights". Besides these phases, the toolbox counts with three more sections of

verification and revision of the introduced data and the final statistical results: “Review data”, “Review kernel & CV” and “Display results”. In this document is described each one of the phases in detail.

- 2) To understand the theoretical development of the statistical models used in this project for the binary classification of the data. Details of the support vector machine (SVM) are offered, and also about the Gaussian process (GP).
- 3) To know the data preprocessing offered by PRoNTo and which of them are applicable to classification problems’ with sMRI data, like in our case. To study if the success rate of the classification improves by using them.
- 4) To check de utility of including a selection mask of voxels in the detection of the OCD made in this Project. A voxels’ selection mask decreases the total set of voxels into a more reduced group. This way, those voxels that don’t provide information are eliminated, like voxels out of the brain. The aim is to keep the voxels in the region of interest (ROI), which are those voxels that contain any useful information for the detection of OCD.
- 5) To be capable of evaluate the final results thanks to the typically used evaluation measures in ML and that PRoNTo offers. These evaluation measures are:
  - ROC curve: In two-class classification, there is always a trade-off between class 1 and class 2 errors. Indeed, a classifier predicting class 1 regardless of input would have excellent accuracy on class 1, but bad accuracy on class2. The ROC curve is a graphical representation of this trade-off, showing how one error rate varies as a function of the other. The area under curve (AUC) is a summary measure of classifier performance, where higher is better.
  - Histogram output of the classifier: The histogram plot is a smoothed density version of the predictions plot, showing how function values are distributed. A good classifier would have minimal overlap between the densities. The error rate of the classifier is proportional to the area of the overlap.
  - Confusion matrix: The confusion matrix shows counts of predicted class labels versus true class labels. An ideal confusion matrix is diagonal: all predicted class labels correspond to the truth. Off-diagonal elements represent errors.

- Predictions plot: A prediction plot displays, for a particular fold, the output value of the machine's decision function for each test sample.
- Statistics table: The statistics table gives a summary of the model's performance. It gives some values like the accuracy, the balanced accuracy and the class accuracies.

To achieve all these aims, along this Bachelor Thesis have been made diverse experiments with all the available data.

The first experiments have been based in the study of the algorithms of ML for binary classification problems, like ours. After experimenting with the data from all the centers for SVM, the mean success rate obtained has been equal to 64.2%. For GP, the mean success rate has been 63.4%. The difference between both models of classification isn't significant. But, to simplify the rest of the experiments, has been decided to use a SVM.

The experiments keep analyzing the preprocessing of data offered by PRoNTTo. Since our problem is about sMRI data type, we have two preprocessing of data: "Mean centre features using training data" and the second called "Divide data vectors by their norm". During the experiments it has been checked that the second preprocessing by itself is not useful. Both preprocessing can be applied at the same time. We have to take into account that the order in which they are selected is the order in which they are applied. Because of this, we first have to select the operation "Mean centre features using training data" and then "Divide data vectors by their norm", since with the second one we don't obtain good results.

The mean success rate applying both preprocessing is of 64.8%. This rate is slightly better than the one obtained without applying these preprocessing. Because of this, the rest of the experiments will be made applying both preprocessing.

The experiments keep going. The next thing to be evaluated is the efficiency or not of applying a voxel selection mask. PRoNTTo obliges to introduce a mask that eliminates the voxels situated outside of the brain. These second masks with which we have experimented have tried to find the region that contributes with more information to the detection of OCD. As it is said in this document, we have worked with three different masks:

- **Mask 1:**

This voxel's selection mask is composed by two values: 1 or 0. In the case that the voxel in question is inside the region of interest, this will take

the value 1. In the other case, if it is out of this subset, it will take the value 0.

- **Mask 2:**

The second mask we have available is similar to the first one, but, in this case, the value of each voxel oscillates between 0 and 172. The voxel takes a value equal to 0 when it isn't found the region of interest. When it is inside that region, it is assigned a value between 1 and 172, depending on the relevance of the voxel between all the subjects. The bigger the relevance, the bigger the value.

- **Mask 3:**

The third mask used is the same as the second one, but, in this case, with more voxels selected. The value of the voxels is also between 0 and 172. The big difference is that this mask is less restrictive, which means that the region of interested considered covers a bigger number of voxels.

In general, the success rates obtained have improved for every one of the centers. We have obtained the best results with the third mask. Its mean success rate is equal to 65.3%.

The last experiments made have been multicenter studies. The interest of this study is the fact that each center has its own scanner and the protocols of data acquisition are different. We want to check how significant these differences are.

First, we have experimented with the data of the six centers together. The success rate has been of the 65.2%. In a result quite acceptable that demonstrates the suitable functioning of the toolbox.

The rest of experiments have been combinations between centers, using the data from one to train and from another to test. In general, the success rate has decreased. In a lot of cases the classification depends of chance. We have checked that exist dependences with the center.

After presenting all the experiments carried out, the present document continues with a description of the planning of the work and the necessary budget.

The planning of the project has been organized in four big stages:

- **Conceptualization:** In this stage must be identified the idea, develop it and check the need of the Project.
- **Planning:** In this point we must study the duration that the project must have. The work is divided in milestones, that is, in activities that must be completed before a concrete moment.
- **Development and execution:** In this stage must be executed the planning of the previous stage and accomplish the planned activities. In the correspondent section of this document are offered the tasks and their duration in detail.
- **Evaluation and closing of the project:** To finish the project must be made a final evaluation of the global result of the project. In our case, we can assure that we have achieved all the initial aims but with a little delay in time.

Referred to the needed budget to make this project, it is offered, in the corresponding section of the document, several tables indicating the cost of human resources, materials and the total cost of the project.

To finish this project, a series of conclusions are offered, being these expressed in the following paragraphs:

After carrying out all the experiments needed, we can assure that we have accomplished all the aims specified at the beginning of the project. We have managed to detect OCD thanks to PRoNTTo, obtaining acceptable success rates.

Because of this, we can say that these toolboxes can be very useful for clinicians and researchers. But these toolboxes are also an advantage for the patient. A quick diagnosis is good because the patient receives adequate treatment sooner and this is very important in a lot of cases.

The incursion of machine learning in medical jobs is increasing and, with studies like this, is demonstrated its efficiency.

It is true that more studies and research are needed to increase the progress and quality of these tools. Thus the use of machine learning classifiers will be much more usual in clinical settings. The world of engineering and medicine are united by these tools which will increase its value and development for a simultaneous advance of both sciences.

By the end of this document, there are a series of annex with useful information to the reader, such as an installation guide of the tool or this summary of the project.

## Bibliografía

---

- [1] Richard P. Swinson, Martin M. Antony, S. Rachman, and Margaret A. Richter. *Obsessive-Compulsive Disorder: Theory, Research, and Treatment*. The Guilford Press; 1998.
- [2] Tom M. Mitchell. *Machine Learning*. McGraw Hill, 1997.
- [3] Carl Edward Rasmussen, Christopher K. I. Williams. *Gaussian Process for Machine Learning*. The MIT Press 2006. ISBN 0-262-18253-X.
- [4] Burges, C.J.C. *A Tutorial on Support Vector Machines for Pattern Recognition*. *Data Min. Knowl. Discov*, 1998, vol. 2, no. 2. pp. 121-167 ISSN 1384-5810. DOI 10.1023/A:1009715923555.
- [5] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [6] Mark Ebden. *Gaussian Processes for Classification: A Quick Introduction*. 2008.
- [7] E. Alpaydin. *Introduction to Machine Learning*. The MIT Press; 2004.
- [8] T. Hastie, R. Tibshirani, J. Friedman. *The Elements of. Statistical Learning: Data Mining, Inference, and Prediction*. Second ed. Springer; 2009.
- [9] X. Zhu. *Semi-supervised learning literature survey*. University of Wisconsin; Madison: 2007.
- [10] C. Christakou, L. Lefakis, S. Vrettos, A. Stafylopatis. A Movie Recommender System Based on Semi-supervised Clustering, International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce; Vienna, Austria. 2005
- [11] J. Ashburner, C. Chu, A. Marquand, J. Mourao-Miranda, M. Monteiro, C. Phillips, J. Richiardi, J. Rondina, M. Rosa, J. Schrouff. *PRoNTTo Manual*. Disponible en web: [http://www.mnl.cs.ucl.ac.uk/pronto/prt\\_manual.pdf](http://www.mnl.cs.ucl.ac.uk/pronto/prt_manual.pdf)
- [12] *Mean Center (Spatial Statistics)*. Disponible en web: [http://resources.esri.com/help/9.3/arcgisengine/java/GP\\_ToolRef/spatial\\_statistics\\_tools/mean\\_center\\_spatial\\_statistics .htm](http://resources.esri.com/help/9.3/arcgisengine/java/GP_ToolRef/spatial_statistics_tools/mean_center_spatial_statistics.htm)



- [13] Nicolas Bourbaki. *Topological vector spaces*. Springer; 1987.
- [14] *How Mean Center (Spatial Statistics) works*. Disponible en web: [http://resources.esri.com/help/9.3/arcgisengine/java/GP\\_ToolRef/spatial\\_statistics\\_tools/how\\_mean\\_center\\_spatial\\_statistics\\_works.htm](http://resources.esri.com/help/9.3/arcgisengine/java/GP_ToolRef/spatial_statistics_tools/how_mean_center_spatial_statistics_works.htm)
- [15] Óscar García Hinde. *Bachelor Thesis. Automatic Design of Neuromarkers for Obsessive Compulsive Disorder Characterisation*. 2014. Universidad Carlos III de Madrid.
- [16] Shijun Wang, Ronald M. Summers. *Machine learning and Radiology*. 2012. Disponible en web: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3372692/>
- [17] Joseph A. Cruz, David S. Wishart. *Applications of Machine Learning in Cancer Prediction and Prognosis*. Disponible en web: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2675494/>
- [18] *Statistical Parametric Mapping*. Disponible en web: <http://www.fil.ion.ucl.ac.uk/spm/>
- [19] The University Of Sheffield. *Research techniques: Structural MRI*. Disponible en web: <https://www.sheffield.ac.uk/neuroscience/psychiatry/research/techniques/structuralmri>
- [20] UC San Diego. School of Medicine. *Structural MRI Imaging*. Disponible en web: <http://fmri.ucsd.edu/Howto/3T/structure.html>
- [21] *PRoNTo Users Archives*. Disponible en web: <https://www.jiscmail.ac.uk/cgi-bin/webadmin?A2=pronto-users;f311bbd1.1304>
- [22] Anxiety and Depression Association of America. Disponible en web: <http://www.adaa.org/understanding-anxiety/obsessive-compulsive-disorder-ocd/treatment>
- [23] Hannah Devlin. *What is a Functional Magnetic Resonance Imaging (fMRI)?* Disponible en web: <http://psychcentral.com/lib/what-is-functional-magnetic-resonance-imaging-fmri/>

- [24] *The role of neuroimaging in diagnosis and personalized medicine-current position and likely future directions*. Disponible en web:  
<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3181933/>
- [25] Andrew W. Moore. *Cross-validation for detecting and preventing overfitting*. Disponible en web: <http://www.autonlab.org/tutorials/overfit10.pdf>
- [26] FH Joanneum. *Cross-validation explained*. Disponible en web:  
<http://genome.tugraz.at/proclassify/help/pages/XV.html>
- [27] J. Guardia, L. Segura, B. Gonzalbo. *Aplicaciones de las técnicas de neuroimagen en las conductas adictivas*. Disponible en web:  
<http://www.adicciones.es/files/395-411%20Revision.pdf>
- [28] Martyn Shuttleworth. *¿Qué es la generalización?* Disponible en web:  
<https://explorable.com/es/que-es-la-generalizacion>
- [29] James H. Watt and Sjef van den Berg. *Research Methods for Communication Science. Part I, Chapter 5 - Populations and Samples: The Principle of Generalization*. Disponible en web:  
<http://www.cios.org/readbook/rmcs/ch05.pdf>
- [30] *Medicina basada en evidencias o en pruebas*. Universidad Peruana Cayetano Heredia. Disponible en web: <http://www.enlacesmedicos.com/mbese.htm>
- [31] Manuel Campos Taberner. *Evaluación de procesos Gaussianos en la estimación de parámetros biofísicos*. 2013. Universidad de Valencia. Disponible en web:  
[http://www.researchgate.net/profile/Manuel\\_Campos-Taberner/publication/259759534\\_Gaussian\\_Processes\\_evaluation\\_on\\_retrieval\\_of\\_biophysical\\_parameters/links/00b7d52dace9a054de000000.pdf](http://www.researchgate.net/profile/Manuel_Campos-Taberner/publication/259759534_Gaussian_Processes_evaluation_on_retrieval_of_biophysical_parameters/links/00b7d52dace9a054de000000.pdf)
- [32] Matthew Brett, Chris Rorden. *The analyze data format*. Disponible en web:  
<http://imaging.mrc-cbu.cam.ac.uk/imaging/FormatAnalyze>