

**Universidad Internacional de la Rioja (UNIR)**

**Escuela Superior de Ingeniería y  
Tecnología**

**Máster en Análisis y Visualización de Datos  
Masivos**

# Diagnóstico Automático de Migrañas Sin Aura y Factores Desencadenantes

**Trabajo Fin de Máster**

**Tipo de trabajo:** Desarrollo software

**Presentado por:** Arroyo Valle, Francisco José

**Directora:** Lancheros Cuesta, Diana Janeth

**Ciudad:** Zaragoza, España

**Fecha:** 19 de Septiembre de 2021

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Planteamiento	3
1.2. Estructura del documento	4
<b>2. Estado del Arte</b>	<b>5</b>
2.1. Antecedentes	5
2.1.1. Métodos de diagnóstico y clasificación de las migrañas	5
2.1.1.1. <i>Automatic migraine classification using artificial neural networks.</i>	7
2.1.1.2. <i>Classification of multi-channel EEG signals for migraine detection.</i>	8
2.1.1.3. <i>Migraine diagnosis by using artificial neural networks and decision tree techniques.</i>	10
2.1.2. Métodos de detección del estrés y el sueño mediante sensores portátiles	12
2.1.2.1. <i>Introducing WESAD, a Multimodal Dataset for Wearable Stress and Affect Detection</i>	13
2.1.2.2. <i>Continuous Stress Detection Using Wearable Sensors in Real Life: Algorithmic Programming Contest Case Study</i>	14
2.1.2.3. <i>Stress Detection Using Low Cost Heart Rate Sensors</i>	17
2.1.2.4. <i>Recent advances in wearable sensors and portable electronics for sleepmonitoring</i>	18
2.2. Artículos y <i>datasets</i> de referencia	19
2.2.1. <i>Analysis of Trigger Factors in Episodic Migraineurs Using a Smartphone Headache Diary Applications</i>	19
2.3. Conclusiones	23
<b>3. Objetivos</b>	<b>24</b>
3.1. Objetivos Generales	24
3.2. Objetivos Específicos	25

<b>4. Metodología de trabajo</b>	<b>26</b>
4.1. Extracción y limpieza de la base de datos de las migrañas . . . . .	29
4.2. Selección del modelo de aprendizaje . . . . .	30
4.2.1. Carga del conjunto de datos . . . . .	30
4.2.2. Correlación entre las características . . . . .	32
4.2.3. Conjunto de datos no balanceado . . . . .	33
4.2.4. Utilizando el modelo de regresión logística <i>multinomial</i> . . . . .	34
4.2.5. Métricas utilizadas para evaluar el rendimiento del modelo . . . . .	35
4.2.6. Todos los datos como datos de entrenamiento (Primer planteamiento) . .	36
4.2.7. División de los datos, (20 %) de prueba y el resto (80 %) de entrenamiento	37
4.2.8. Aplicar una validación cruzada <i>Stratified K-fold Cross-Validation</i> . . . . .	39
4.3. Técnicas de mejora del modelo de regresión logística <i>multinomial</i> . . . . .	41
4.3.1. Ajustar los pesos en función de la frecuencia de las clases objetivo . . . .	41
4.3.2. Ajuste del umbral óptimo del modelo . . . . .	43
4.3.2.1. <i>Utilizar la curva ROC</i> . . . . .	43
4.3.2.2. <i>Utilizar la curva de Recuperación de Precisión (Precision-Recall)</i>	47
4.3.2.3. <i>Método genérico para obtener el ajuste óptimo del umbral para</i> <i>mejorar la métrica F1 score</i> . . . . .	49
4.3.2.4. <i>Método genérico para obtener el ajuste óptimo del umbral para</i> <i>mejorar la precisión</i> . . . . .	54
4.3.3. Conclusiones de lo métodos utilizados para optimizar el rendimiento del modelo . . . . .	58
4.4. Implementación del modelo de aprendizaje con <i>Flask</i> . . . . .	59
4.4.1. Guardar y empaquetar el modelo . . . . .	60
4.4.2. Cargar el modelo en la aplicación ADMTF (Flask) . . . . .	61
4.5. Procesar los datos de salud del <i>Apple Watch</i> . . . . .	62
4.5.1. Registrar el <i>HRV</i> mediante la aplicación nativa <i>Respirar</i> . . . . .	64
4.5.2. Registrar el déficit o carencia de sueño . . . . .	65
4.5.3. Transferencia de datos desde el <i>Apple Watch</i> al smartphone <i>Iphone</i> . .	65
4.5.4. Descarga y procesado de datos en formato <b>csv</b> . . . . .	65
4.6. Diagnóstico de la <i>migraña sin aura</i> . . . . .	68
4.7. Almacenar los resultados en <i>MySQL</i> para su consulta . . . . .	71
<b>5. Resultados</b>	<b>75</b>
5.1. Resultados obtenidos del <i>Apple Watch</i> . . . . .	75

5.1.1. Gráfica de la evolución del <i>HRV</i> . . . . .	77
5.1.2. Gráfica de la evolución de la ausencia de sueño . . . . .	78
5.2. Resultados obtenidos de la aplicación . . . . .	79
5.3. Resultados obtenidos de la base de datos de la aplicación . . . . .	80
5.3.1. Visualización de los desencadenantes más habituales particularizados al usuario . . . . .	80
5.3.2. Desencadenantes para todos los usuarios . . . . .	82
<b>6. Contribución</b>	<b>83</b>
6.1. Utilización y protección de los datos . . . . .	84
<b>7. Conclusiones y trabajo futuro</b>	<b>85</b>
7.1. Conclusiones . . . . .	85
7.2. Trabajo futuro . . . . .	87
<b>8. Bibliografía</b>	<b>89</b>
Referencias . . . . .	89
<b>Appendices</b>	<b>95</b>
<b>Apéndice A. Tablas de datos y Notebooks</b>	<b>96</b>
A.1. Ficheros de tablas y datos . . . . .	96
A.2. <i>Notebooks</i> y repositorios . . . . .	97
A.3. Fichero con el modelo obtenido . . . . .	98



## Resumen

El objetivo de este trabajo es diagnosticar las **migrañas** sin aura en personas que sufren cefaleas de manera habitual y así poder identificarlas respecto a otro tipo de cefaleas. El diagnóstico se realiza mediante una aplicación que utiliza técnicas de aprendizaje automático, clasificación y visualización.

La base de conocimiento de la aplicación utiliza casos reales para clasificar la migraña sin aura utilizando el criterio *1.1 Migraine without aura* del artículo (“Headache Classification Committee of the International Headache Society (IHS) The International Classification of Headache Disorders, 3rd Edition”, 2018).

La aplicación, como novedad para el diagnóstico, integra en la base de datos los **desencadenantes** más habituales de las migrañas mediante su registro y visualización, para así conocer los más habituales asociados a cada persona y tratar de evitarlos en lo posible. El registro de estos desencadenantes está implementado mediante un cuestionario en la aplicación donde se han incluido los más habituales según los estudios ya realizados, con la posibilidad de añadir alguno distinto, y así poder generar una visualización particularizada a cada persona con cefalea.

Además, en la aplicación, se integra la posibilidad responder de manera lo más objetiva posible a un par de las cuestiones planteadas; *¿Ha sido el **estrés** uno de los desencadenantes o no? ¿Ha sido la **carencia de sueño** uno de los desencadenantes o no?*

El estrés y la carencia de sueño son unos de los principales desencadenantes de las migrañas como puede comprobarse en todos los estudios relacionados con este trabajo y su diagnóstico se integra en la aplicación mediante la captura y análisis de datos de salud extraídos de un reloj de pulsera comercial. En este caso, el seleccionado ha sido el **Apple Watch**, que es más común y accesible que otros dispositivos de salud más específicos para el diagnóstico del estrés y la falta de sueño.

**Palabras clave:** Migraña, Desencadenante, Estrés, Apple Watch, Diagnóstico

## Abstract

The goal of this work is to diagnose **migraines** without aura in people who suffer often headaches and thus be able to identify them with respect to others headaches. The diagnosis is possible by an application that uses machine learning, classification and visualization methods.

The knowledge base of the application uses real cases to classify the migraines without aura using criteria based on *1.1 Migraine without aura* from the article (“Headache Classification Committee of the International Headache Society (IHS) The International Classification of Headache Disorders, 3rd Edition”, 2018).

As an innovation, the database of the application integrates for diagnosis the most common **triggers** of migraines to be able to study them in order to know the most common triggers associated with each headache person and so they could be avoided as far as possible. The registration of these triggers is implemented through a questionnaire in the application based on most common triggers, which have been included according to studies already carried out, with the possibility of adding a different one, and thus be able to generate a specific study to each headache person.

In addition, the application includes the possibility of objectively answering a couple of questions; *Has **strees** been a trigger or not? Has **the lack of sleep** been a trigger or not?*

Stress and lack of sleep are the main triggers of migraines as we can see in all the studies related about this work and its diagnosis is integrated into the application by capturing and analyzing of health data extracted from a commercial wrist watch. In this case, the selected one has been the **Apple Watch**, which is more common than other specific health devices for the diagnosis of stress and lack of sleep.

**Keywords:** Migraine, Trigger, Stress, Apple Watch, Diagnostic

# Índice de figuras

1.	GBD Global Compare (1990-2019) years - Females between (15-49) years - YLDs (Years Lived with Disability) per 100000. . . . .	2
2.	Lista de variables utilizadas como entrada de la red neuronal. . . . .	8
3.	Posición de los electrodos del <b>EEG</b> siguiendo el sistema 10-20. . . . .	9
4.	Diagrama de flujo del diario completo de registros de pacientes con migrañas (Park et al., 2016). . . . .	20
5.	Fragmento de la tabla original (Park et al., 2016) <i>total diary 4579</i> en formato Excel. . . . .	20
6.	Diagrama general de la metodología de trabajo. . . . .	28
7.	Fragmento de la tabla <i>totalDiary4578Filtered</i> . . . . .	29
8.	Correlación entre las características del conjunto de datos. . . . .	32
9.	Distribución de las clases en el <i>dataset</i> . . . . .	33
10.	Matriz de confusión - Regresión Logística Multinomial - Sin separar el <i>conjunto de datos</i> . . . . .	36
11.	Matriz de confusión - Regresión Logística Multinomial - 80 % de entrenamiento y 20 % de test. . . . .	38
12.	Curva ROC de la clase <i>Probable Migraine</i> utilizando la estadística <i>J</i> . . . . .	45
13.	Curva Precision-Recall de la clase <i>Probable Migraine</i> utilizando la estadística <i>F1</i> . . . . .	48
14.	Umbral óptimo 0.414, método paso a paso de la clase <b>No Migraña</b> utilizando la métrica <b>F1</b> . . . . .	51
15.	Umbral óptimo, método paso a paso de la clase <b>Probable Migraña</b> utilizando la estadística <b>F1</b> . . . . .	52
16.	Umbral óptimo, método paso a paso de la clase <b>Migraña</b> utilizando la métrica <b>F1</b> . . . . .	53
17.	Diagrama de bloques funcional para obtener los umbrales óptimos que mejoran el rendimiento del modelo mejorando la <i>precisión</i> . . . . .	55

18. Evolución de los umbrales óptimos para la precisión <i>precisión</i> de las clases minoritarias. . . . .	58
19. Diagrama funcional de la aplicación <b>ADMTF</b> enfocado en la carga del modelo de aprendizaje. . . . .	61
20. Esquema de la metodología seguida para procesar los datos del Apple Watch. . . . .	63
21. Descarga de fichero de datos con la métricas de sueño y HRV desde la aplicación <b>AutoSleep</b> . . . . .	66
22. Procedimiento de carga de más datos de salud procedentes del <i>Apple Watch</i> en la aplicación <b>ADMTF</b> . . . . .	66
23. Ventana de la aplicación <b>ADMTF</b> para lanzar el test de diagnóstico de las migrañas. . . . .	68
24. Ventana de la aplicación <b>ADMTF</b> correspondiente a la primera de las preguntas del test. . . . .	70
25. Ventana de la aplicación <b>ADMTF</b> correspondiente a la séptima de las preguntas del test. . . . .	70
26. Ventana de la aplicación <b>ADMTF</b> correspondiente a la octava de las preguntas del test, la relacionada con los desencadenantes más habituales. . . . .	71
27. Estructura de la base de datos de la aplicación <b>ADMTF</b> . . . . .	72
28. Fragmento de datos obtenidos después del procesarlos, destinados al análisis del estrés y la carencia de sueño de la aplicación ( <i>AutoSleep. Sleep Tracking, s.f.</i> ). . . . .	75
29. Captura de la aplicación en el proceso de cargar de datos para el análisis del estrés y la carencia de sueño. . . . .	76
30. Evolución del HRV obtenida de la aplicación. . . . .	77
31. Evolución del déficit de sueño obtenida de la aplicación. . . . .	78
32. Ventana de la aplicación <b>ADMTF</b> correspondiente al resultado del test de detección de la migraña sin aura. . . . .	79
33. Desencadenantes más habituales de las migrañas para los usuarios <i>cmc-0031</i> y <i>cmc-0042</i> . . . . .	81
34. Desencadenantes más habituales de las migrañas para todos los usuarios de la base de datos. . . . .	82

# Índice de Tablas

1.	Antecedentes de otros clasificadores de migrañas. . . . .	6
2.	Antecedentes correspondientes a los diferentes métodos de detección del estrés y la carencia del sueño mediante dispositivos portátiles. . . . .	12
3.	Características extraídas de la variabilidad del ritmo cardiaco (HRV) usando la herramienta HRVTool (Vollmer, 2019). . . . .	15
4.	Artículos de referencia de los desencadenantes naturales de las migrañas. . . . .	21
5.	Regresión Logística Multinomial - Métricas obtenidas con el 80 % de entrenamiento y 20 % de test. . . . .	39
6.	Regresión Logística Multinomial - Métricas promedio de la validación cruzada de los datos en 8 fracciones distribuidas. . . . .	41
7.	Regresión Logística Multinomial - Métricas promedio de la validación cruzada con <i>class_weight = 'balanced'</i> . . . . .	42
8.	Regresión Logística Multinomial - Métricas promedio con umbrales óptimos obtenidos de las curvas ROC. . . . .	47
9.	Regresión Logística Multinomial - Métricas promedio con umbrales óptimos obtenidos de la curva de <i>recuperación de precisión</i> . . . . .	49
10.	Regresión Logística Multinomial - Métricas promedio con umbrales óptimos para la precisión <i>accuracy</i> . . . . .	57
11.	Métricas obtenidas del Apple Watch mediante al aplicación AutoSleep. . . . .	76
A1.	Descripción de los ficheros de datos y tablas disponibles en <i>GitHub</i> . . . . .	96
A2.	Descripción de los cuadernos de <i>Jupyter</i> . . . . .	97
A3.	Descripción del fichero con el modelo de aprendizaje empaquetado. . . . .	98

## Lista de fragmentos de código

4.1. Carga del <i>dataset</i> de las migrañas en el cuaderno de <i>Jupyter</i> . . . . .	30
4.2. Instancia del modelo elegido de regresión logística multinominal. . . . .	34
4.3. Método de división simple de los datos 20 % para test y el resto de entrenamiento. . . . .	38
4.4. Método validación cruzada de los datos en 8 fragmentos y cálculo de las métricas promedio. . . . .	39
4.5. Método validación cruzada con el parámetro <i>class_weight = 'balanced'</i> en el <i>constructor</i> . . . . .	41
4.6. Curvas ROC obtenidas utilizando la estadística <i>J</i> con un <i>dataset</i> 80 % de entrenamiento y 20 % de prueba. . . . .	44
4.7. Método de cálculo de las métricas en función de los umbrales óptimos. . . . .	46
4.8. Fragmento de código para obtener la curva de recuperación de precisión utilizando la métrica <i>F1</i> . . . . .	48
4.9. Método genérico para obtener los umbrales óptimos para la métrica <i>F1</i> . . . . .	50
4.10. Método genérico para obtener los umbrales óptimos para la precisión. . . . .	56
4.11. Empaquetado del modelo utilizando <i>Joblib Accuracy</i> . . . . .	60
4.12. Procesado de los datos del <i>Apple Watch</i> en la aplicación <b>ADMTF</b> . . . . .	67
4.13. Fragmento de código del proceso de guardar los resultados en la base de datos de la aplicación <i>ADMTF</i> . . . . .	73

# Capítulo 1

## Introducción

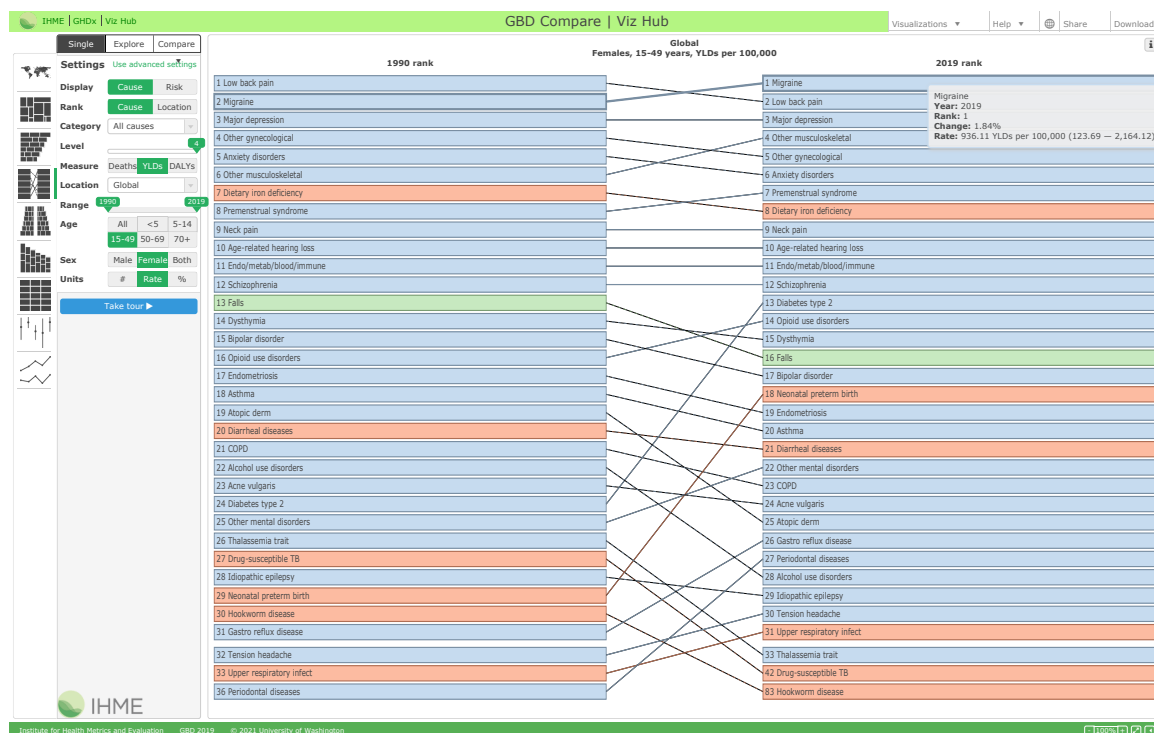
Según la OMS (*Headache Disorders*, s.f.) y (Ahmed, 2012) entorno al 50 % de las personas han padecido algún dolor de cabeza este último año, de los cuales se calcula que un tercio son migrañas. La migraña es un tipo de dolor de cabeza recurrente que comienza a partir de la adolescencia y que alcanza su momento de mayor incidencia a las edades comprendidas entre los 30 y 50 años, afectando fundamentalmente a las mujeres.

Además de tener una gran incidencia, provoca una discapacidad laboral evidente en el rango de edad más productivo convirtiéndose en la primera causa de discapacidad como puede verse en la [Figura 1](#), sin embargo, el hecho de no ser una enfermedad grave, en el sentido de que pueda provocar la muerte, le ha restado la importancia social-económica que tiene.

Desde un punto de vista económico el impacto de las migrañas es evidente, como referencia el estudio reciente realizado sobre la población alemana (Seddik et al., 2020) se apunta que el gasto promedio por paciente al año debido a las migrañas en Alemania es de **6493 €**. En otro artículo de referencia (Ahmed, 2012) se indica que se pierden entorno a **25 millones** de días de trabajo al año en el Reino Unido con un coste estimado de **2 billones de £**, lo cual da idea del impacto económico de esta afección del sistema nervioso.

Teniendo en cuenta que la gran mayoría de las migrañas corresponden a las *migrañas sin aura* (Ruiz et al., 2010), más del doble respecto a las migrañas con aura, uno de los objetivos de este trabajo es diagnosticar las mismas siguiendo los criterios de *1.1 Migraine without aura* del artículo (“Headache Classification Committee of the International Headache Society (IHS) The International Classification of Headache Disorders, 3rd Edition”, 2018) para poder distinguirlas con precisión respecto a otro tipo de cefaleas.

**Figura 1.** GBD Global Compare (1990-2019) years - Females between (15-49) years - YLDs (Years Lived with Disability) per 100000.



*Nota.* Esta gráfica describe a escala global la evolución desde 1990 hasta el año 2019 de las principales causas que provocan discapacidad en años por cada 100000. El filtro se ha hecho para mujeres con un rango de edades de 15 a 49 años.

*Fuente.* Tomado de (Institute for Health Metrics and Evaluation, s.f.)(GBD Compare | IHME Viz Hub, s.f.).

Por otra parte, diferentes estudios como los de (Marmura, 2018) y (Kelman, 2007), han encontrado una relación entre las migrañas y una serie de factores **desencadenantes** que las provocan. Entre los desencadenantes naturales más habituales, aunque no son únicos y cada persona puede tener los suyos, se destacan los siguientes: **estrés, sueño, ejercicio, fatiga, hormonales, emocionales, tiempo, luz, ruido, olores, comidas irregulares, cafeína, alcohol, comer chocolate/queso...**

Como se indica en (Marmura, 2018), estos desencadenantes pueden ser difíciles de predecir y varían mucho de unas personas a otras pudiendo ser confundidos con factores premonitorios de la migraña como pueden ser los cambios de humor u otros factores. Esta confusión, puede provocar frustración a la hora de tratar de limitar el efecto de los verdaderos desencadenantes, provocando un empeoramiento de la calidad de vida ya de por sí deteriorada. Para tratar de reducir esta frustración en lo posible y como novedad se integran estos desencadenantes en el procedimiento de diagnóstico de las *migrañas sin aura*.

Dentro de los factores desencadenantes más habituales, como indica (Kelman, 2007) el más destacado es el **estrés**.



Hay que tener en cuenta que el *estrés* no solamente puede desencadenar ataques de *migraña*, sino que también se le relaciona con el enorme incremento de otros problemas muy importantes para la sociedad en los últimos 30 años, como la *ansiedad* y la *depresión* según la **OMS** (*La inversión en el tratamiento de la depresión y la ansiedad tiene un rendimiento del 400 %*, s.f.).

Otro de los principales desencadenantes como indica (J. Kim et al., 2017) es la *carencia de sueño*, el cual, es particularmente destacado como desencadenante en las migrañas frente a otra tipo de dolores de cabeza.

Estos 2 factores desencadenantes principales de las *migrañas*, como son el *estrés* y la *carencia de sueño* han sido seleccionados en este trabajo para poder determinar de una manera objetiva si han podido formar parte o no de la *migraña*. Ambos, pueden ser monitorizados mediante dispositivos de muñeca comerciales que hoy en día disponen de sensores de salud bastante completos, para medir la calidad del sueño, el ritmo cardiaco, etc. Además, parten con la ventaja de ser más asequibles que otros con funcionalidades más enfocadas dentro del campo de las salud como el *Empatica E4*.

Entre estos dispositivos comerciales, el utilizado en este estudio es el *Apple Watch*.

## 1.1. Planteamiento

En este trabajo, se propone como objetivo desarrollar de una aplicación que permita diagnosticar *migrañas sin aura*, registrando los factores que las desencadenan en una base de datos específica a cada persona. La base de datos permite de una manera visual la consulta de los desencadenantes que provocan las migrañas.

Además, la aplicación trata de determinar de manera objetiva si el *estrés* y la *carencia de sueño* han podido desencadenar la migraña o no mediante el análisis de los datos obtenidos de un reloj de pulsera comercial como el *Apple Watch*.

## 1.2. Estructura del documento

A partir de ahora, este documento, está estructurado en una serie de capítulos que se describen a continuación:

- **2 Estado del Arte.** En este capítulo se exponen los estudios y datos ya existentes relacionados con este trabajo, los cuales son necesarios para un correcto desarrollo de la aplicación.
- **3 Objetivos.** Donde se describen los objetivos generales y específicos de este trabajo.
- **4 Metodología de trabajo.** Aquí se exponen los pasos que se han llevado a cabo para el desarrollo de la aplicación, procesos de captura, preparación de los datos, algoritmos de aprendizaje automático utilizados, software seleccionado para realizar la aplicación y para gestionar la base de datos...
- **5 Resultados.** Se muestran los resultados obtenidos siguiendo la metodología planteada en el capítulo anterior.
- **6 Contribución.** Cual ha sido la contribución de este trabajo, en este caso una aplicación que permite un diagnóstico simple de las *migrañas sin aura*, así como el tratamiento de los datos que han sido utilizados en la misma.
- **7 Conclusiones y trabajo futuro.** Dentro de este capítulo se describen las conclusiones obtenidas a través de la relación entre los objetivos planteados y los resultados obtenidos, además de describir las diferentes vías de trabajo futuro a desarrollar basadas en las ya desarrolladas en este documento.
- **8 Bibliografía.** Capítulo correspondiente a la bibliografía obtenida como referencia.
- **A Tablas de datos y Notebooks.** Apéndice destinado a indicar el acceso a los documentos con las tablas y *notebooks* utilizados en este trabajo, así como una breve descripción de los mismos.

## Capítulo 2

# Estado del Arte

Este capítulo, describe la situación actual relacionada con este trabajo, haciendo referencia a los antecedentes y artículos que han permitido obtener una base de conocimiento sólida para la realización de la contribución desarrollada en [4 Metodología de trabajo](#).

Para ello, primero se exponen los antecedentes relacionados con este trabajo, después se profundiza en los artículos y conjuntos de datos de referencia utilizados para el diagnóstico de las *migrañas sin aura*, el *estrés* y la *carencia de sueño*.

### 2.1. Antecedentes

#### 2.1.1. Métodos de diagnóstico y clasificación de las migrañas

Hay una serie de artículos relacionados con la creación de diferentes modelos de aprendizaje automático como se puede ver en la [Tabla 1](#), los cuales permiten realizar una clasificación de los distintos tipos de migrañas, aunque, y como denominador común, todos ellos clasifican la *migraña sin aura*, ya que es la más habitual. Excepto el (Akben et al., 2016), que basa la clasificación en el análisis de los datos que ofrece el electroencefalograma, los otros dos (Sanchez-Sanchez et al., 2020) y (Celik et al., 2014) utilizan criterios similares a los utilizados en este trabajo, es decir, siguiendo los criterios de [1.1 Migraine without aura](#) del artículo (“Headache Classification Committee of the International Headache Society (IHS) The International Classification of Headache Disorders, 3rd Edition”, 2018).

Sin embargo, ninguno de ellos propone incorporar a la base de datos del clasificador los *desencadenantes* más comunes de las *migrañas* para así poder disponer de una base de conocimiento específica para cada usuario respecto a la causa de sus *migrañas*.

**Tabla 1. Antecedentes de otros clasificadores de migrañas.**

<b><i>Automatic migraine classification using artificial neural networks</i></b>	
Referencia	(Sanchez-Sanchez et al., 2020)
Tipo de Migraña	Migraña sin aura, Migraña con aura, Migraña hemipléjica, Otras.
Modelo	ANN - Red neuronal.
Desencadenantes	No Incluidos.
Enfoque	Clasificación de varios tipos de migrañas usando una red neuronal.
<b><i>Classification of multi-channel EEG signals for migraine detection</i></b>	
Referencia	(Akben et al., 2016)
Tipo de Migraña	Migraña sin aura.
Modelo	SVM - Support Vector Machine.
Desencadenantes	No Incluidos.
Enfoque	Clasificación de migrañas sin aura basado en la captura del EEC - Electroencefalograma.
<b><i>Migraine diagnosis by using artificial neural networks and decision tree techniques</i></b>	
Referencia	(Celik et al., 2014)
Tipo de Migraña	Migrañas sin aura y posibles migrañas.
Modelo	Árboles de decisión.
Desencadenantes	No Incluidos.
Enfoque	Clasificación de Migrañas sin aura y probables migrañas usando modelos de árboles de decisión.

*Fuente.* Adaptado de (Sanchez-Sanchez et al., 2020), (Akben et al., 2016) y (Celik et al., 2014).

A continuación, y de manera resumida, se desarrollan los artículos precedentes relacionados con métodos de clasificación de las migrañas que han sido recopilados en la [Tabla 1](#) para el estado del arte.

#### **2.1.1.1. *Automatic migraine classification using artificial neural networks.***

El objetivo de este artículo ([Sanchez-Sanchez et al., 2020](#)), es lograr un método de clasificación de diferentes tipos de migrañas mediante una red neuronal, facilitando una mayor accesibilidad para la obtención de unos resultados que los métodos habituales basados en el estudio de ondas cerebrales. En muchas ocasiones, el diagnóstico habitual se basa en la experiencia del médico y en el tiempo disponible para realizar las pruebas, pudiendo provocar errores. Estos errores provocan tratamientos inapropiados e innecesarios que pueden causar el efecto contrario en los pacientes con migrañas.

La red neural utilizada como clasificador permite obtener niveles de precisión **>97 %** y mejora las de otros clasificadores utilizados, como árboles de decisión, métodos de regresión o *SVM-Support Vector Machine* obteniendo diagnósticos precisos y rápidos de los siguientes 7 tipos de afecciones:

- Migraña con aura.
- Migraña sin aura.
- Aura sin migraña.
- Migraña hemipléjica familiar.
- Migraña hemipléjica esporádica.
- Aura de tipo basilar.
- Otras.

Los datos para la realización del clasificador mediante la red neuronal fueron obtenidos del Hospital Materno Infantil de Soledad en Colombia, año 2013 y están recopilados en la tesis ([Romero De la Hoz y Rúa Ascar, 2014](#)).

Para el desarrollo de la red neuronal se ha utilizado la librería de *python* gratuita *Scikit-learn* ([Pedregosa et al., 2011](#)) utilizando 23 variables de entrada, descritas en la [Figura 2](#), y 7 de salida de la red neuronal que corresponden a las afecciones buscadas.

**Figura 2.** Lista de variables utilizadas como entrada de la red neuronal.

F1000Research 2020, 9:618 Last updated: 16 NOV 2020

**Table 1.** List of identified variables.

	Description	Name		Description	Name
1	Patient's age	Age	13	Lack of speech coordination	Dysphasia
2	duration of last episode in days	Duration	14	Disarticulated sounds and words	Dysarthria
3	Frequency of episodes per month	Frequency	15	Dizziness	Vertigo
4	Unilateral or bilateral pain location	Location	16	ringing in the ears	Tinnitus
5	Throbbing or constant pain	Character	17	Hearing loss	Hypoacusis
6	Pain intensity, i.e., mild, medium, or severe	Intensity	18	Double vision	Diplopia
7	Nauseous feeling	Nausea	19	Simultaneous frontal eye field and nasal field defect and in both eyes	Visual defect
8	Vomiting	Vomit	20	Lack of muscle control	Ataxia
9	Noise sensitivity	Phonophobia	21	Jeopardized conscience	Conscience
10	Light sensitivity	Photophobia	22	Simultaneous bilateral paresthesia	Paresthesia
11	Reversible visual symptoms	Visual	23	Family background	Family
12	Reversible sensory symptoms	Sensory	24	Diagnosis of migraine type	Type

Fuente. Tomado de (Sanchez-Sanchez et al., 2020).

En este artículo no se han incluido como entrada los disparadores más comunes de las migrañas.

Por otra parte, para la utilización del modelo basado en la red neuronal es necesario tener unos ciertos conocimientos en *python*, lo que limita su utilización generalizada.

### 2.1.1.2. Classification of multi-channel EEG signals for migraine detection.

En este artículo (Akben et al., 2016), se ha buscado un método de clasificación utilizando los datos disponibles de posicionamiento de los electrodos del electroencefalograma (EEG), el estudio se ha basado en el estándar 10-20 (Homan, 1988) para la colocación de los electrodos en el cuero cabelludo.

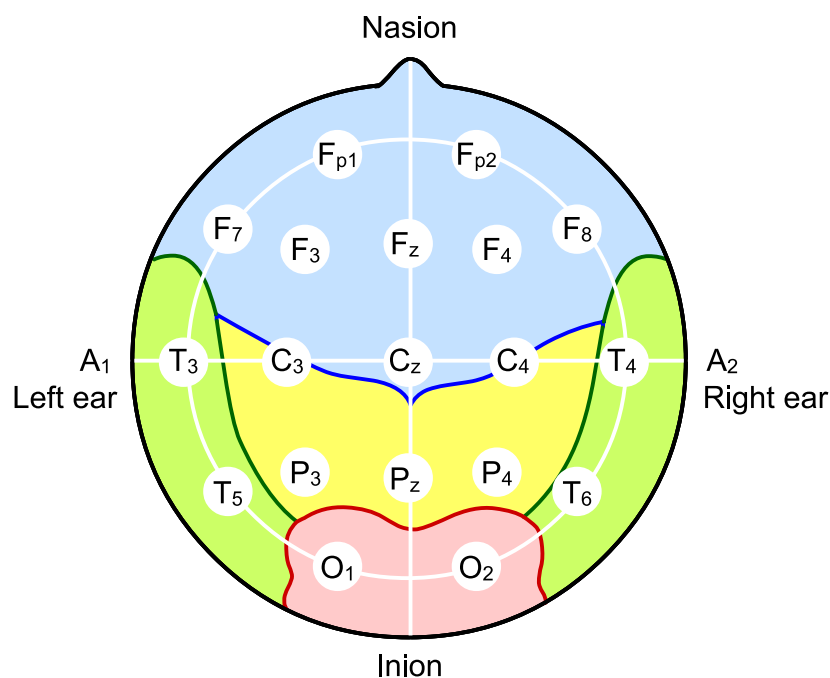
El modelo de clasificación utilizado en este artículo de referencia es un modelo de SVM (Support Vector Machines), es un buen modelo de clasificación sobre todo cuando el número de variables es elevado respecto a los datos de entrenamiento (Roy et al., 2015) aunque no es muy eficiente para grandes conjuntos de datos (Cervantes et al., 2008).

Los datos de las personas que han sido utilizados en este modelo de aprendizaje, fueron extraídos del departamento de neurología de la universidad turca *Kahramanmaraş Sutcu Imam University* y se agrupan de la siguiente manera:

- 30 personas con *migrañas* sin aura, de las cuáles 9 eran hombres y 21 mujeres.
- 30 personas sanas, de las cuáles 19 eran mujeres y 11 hombres.

Los datos del estudio fueron capturados mediante un EEG (electroencefalograma) siguiendo como criterio de colocación de los electrodos el estándar 10-20 (Homan, 1988) con la distribución craneal para los electrodos siguiente Figura 3:

**Figura 3.** Posición de los electrodos del **EEG** siguiendo el sistema 10-20.



Fuente. Tomado de (Plass-Oude Bos, 2006).

El objetivo del estudio es determinar qué electrodos, además de los T3 y T5 que ya se había demostrado en otros estudios su prevalencia, son los más importantes para realizar las predicciones de las migrañas sin aura. Los resultados obtenidos mediante el clasificador SVM indican que los electrodos **T3**, **F7**, **O1** y **O2** son los más determinantes para el diagnóstico de migraña ya que obtienen unos niveles de precisión más elevados, desde el **81.8%** para el **O2** hasta el **88.4%** para el **T3**.

Es un estudio que obtiene muy buenos niveles de clasificación de las *migrañas sin aura* en los electrodos indicados, y además de manera novedosa. Aunque desde el punto de vista del autor de este documento, se necesita una infraestructura y medios técnicos que dificultan la realización de un método de clasificación de las *migrañas sin aura* rápido y generalizado utilizando el EEG.

### 2.1.1.3. *Migraine diagnosis by using artificial neural networks and decision tree techniques.*

En este artículo (Celik et al., 2014), se busca como objetivo principal tratar de reducir el error debido al factor humano en lo posible para el diagnóstico de las *migrañas sin aura*.

El conjunto de datos original consta de 70 personas, realizando una distribución de un 70 % para los datos de entrenamiento y un 30 % para los de prueba. De las 70 personas, la distribución de población es de 16 pacientes con migraña, 33 con probable migraña y 21 pacientes sin migraña. Para el desarrollo del modelo de aprendizaje automático, se han utilizado diferentes técnicas como son:

- *GINI* (árboles de decisión).
- *DDN* (red de retardo distribuida).
- *PNN* (red neuronal probabilística).
- *FFN* (red de alimentación).
- *LVQ* (aprendizaje de cuantificación de vectores).

Los mejores resultados se obtuvieron con el modelo basado en DNN, con una precisión del **95.45 %**. Después se amplió el conjunto de datos que llevó a un aumento en el error de las predicciones, sobre todo en el árbol de decisión, asociadas según los autores a la mala distribución en el nuevo conjunto de datos existente.

Se utilizaron los criterios de clasificación de las *migrañas sin aura* y de las *probables migrañas sin aura* de la beta *ICHD-3 (beta)* (“The International Classification of Headache Disorders, 3rd Edition (Beta Version)”, 2013), los cuales, son los mismos que los utilizados en este trabajo correspondientes a la versión actual del *ICHD-3* (“Headache Classification Committee of the International Headache Society (IHS) The International Classification of Headache Disorders, 3rd Edition”, 2018). Las diferencias en cuanto a los métodos de diagnóstico de las migrañas entre una versión y otra son mínimas y están centradas en las *migrañas con aura* donde se han añadido criterios más específicos (Göbel et al., 2020).

Hay que subrayar, que no se han incluido los desencadenantes, mientras que si han sido incluidos en este trabajo. Por ello, el número de atributos es más reducido, un total de 8:

- Mínimo 5 ataques de migraña anteriores.
- Duración entre 4 y 72 horas.



- Dolor de cabeza unilateral.
- Dolor de cabeza pulsante.
- Nivel de dolor moderado o severo.
- Dolor de cabeza que se incrementa con el movimiento.
- Náuseas y/o vómitos.
- Fotofobia o fonofobia.

Existen 3 clases de salida como son **migraña**, **no-migraña** y **posible-migraña** análogas a las utilizadas en este trabajo. Por otra parte, en este artículo no queda definido cómo han de utilizarse los modelos obtenidos para otros conjuntos de datos.

### 2.1.2. Métodos de detección del estrés y el sueño mediante sensores portátiles

Teniendo en cuenta que entre los desencadenantes naturales más importantes de las migrañas (Marmura, 2018) se encuentran el *estrés* y la *carencia de sueño*, a continuación y como antecedentes a este estudio en la [Tabla 2](#) se describen los artículos relacionados con los diferentes métodos para la detección de los mismos mediante sensores portátiles similares al *Apple Watch* utilizado en este estudio:

**Tabla 2.** Antecedentes correspondientes a los diferentes métodos de detección del estrés y la carencia del sueño mediante dispositivos portátiles.

<b><i>Introducing WESAD, a Multimodal Dataset for Wearable Stress and Affect Detection</i></b>	
Referencia	(Schmidt et al., 2018)
Dispositivos utilizados	RespiBAN Professional y Empatica E4.
Sensores	Electrocardiograma (ECG), actividad electrodérmica (EDA), electromiograma (EMG), respiración, temperatura corporal, acelerómetro en los 3 ejes y volumen sanguíneo (BVP).
Enfoque	Clasificación binaria ( estrés / no estrés ) mediante la aplicación de una serie de actividades a las 15 personas que se sometieron al estudio.
<b><i>Continuous Stress Detection Using Wearable Sensors in Real Life: Algorithmic Programming Contest Case Study</i></b>	
Referencia	(Can et al., 2019)
Dispositivos utilizados	Samsung Gear S1 / S2 / S3 y Empatica E4.
Sensores	PPG (photoplethysmography), actividad electrodérmica (EDA), temperatura y acelerómetro en los 3 ejes.
Enfoque	Durante 9 días participaron 21 personas en el estudio para clasificar 3 niveles de estrés: Día libre, conferencia y concurso, asignando el nivel de estrés más bajo al día libre, el intermedio para la conferencia y el alto para el concurso.
<b><i>Stress Detection Using Low Cost Heart Rate Sensors</i></b>	
Referencia	(Salai et al., 2016)
Dispositivos utilizados	CardioSport TP3 Heart Rate Transmitter y el Schiller MT-101/MT-200.
Sensores	Frecuencia cardiaca e intervalos RR.

Enfoque	5 personas fueron conectadas a ambos aparatos el económico CardioSport y el dispositivo médico de referencia Schiller MT-101/MT-200 durante 24 horas, donde después de un preprocesado y sincronización de ambos sensores se obtuvo una correlación entre ambos para detectar el estrés con una precisión alrededor del 75 %.
<b><i>Recent advances in wearable sensors and portable electronics for sleep monitoring</i></b>	
Referencia	(Kwon et al., 2021)
Dispositivos utilizados	Empatica E4, Galaxy Watch 3 y Apple Watch 6 entre otros.
Sensores	Saturación de oxígeno y actividad electrodérmica (EDA).
Enfoque	Expone los avances en los dispositivos portátiles de muñeca, mucho más económicos y asequibles que el estándar de medicina utilizado para hacer mediciones de la calidad del sueño como es el PSG (Polysomnography).

*Fuente.* Adaptado de (Schmidt et al., 2018), (Can et al., 2019), (Salai et al., 2016) y (Kwon et al., 2021).

### 2.1.2.1. ***Introducing WESAD, a Multimodal Dataset for Wearable Stress and Affect Detection***

El objetivo del estudio (Schmidt et al., 2018) del conjunto de datos **WESAD** es realizar una clasificación binaria (estrés/no estrés) mediante la aplicación de una serie de actividades a las que se sometieron a 15 individuos. Estas actividades están identificadas como *Línea de base*, *Estrés*, *Descanso*, *Meditación*, *Diversión* y *Meditación 2*.

En el procedimiento de captura de datos se realizaron 2 tipos de prueba A y B para evitar los efectos provocados por utilizar siempre el mismo orden en las actividades, los participantes A fueron sometidos a un orden de actividades distinto a los participantes B.

Los datos están estructurados en 13 carpetas, cada una de ellas contiene los ficheros con los sensores, test de preguntas, etc, que corresponden a cada individuo objeto del estudio, salvo el S1 y el S12 que se han descartado por problemas con la colocación correcta de los sensores mientras se realizaban las pruebas.

Los datos contenidos en la base de datos **WESAD** han sido tomados con el objetivo de establecer una relación entre los datos capturados desde los dos dispositivos que forman parte de las pruebas. Estos dispositivos son:

- El **RespiBAN Professional** (*Biosignalsplux | respiBAN Pro, s.f.*), que es un dispositivo profesional que se ajusta al pecho y proporciona los siguientes datos: Electrocardio-

grama (*ECG*), Actividad electrodérmica (*EDA*), Electromiograma (*EMG*), Respiración, Temperatura corporal y Acelerómetro en los 3 ejes.

La frecuencia de muestreo de este dispositivo profesional es de 700Hz, la cual es bastante superior a la de los dispositivos de muñeca como el *Apple Watch* actualmente existentes en el mercado, pero hay que tener en cuenta que es un dispositivo exclusivo de uso médico.

- El **Empatica E4** (*E4 Wristband | Real-Time Physiological Signals | Wearable PPG, EDA, Temperature, Motion Sensors, s.f.*) el cual es un reloj de muñeca orientado a la salud, con multitud de sensores específicos, es utilizado como paradigma en la captura de datos de salud a través de sensores en la muñeca, proporcionando los siguientes datos: Volumen sanguíneo (*BVP, 64 Hz*), Actividad electro dérmica (*EDA, 4 Hz*), Temperatura corporal (4 Hz) y Acelerómetro en los 3 ejes (32Hz).

De los datos disponibles en este conjunto de datos, los más interesantes, para ser relacionados con los sensores disponibles en el (*Apple Watch, s.f.*) son los del Electrocardiograma (*ECG*) del *RespiBAN*, ya que del *ECG* se pueden extraer una serie de parámetros que permiten medir el *estrés*.

Para extraer estos parámetros adicionales del *ECG*, en este trabajo se han utilizado las librerías en *Python* para el análisis de datos de salud **Neurokit** (Makowski et al., 2020). Entre los diferentes parámetros que se pueden obtener, están los datos de la variabilidad del ritmo cardiaco (**HRV**), el cual si que está disponible en la base de datos de salud del (*Apple Watch, s.f.*) y es un parámetro claramente relacionado con el *estrés* (H.-G. Kim et al., 2018).

#### **2.1.2.2. Continuous Stress Detection Using Wearable Sensors in Real Life: Algorithmic Programming Contest Case Study**

Es un estudio (Can et al., 2019) cuyo objetivo principal es obtener unos métodos de detección del *estrés* fuera del entorno médico o del laboratorio, donde tradicionalmente se realizaban, para poder realizar una detección del *estrés* mediante dispositivos de muñeca mucho más económicos y accesibles.

El procedimiento consistió en que durante 9 días, un total de 21 personas formaran parte del estudio para obtener una clasificación basada en 3 niveles de *estrés*, identificados de la siguiente manera:

- **Día libre**, para el nivel de *estrés* más bajo.

- **Conferencia**, para el nivel de estrés mediano.
- **Concurso**, para el nivel de estrés más alto.

En este caso y para lograrlo, se utilizan varios relojes de la marca *Samsung*, concretamente los modelos **Gear S1, S2 y S3** y además el **Empatica E4**.

Basado en estudios anteriores, en el artículo, se justifica la utilización de los sensores expuestos a continuación sobre otros, ya que de los mismos, se obtuvieron buenos resultados para clasificar el estrés:

- **PPG** (photoplethysmography), utilizado para obtener la actividad cardiaca y los intervalos entre picos R y así calcular la variabilidad de la frecuencia cardiaca (*HRV*).
- **EDA** (actividad electro-dérmica), mide la variación en las sudoración y conductancia de la piel que se produce cuando hay situaciones de estrés respecto a cuando no las hay.
- La **temperatura** corporal, no es utilizada en el proceso de extracción de las características pero si para filtrar los datos obtenidos de la actividad electro-dérmica (*EDA*).
- El **acelerómetro** en los 3 ejes, ya que es uno de los sensores más utilizados para la detección de niveles de estrés (García-Ceja et al., 2016).

Para la obtención de las características relacionadas con la actividad cardiaca y el *HRV* se ha usado *MATLAB* junto con la herramienta *HRVTool* (Vollmer, 2019) obteniendo las características mostradas en la [Tabla 3](#):

**Tabla 3.** Características extraídas de la variabilidad del ritmo cardiaco (*HRV*) usando la herramienta *HRVTool* (Vollmer, 2019).

<b>Propiedad</b>	<b>Descripción</b>
<b>Mean RR</b>	Valor medio de los intervalos entre latidos (RR).
<b>STD RR</b>	Desviación estándar de los intervalos entre latidos.
<b>RMSSD</b>	Raíz cuadrada media de la diferencia de los intervalos entre latidos (RR).
<b>pNN50</b>	Porcentaje del número de intervalos entre latidos (RR) sucesivos que varían más de 50ms respecto al intervalo anterior.
<b>HRV triangular index</b>	Número total de intervalos entre latidos (RR) dividido por la altura del histograma de todos los intervalos (RR) medidos en una escala con intervalos de 1/128s.
<b>TINN</b>	Interpolación triangular del histograma de los intervalos entre latidos (RR).
<b>LF</b>	Potencia en la banda baja de frecuencia (0.04–0.15 Hz).

<b>HF</b>	Potencia en la banda alta de frecuencia (0.15–0.4 Hz).
<b>LF/HF</b>	Relación entre la banda baja (LF) y la alta (HF).
<b>pLF</b>	Baja frecuencia de oscilación del ritmo cardiaco.
<b>pHF</b>	Alta frecuencia de oscilación del ritmo cardiaco.
<b>VLF</b>	Potencia en la banda de baja frecuencia (0.00–0.04 Hz).
<b>SDSD</b>	Desviación estándar relacionada con las sucesivas diferencias del intervalo entre latidos (RR).

*Fuente.* Adaptado de (Can et al., 2019) .

La característica obtenida a partir de los datos en bruto del acelerómetro ha sido el valor medio de cada ventana temporal, así como la energía calculada a través de la transformada de Fourier (*FFT*) (Spilsbury y Euceda, 2016).

Además el conjunto de datos en los que se basa el artículo no está balanceado, es decir hay un desequilibrio entre los miembros de las clases de salida y se ha optado en retirar datos de la clase mayoritaria para balancearlos. Hay que reseñar en este punto, que el camino seguido por (Can et al., 2019) es distinto al seguido en este trabajo, en el cual, también se utiliza un conjunto de datos no balanceado, sin embargo, en vez de retirar datos de la clase mayoritaria para obtener un buen ajuste se ha optado por ajustar el umbral óptimo (Zou et al., 2016) (Aprilliant, 2021) y así evitar pérdida de información a la hora de calcular los modelos de aprendizaje.

También se han probado varios modelos de clasificación obteniéndose los mejores resultados con el **MLP** (*Multilayer Perceptron*) con una precisión del **92.15%** con datos obtenidos a través del **Empatica E4** correspondientes al ritmo cardiaco *HR*, *EDA* y *ACC* respecto a los datos de los sensores obtenidos de los **Gear S1..S3**, demostrándose después del análisis de los datos una mayor precisión en general obtenida de la **Empatica**.

### 2.1.2.3. *Stress Detection Using Low Cost Heart Rate Sensors*

El artículo (Salai et al., 2016) presenta los resultados de dos estudios para la detección del estrés donde se ha usado un dispositivo de bajo coste, como es un simple sensor de ritmo cardiaco pectoral, concretamente el *CardioSport TP3 Heart Rate Transmitter* para el análisis de los datos correspondientes a la frecuencia cardiaca.

Como logro, en el primer estudio, los datos obtenidos del *TP3* se han relacionado con gran precisión respecto a los capturados desde un dispositivo de referencia para uso clínico como es el *Schiller MT-101/MT-200 Holter*. Después de realizar un proceso de limpieza y sincronización de los datos se obtuvo una precisión en la correlación **>97 %**.

En el segundo estudio, se ha realizado un test de estrés a 46 voluntarios sanos, 19 son mujeres, y los 27 restantes corresponden a hombres. La edad promedio de los participantes del estudio es de 24.6 años.

Este segundo estudio ha consistido en 10 minutos de relajación mental por medio de escuchar música relajante seguidos de otros 10 minutos de estrés mediante la prueba *Stroop Test* (Renaud y Blondin, 1997) desde una *smartphone* para un total de 20 minutos.

La extracción de las características de la variabilidad de la frecuencia cardiaca (*HRV*), únicamente 3 en el dominio del tiempo, ha sido realizada mediante el software *Kubios* (Tarvainen et al., 2014), siendo las siguientes:

- **Media de la frecuencia cardiaca (*HR*).**
- **pNN50.**
- **RMSSD.**

Como nota destacable del análisis de este artículo, es el bajo número de características extraídas (solo 3) de la *HRV*, respecto a las extraídas, por ejemplo, en el apartado 2.1.2.2, sin embargo, no han sido necesarias más características para obtener unos niveles de precisión en los resultados bastante elevados para clasificar el estrés con una precisión del **75 %** respecto a otros estudios donde se utilizan más características y datos procedentes de otros sensores.

#### **2.1.2.4. *Recent advances in wearable sensors and portable electronics for sleep monitoring***

El estudio (Kwon et al., 2021) trata sobre los avances disponibles en los dispositivos portátiles basados en una electrónica pequeña y compacta con sensores cada vez más sofisticados que permiten hacer un seguimiento de la evolución del sueño y detectar trastornos en el mismo.

El objetivo es tener una visión global de los dispositivos actuales que pueden sustituir en gran medida al dispositivo de referencia en la monitorización del sueño que es el polisomnógrafo (PSG) (Stevens y Clark, 2004), de entre los dispositivos mas destacados se encuentran:

- **Empatica E4.**
- **Galaxy Watch.**
- **Apple Watch.**



## 2.2. Artículos y *datasets* de referencia

### 2.2.1. Analysis of Trigger Factors in Episodic Migraineurs Using a Smartphone Headache Diary Applications

Sobre este artículo (Park et al., 2016) y su base de datos pivota en buena medida este trabajo, ya que es de donde se extrae la información necesaria para clasificar las migrañas incluyendo los disparadores.

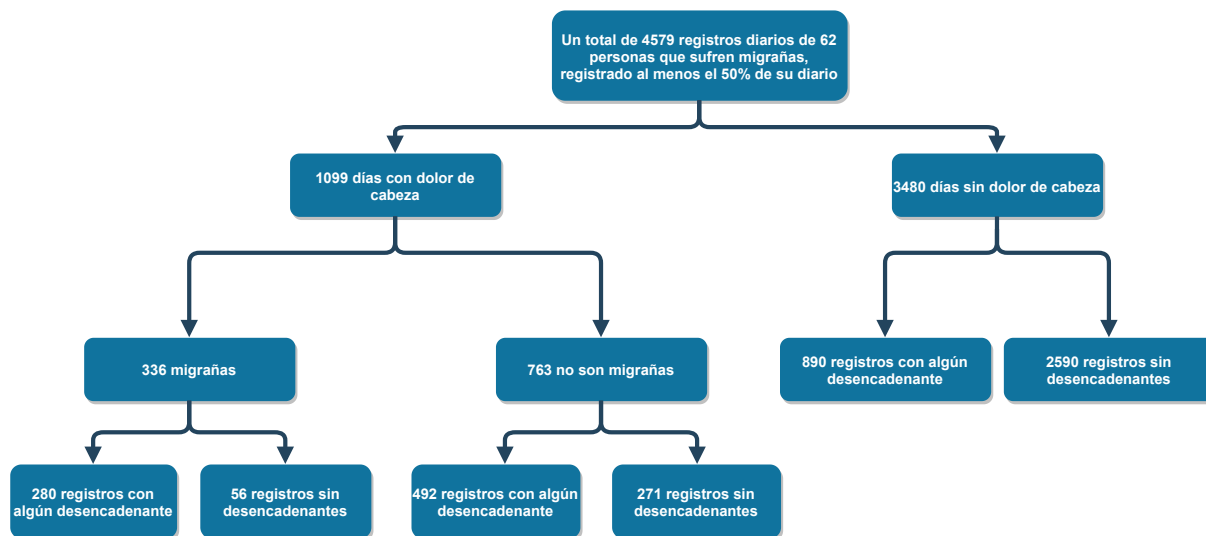
En el artículo, se analizan los episodios de migraña durante 3 meses de 62 pacientes con dicha enfermedad, de los cuales 60 padecían habitualmente migrañas sin aura y los 2 restantes migrañas con aura, los registros diarios eran almacenados en una aplicación de teléfono móvil específica para ello.

La base de datos facilitada en el artículo (Park et al., 2016), consta de 3 tablas:

- **62patients.** Es la tabla resumen donde se describen los 62 pacientes con algún tipo de migraña. Además de una descripción física de los pacientes como la edad, sexo, peso, etc, se detallan los tipos de *desencadenantes* que han sido registrados en las tablas diario.
- **headache diary 1099.** Es la tabla diaria donde solamente se incluyen los registros con algún tipo de dolor de cabeza sea debido a una migraña o no.
- **total diary 4579.** Es la tabla diario total donde se encuentran todos los registros, es decir, días de dolor de cabeza y sin dolor de cabeza registrando los desencadenantes más habituales, que finalmente es la que se ha utilizado como base de conocimiento de la aplicación ya que contiene un amplio registro de varios meses de los pacientes con registros positivos y negativos de algún tipo de cefalea, entre ellas las migrañas.

En la **Figura 4** se expone de manera visual y comprensible la distribución de cefaleas obtenidas de la base de datos:

**Figura 4.** Diagrama de flujo del diario completo de registros de pacientes con migrañas (Park et al., 2016).



Fuente. Adaptado de (Park et al., 2016).

Como puede verse en la **Figura 5**, se expone un fragmento, con las cabeceras traducidas, de la tabla original de (Park et al., 2016) que se ha utilizado como conjunto de datos para la base de conocimiento del modelo de aprendizaje de las *migrañas*.

**Figura 5.** Fragmento de la tabla original (Park et al., 2016) total diary 4579 en formato Excel.

유발요인	내인의 요인										외부의 요인							기타	
유발요인 수	스트레스	수면과다	수면부족	운동	운동안하기	육체적피로	생리주기: 월경기	생리주기: 배란기	과도한 감정변화	합계	날씨/온도 변화	과도한 햇빛	소음	부적절한 조명	특정 냄새(향수, 화장품 등)	합계	과도한 음주	불규칙한 식사(공복 등)	과식
Triggers	In Human Factors										External factors							Etc	
Factors can cause	Stress	Sleeping too much	Lack of sleep	Exercise	Not to Exercise	Physical fatigue	Menstruation: menstruation	Menstrual cycle: ovulation	Excessive emotional changes	Sum	Weather / temperature changes	Excessive sunlight	Noise	Inappropriate lighting	Certain smells (perfume cosmetics, etc)	Sum	Excessive drinking	Irregular eating (fasting, etc.)	Surfeit (ex extra, gl overage)
N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fuente. Adaptado de (Park et al., 2016).

De la tabla representada en la **Figura 5**, se extraen los disparadores más habituales como causantes de migrañas, como son:

El **estrés** (36%), **carencia de sueño** (24,4%), **exceso de sueño**, **ejercicio**, **fatiga** (22,3%), **hormonales** (18,5%), **emocionales**, **cambios en el tiempo**, **luz**, **ruido**, **olores**, **comidas irregulares**, **caféina**, **alcohol**, **comer (chocolate / queso)**...

En la tabla “*Comparison of Triggers between Migraines and Non-migraine headaches on the Smartphone Application-based Electronic Headache Diary.*” del estudio de (Park et al., 2016) ha quedado palpable que los 2 desencadenantes seleccionados para este trabajo como el estrés y la carencia de sueño son los más importantes para las cefaleas que corresponden con las migrañas dentro la población que fue seleccionada en ese estudio, población que ha sido descrita al comienzo de este apartado.

Para explicar porqué son estos los desencadenantes naturales más habituales para las migrañas se han consultado los siguiente artículos mostrados en la [Tabla 4](#):

**Tabla 4.** *Artículos de referencia de los desencadenantes naturales de las migrañas.*

<b><i>Triggers, Protectors, and Predictors in Episodic Migraine</i></b>	
Referencia	(Marmura, 2018)
Desencadenantes Naturales	El estrés, los cambios menstruales, de tiempo, desordenes del sueño, alcohol, ayuno, baja presión barométrica, determinados alimentos.
Enfoque	Se realiza un análisis punto por punto de los desencadenantes naturales, justificando su inclusión como desencadenantes principales con numerosas citas a artículos científicos. Se indica la dificultad a la hora de evitar dichos desencadenantes y además se debe tener en cuenta que los síntomas premonitorios de la migraña como son el dolor de cuello, la fatiga y la sensibilidad a las luces u olores pueden ser confundidos en incluso imitar a los desencadenantes naturales.
<b><i>The Triggers or Precipitants of the Acute Migraine Attack</i></b>	
Referencia	(Kelman, 2007)
Desencadenantes Naturales	Estrés, hormonales en la mujer, ayuno, desordenes del sueño, cambios del tiempo, determinados olores, alcohol, fumar, dormir tarde, ejercicio excesivo, actividad sexual.

Definir los factores desencadenantes más habituales. Para ello, se han analizado los datos obtenidos de 1207 pacientes con diferentes trastornos de dolor de cabeza, entre ellos la migrañas.

Donde un **75.9 %** del total reportaron algún tipo de desencadenante. La frecuencia de los desencadenantes obtenida de este estudio (Kelman, 2007) es la siguiente:

#### Enfoque

- Estrés 79.7 %
- Hormonales 65.1 %
- No comer 57.3 %
- Cambios de tiempo 53.2 %
- Desordenes del sueño 49.8 %
- Perfumes u olores 43.7 %
- Dolor de cuello 38.4 %
- Luces 38.1 %
- Bebidas alcohólicas 37.8 %
- Fumar 35.7 %
- Dormir muy tarde 32 %
- Calor climático 30.3 %
- Comidas específicas 26.9 %
- Ejercicio físico 22.1 %
- Sexo 5.2 %

*Fuente.* Adaptado de (Marmura, 2018) y (Kelman, 2007).

## 2.3. Conclusiones

De este estudio del arte, se han extraído una serie de conclusiones, como son:

- La posibilidad de crear un modelo basado en una base de datos de personas que haya sufrido migrañas para diagnosticar la migraña sin aura, la más común, siguiendo los criterios de clasificación de *1.1 Migraine without aura* del artículo (“Headache Classification Committee of the International Headache Society (IHS) The International Classification of Headache Disorders, 3rd Edition”, 2018).
- Que la utilización de dicho modelo sea accesible mediante una aplicación que permita guardar diagnósticos incluyendo además en el modelo los desencadenantes más habituales.
- Determinar mediante la captura de los datos de la variabilidad de la frecuencia cardiaca (*HRV*) obtenidos directamente a través de la aplicación **Respirar** del *Apple Watch* si una persona puede estar más o menos expuesta a una situación de estrés que provoque un ataque de migraña.
- A través de la aplicación de salud **AutoSleep** (*AutoSleep. Sleep Tracking*, s.f.) del *Apple Watch* se pueden obtener los datos del sueño y determinar si al ausencia del sueño <6 horas puede ser un desencadenante de la migraña de una manera objetiva.

## Capítulo 3

# Objetivos

En este capítulo se describen unos objetivos a priori, para cuyo logro se ha plasmado este documento.

### 3.1. Objetivos Generales

A continuación se describe el objetivo fundamental sobre el que pivota este trabajo:

- **Determinar** mediante un auto diagnóstico predictivo basado en el análisis y captura de datos en una aplicación si una persona ha sufrido una afección o no.

La afección a diagnosticar es la **migraña sin aura**, la cual tiene mucha prevalencia en la sociedad actual, y además es difícil de detectar ya que se confunde fácilmente con otro tipo de cefaleas.

Para la predicción, la aplicación incluye en la base de datos no solamente los criterios ya establecidos en *1.1 Migraine without aura* sino que además se trata de **identificar** cuáles de los desencadenantes más habituales han podido influir en la aparición de la migraña.

Además, teniendo en cuenta que 2 de los desencadenantes de la migraña con mayor prevalencia son el **estrés** y la **carencia de sueño** se pretende determinar de manera objetiva, si los mismos han podido desencadenar la migraña o no. Este hito se lleva a cabo mediante la captura y procesado de los datos descargados de un reloj comercial de muñeca como el *Apple Watch*, pudiendo permitir a la persona que sufre las migrañas determinar que tipo de medidas tomar relacionadas con estos dos desencadenantes, como pueden ser las técnicas de relajación, descanso, etc para así reducir los episodios de migraña evitando en lo posible la frustración que aparece cuando estas medidas no son efectivas por un diagnóstico fallido.

## 3.2. Objetivos Específicos

Teniendo como base el objetivo fundamental planteado en 3.1, se plantean los siguientes objetivos más específicos:

- **Establecer** una base de conocimiento que permita realizar el diagnóstico de la *migraña sin aura* mediante el análisis y preparación de los datos disponibles en diferentes fuentes.
- **Diseñar** un sistema de diagnóstico predictivo preciso basado en técnicas de *aprendizaje automático e inteligencia artificial*, para así poder **identificar** de una manera específica para cada usuario, qué desencadenantes están provocando las migrañas y cuáles son los de mayor prevalencia.
- **Establecer** unas conclusiones que permitan ampliar el desarrollo de la aplicación y otras posibles vías de investigación.

## Capítulo 4

# Metodología de trabajo

La metodología utilizada en este trabajo, permite describir de manera pormenorizada como han sido analizados los datos de las personas que formaron parte del estudio (Park et al., 2016) así como las fases desarrolladas para crear la aplicación que ha permitido lograr los **Objetivos** planteados.

Estas fases son descritas a continuación de manera concisa y desarrolladas en las siguientes secciones de este capítulo:

1. Extracción y limpieza de los datos disponibles de los participantes en (Park et al., 2016) que ya han sido descritos en **2 Estado del Arte**. La herramienta utilizada en esta fase ha sido la hoja de cálculo *Google Sheets* ya que es gratuita y permite aplicar macros para las traducciones de los textos en Koreano, y sobre todo exportar fácilmente la hoja de cálculo con los datos del formato Excel al formato de valores separados por comas CSV.
2. Selección del modelo de aprendizaje, para ello se ha utilizado la interfaz de programación de aplicaciones (API) *Scikit-learn* (Pedregosa et al., 2011) que contiene una amplia variedad de algoritmos específicos para el aprendizaje automático en el lenguaje *Python*.

Esta fase conlleva una serie de procedimientos para determinar si el modelo seleccionado, en este caso el modelo basado en un método de **regresión logística multinomial** es el más adecuado y como puede optimizarse, para lo cual se han seguido los siguientes procedimientos:

- a) Carga del conjunto de datos, en este procedimiento, se han eliminado las características no utilizadas en el modelo de aprendizaje para obtener la matriz de entrada de datos  $X$  así como el vector de salida  $y$  con las tres clases objetivo, como son *No Migraña*, *Posible Migraña* y *Migraña*.



- b) Descripción gráfica de la correlación de las características del conjunto de datos, así como de la distribución no balanceada del mismo.
  - c) Análisis de la efectividad del modelo de regresión logística multinomial por medio de diferentes procedimientos de obtención de datos para el entrenamiento y test que permiten la evaluación de las métricas utilizadas para descartar la existencia de sobre-muestreo y así determinar si el modelo elegido es efectivo o no.
3. Una vez que se ha verificado que el modelo es adecuado, se trata de mejorar su eficacia mediante la implantación de diferentes métodos de optimización hasta obtener el más óptimo de acuerdo a la métrica seleccionada, en este caso la precisión.

Los diferentes métodos utilizados para obtener el modelo de regresión logística más efectivo, así como los resultados obtenidos han sido realizados en un entorno de programación basado en un cuaderno de *jupyter*, donde se ha creado el cuaderno ***ADMTF ML Models for Diagnosing Migraine Situations (DMS).ipynb*** cuyo desarrollo y contenido se puede consultar en el capítulo [A Tablas de datos y Notebooks](#).

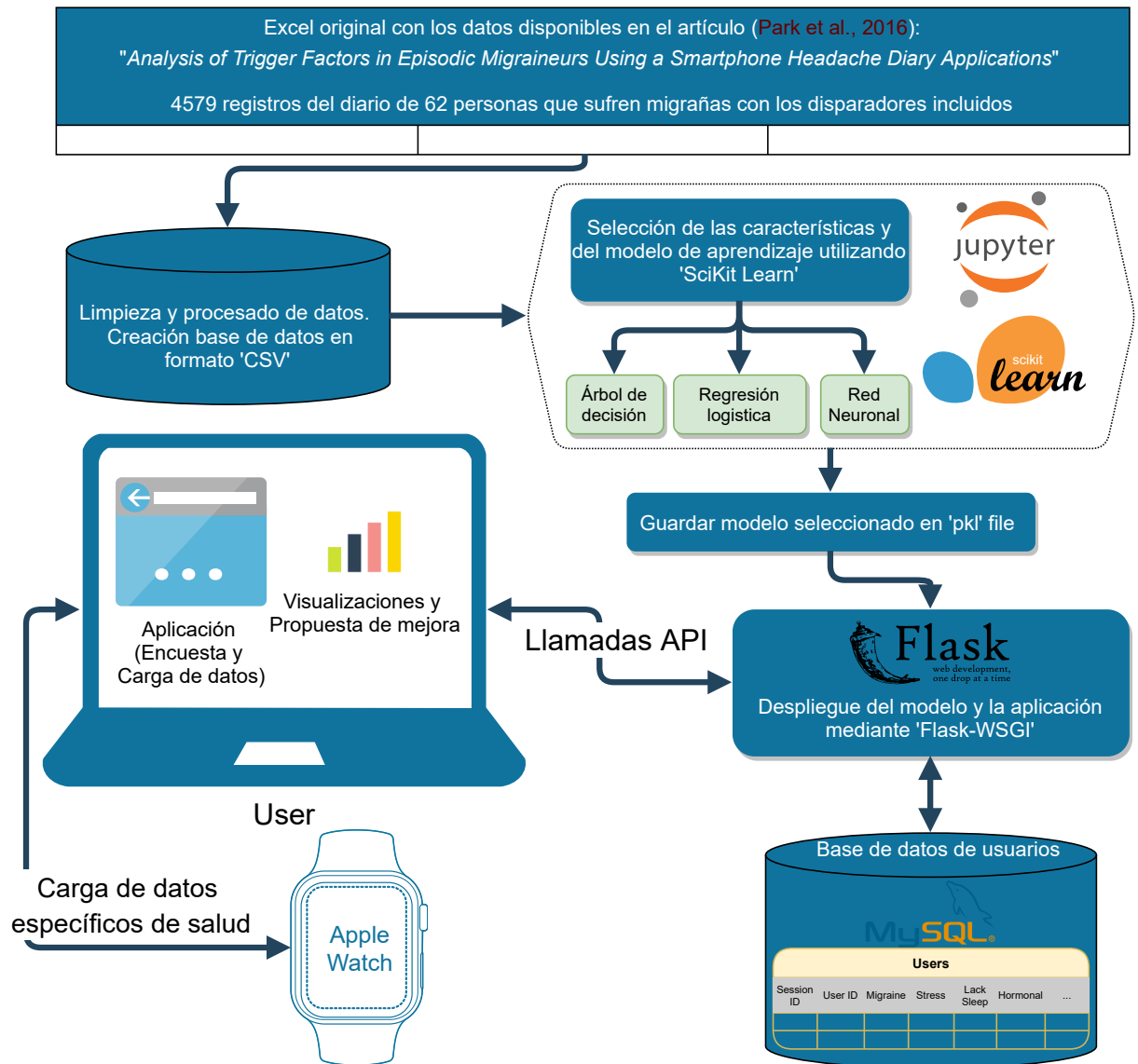
Una vez obtenido el modelo de aprendizaje, este ha sido empaquetado, generando el fichero ***pkl*** para que pueda ser utilizado en la aplicación y así mediante su aplicación poder determinar si se ha producido una migraña o no incluyendo además los desencadenantes más habituales.

4. Implementación del modelo de aprendizaje de la aplicación, para lo cual se ha utilizado ***Flask (Pallets/Flask, 2021)***, el cual es un marco ligero para generar aplicaciones web en lenguaje *Python* de manera rápida y limpia, está basado en una interfaz de puerta de enlace del servidor web (*WSGI*).
5. Procesar los datos de salud del *Apple Watch* en la aplicación web para poder responder de manera objetiva, si tanto el *estrés* como la *carencia de sueño* han podido provocar la migraña o no.
6. Diagnóstico de la *migraña sin aura*, mediante una encuesta que se va realizando al usuario, se complementan las entradas necesarias al modelo para que la aplicación determine si se ha producido una *migraña sin aura*, si es una *posible migraña* o por el contrario no se corresponde con una migraña sino que es otro tipo de cefalea.
7. Los resultados de la predicción se almacenan en una base de datos *MySQL* para que cada usuario pueda consultar los resultados y determinar que desencadenantes son más comunes en las migrañas de una manera gráfica, para que con el tiempo permitan

obtener algún tipo de relación entre algunos desencadenantes típicos y el usuario en concreto.

El diagrama de la Figura 6 describe la metodología seguida en este trabajo:

**Figura 6.** Diagrama general de la metodología de trabajo.



*Nota.* Este diagrama permite visualizar de manera gráfica la metodología seguida para lograr los objetivos.

### 4.1. Extracción y limpieza de la base de datos de las migrañas

La extracción y limpieza de los datos sobre los que se ha desarrollado el modelo de aprendizaje de las *migrañas sin aura* es una parte fundamental del documento ya que las tablas de origen tienen datos que si han sido utilizados pero otros no. Además siempre hay algún dato erróneo o que no aporta información al modelo que ha tenido que ser retirado.

De las 3 tablas originales descritas en el [2 Estado del Arte](#) que conforman las base de datos utilizada en este estudio (Park et al., 2016) se ha seleccionado la tabla **total diary 4579** para extraer las características necesarias y así formar la base de conocimiento del modelo elegido.

En el proceso correspondiente a la limpieza y selección de las características se ha utilizado el programa *Google Sheets* porque a pesar de que las tablas originales están en Excel, *Google Sheets* facilita las traducciones de las cabeceras mediante macros, por ejemplo la macro:

GOOGLETRANSLATE (T2, "ko", "en"), la cual traduce del Koreano al Ingles la casilla T2.

Por otra parte, se ha eliminado un registro de datos de los 4579 originales porque no contenía información relevante, además, muchas de las cabeceras referentes a medicamentos, tratamientos recibidos, hospitalizaciones, etc, por parte de las 62 personas que participaron en el estudio durante 3 meses han sido retirados por no aportar información relevante al modelo utilizado. Como referencia de la tabla utilizada para obtener el modelo, a continuación en la [Figura 7](#) se muestra un fragmento de tabla obtenida **totalDiary4578Filtered**:

**Figura 7. Fragmento de la tabla totalDiary4578Filtered.**

registration_n umber	gender	age	headache_fre q_days_mont h	duration_hou rs_times	ICHD_3_C_4_ pain_in_move ment	pain_degree	ICHD_3_C_3_ pain_intensit y	pain_area_left head	pain_area_righ t_head	pain_area_mid dle	pain_area_full	pain_area_aro und_eyes	pain_area_aro und_neck	ICHD_3_C_1_p ain_area_unila teral	pain_area_bila teral	pain_shape_th robbing	pain_shape_b ghtening	pain_shape_r ngling	pain_shape_h cavy
cmc-0003	Female	20	8	24	1 middle	1	0	0	0	0	1	1	0	0	0	1	0	1	0
dha-0014	Female	36	8	12	0 middle	1	0	0	0	0	1	1	0	0	0	0	1	1	0
cmc-0022	Female	40	5	24	0 middle	1	1	0	0	0	0	0	0	1	0	1	0	0	0
cmc-0022	Female	40	5	24	1 middle	1	1	0	0	0	0	0	0	1	0	1	0	0	0
cmc-0022	Female	40	5	24	0 middle	1	1	0	0	0	0	0	0	1	0	1	0	0	0
cmc-0025	Female	39	15	7	0 weakness	0	0	0	0	1	0	0	0	0	0	0	0	1	0
cmc-0027	Female	41	15	24	0 weakness	0	0	0	1	0	0	0	0	0	0	0	0	0	1
cmc-0027	Female	41	15	24	0 weakness	0	1	0	0	0	0	0	0	1	0	0	0	0	1
cmc-0027	Female	41	15	24	0 weakness	0	1	0	0	0	0	0	0	1	0	0	0	0	1
cmc-0028	Female	31	8	24	1 weakness	0	1	0	0	0	0	0	0	1	0	1	0	0	0
cmc-0028	Female	31	8	24	0 weakness	0	0	0	0	1	0	0	0	0	0	0	0	1	0
cmc-0033	Male	23	4	48	0 weakness	0	1	0	0	0	0	0	0	1	0	0	1	0	0
cmc-0033	Male	23	4	48	0 weakness	0	1	0	0	0	0	0	0	1	0	0	0	1	0
cmc-0033	Male	23	4	48	0 weakness	0	1	0	0	0	0	0	0	1	0	0	1	0	0
cmc-0064	Female	48	15	4	0 weakness	0	0	0	1	0	0	0	0	0	0	1	0	0	0
dha-0009	Male	33	7	12	0 weakness	0	0	1	0	0	0	0	0	1	0	0	0	0	1
dha-0009	Male	33	7	12	0 weakness	0	0	1	0	0	0	0	0	1	0	0	0	1	0
dha-0010	Female	37	6	6	1 middle	1	0	1	0	0	1	1	1	0	1	0	0	0	1
dha-0010	Female	37	6	6	0 weakness	0	0	1	0	0	0	1	1	0	1	0	0	0	1
dha-0010	Female	37	6	6	0 weakness	0	0	0	0	1	0	0	0	0	0	0	0	0	1

**Nota.** La tabla contiene todas las características necesarias para el modelo de aprendizaje, conteniendo un total de 4578 registros diarios a lo largo de 3 meses de las 62 personas que se sometieron al estudio.

## 4.2. Selección del modelo de aprendizaje

Para la selección del modelo de aprendizaje más adecuado teniendo en cuenta la base de datos obtenida, se han utilizado algunas de las librerías más conocidas para el aprendizaje automático en lenguaje *python* utilizando **Jupyter** de la plataforma *Anaconda-Navigator* para desarrollar el cuaderno **ADMTF ML Models for Diagnosing Migraine Situations (DMS).ipynb**, el cual, contiene todos los cálculos y desarrollos realizados para la selección del modelo de aprendizaje.

Entre las librerías habituales para la manipulación de datos en *Python* como *Pandas* (*Pandas - Python Data Analysis Library, s.f.*), *Numpy* (*NumPy, s.f.*), etc, la librería fundamental utilizada para crear el modelo de aprendizaje seleccionado ha sido **Scikit-learn**, (Pedregosa et al., 2011) la cual integra una gama de algoritmos de aprendizaje supervisado y no supervisado muy adecuados para un rango de datos intermedio como es la base de datos utilizada en este trabajo que ya ha sido descrita en 2 Estado del Arte y en 4.1 Extracción y limpieza de la base de datos de las migrañas.

A continuación se detallan las diferentes fases que permiten la obtención del modelo de aprendizaje adecuado para los datos que se han utilizado.

### 4.2.1. Carga del conjunto de datos

Utilizando la librería de *Pandas* (*Pandas - Python Data Analysis Library, s.f.*) se cargan los datos obtenidos del proceso anteriormente descrito en el apartado anterior, los cuales se encuentran en el fichero **totalDiary4578Filtered.csv**.

El objetivo es obtener un marco de datos para estructurarlos y así poder trabajar con las librerías de *Python* de manera más sencilla. Del marco de datos obtenido, se seleccionan las columnas que van a ser utilizadas en la creación de los modelos eliminando las columnas que no van a ser utilizadas respecto al fichero original como puede verse en el fragmento de código Listado 4.1.

#### Fragmento de código 4.1. Carga del *dataset* de las migrañas en el cuaderno de *Jupyter*.

```

1
2 attributes_selected = ['gender', 'age', 'ICHD_3_B_duration_criteria', 'ICHD_3_C_1_pain_area_unilateral', '
    ICHD_3_C_2_pain_shape_pulsatile', 'ICHD_3_C_3_pain_intensity', 'ICHD_3_C_4_pain_in_movement', '
    ICHD_3_D_1_symptoms_nausea_vomiting', 'ICHD_3_D_2_symptoms_photo_phono_phobia', 'trigger_stress',
    trigger_excessive_sleep', 'trigger_lack_sleep', 'trigger_exercise', 'trigger_not_exercise', 'trigger_fatigue', '
    trigger_hormonal_changes', 'trigger_emotional_changes', 'trigger_weather_changes', 'trigger_excessive_sunlight', '
    trigger_inappropriate_lighting', 'trigger_noise', 'trigger_smells', 'trigger_irregular_eating', 'trigger_overeating'
    , 'trigger_caffeine', 'trigger_smoking', 'trigger_drinking_alcohol', 'trigger_chocolate_cheese', 'trigger_traveling'
    , 'trigger_other', 'ICHD_3_migraine', 'ICHD_3_probable_migraine', 'ICHD_3_no_migraine', 'ICHD_3_migraine_classes']
3
4 # Read the dataset in format comma-separated values (csv) into a dataframe

```

```
5 dframe = pd.read_csv('./DMS_datasets/totalDiary4578Filtered.csv', skipinitialspace=True, usecols=attributes_selected)
6
7 # Removing uninteresting headers first
8 dframe = dframe.drop(columns=['gender', 'age', 'trigger_other'])
9
10 # Splitting dataset into classes and features
11 # y['ICHD_3_migraine_classes'] -> (2, ICHD_3_migraine) (1, ICHD_3_probable_migraine) and (0, ICHD_3_no_migraine)
12 y = dframe.filter(['ICHD_3_migraine_classes'])
13 # features are all columns except classes
14 X = dframe.iloc[ : , :(len(dframe.columns) - 4)]
```

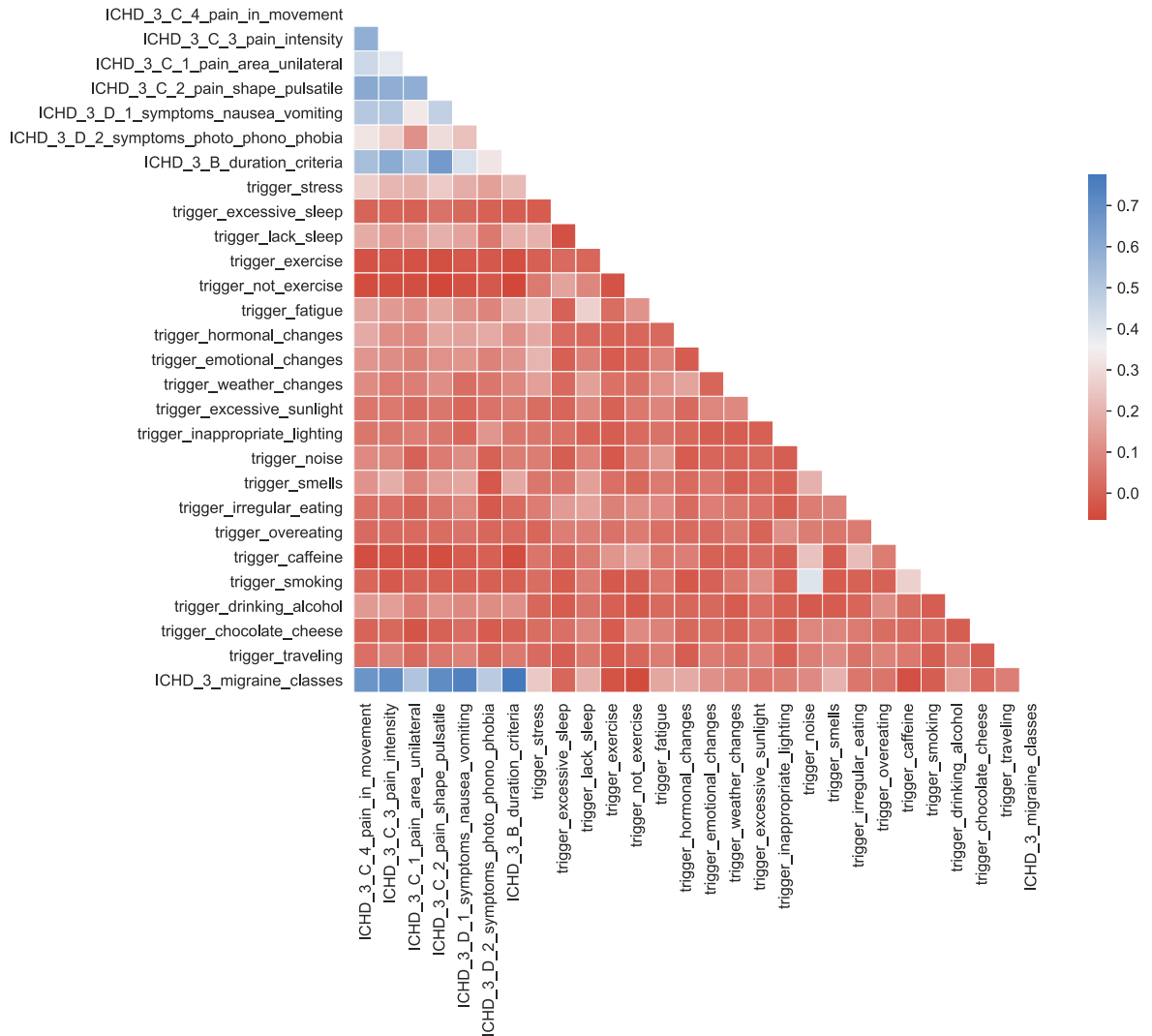
Creando el marco de datos con las características de entrada **X** y el de las clases buscadas **y** (*ICHD\_3\_migraine*, *ICHD\_3\_probable\_migraine*, *ICHD\_3\_no\_migraine*) las cuales han sido agrupadas en la columna **ICHD\_3\_migraine\_classes** y puede tener 3 valores:

- **2** para *ICHD\_3\_migraine*.
- **1** para *ICHD\_3\_probable\_migraine*.
- **0** para *ICHD\_3\_no\_migraine*.

### 4.2.2. Correlación entre las características

La siguiente visualización de la [Figura 8](#) permite de manera intuitiva y rápida determinar qué características están más relacionadas entre sí y por lo tanto tienen mayor importancia en la obtención del modelo predictivo.

**Figura 8.** Correlación entre las características del conjunto de datos.

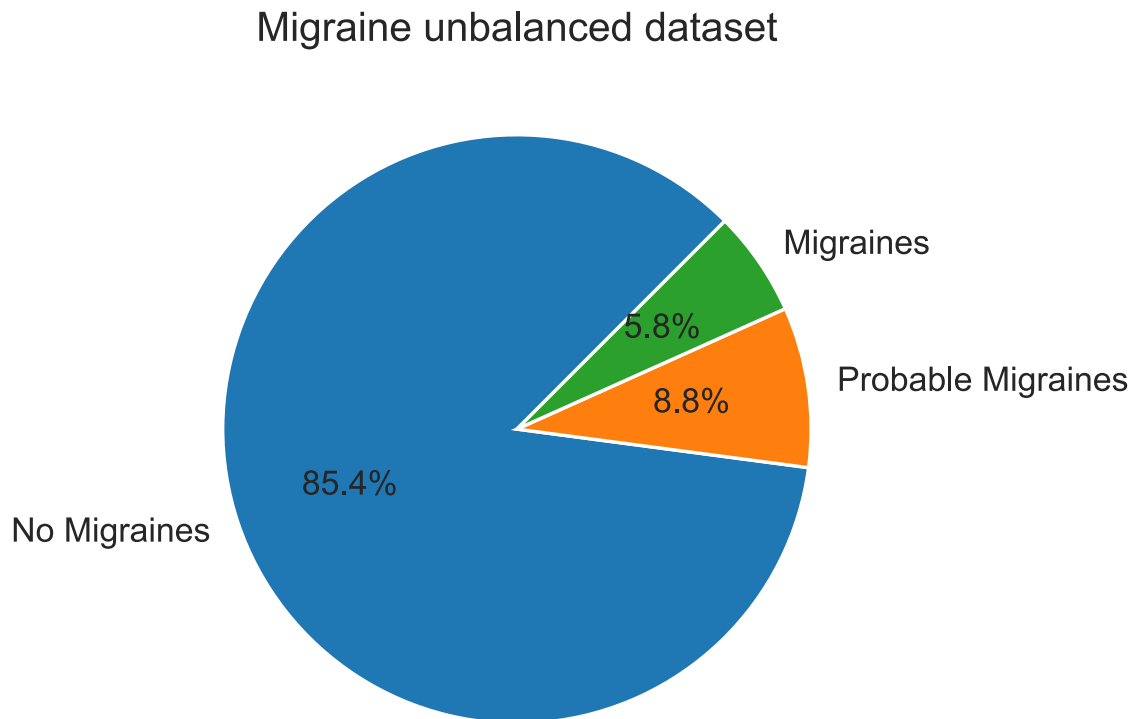


*Nota.* La correlación entre las características con los desencadenantes son difíciles de relacionar, pero si se puede ver como hay alguna relación evidente entre algunas como la *fatiga* y la *carencia de sueño*.

#### 4.2.3. Conjunto de datos no balanceado

En este caso el objetivo buscado es predecir las clases *Migraña* o *Posible migraña*, las cuales corresponden a las clases minoritarias del conjunto de datos no balanceado con un ratio del **5.8 %** y del **8.8 %** respectivamente, frente al **85.4 %** de la clase mayoritaria *No migraña* como se puede observar en la [Figura 9](#):

**Figura 9.** Distribución de las clases en el dataset.



*Nota.* El conjunto de datos es claramente **no balanceado** con lo que a la hora de obtener el modelo hay que tenerlo en consideración.

#### 4.2.4. Utilizando el modelo de regresión logística *multinomial*

Este modelo de aprendizaje es conveniente cuando se dispone de muchas variables independientes de entrada con valores binarios (0/1) y una o varias variables dependientes de salida como es el caso de este estudio, donde hay **27** variables de entrada con posibles valores (1/0) y **3** clases de salida, *no migraña*, *posible migraña* y *migraña*. La posibilidad de utilizar otros algoritmos como las *redes neuronales*, es una posibilidad a futuro, como posible mejora, aunque la elevada precisión obtenida con el modelo de regresión logística haciendo las modificaciones oportunas y sus menores requerimientos de cálculo así como menor riesgo de sobre ajuste (Tu, 1996) hacen del modelo de regresión una buena elección para obtener una buena precisión para calcular la probabilidad de que se produzca la clase buscada. Al tratarse de múltiples clases buscadas, se ha utilizado de la librería (Pedregosa et al., 2011) el modelo de regresión logística siguiente Listado 4.2, en el cual, se muestra la instancia del modelo creado y sus parámetros correspondientes.

##### Fragmento de código 4.2. Instancia del modelo elegido de regresión logística multinomial.

```

1 #####
2 # Multinomial logistic regression model instance
3 #####
4 # In this case, the 'random_state' passed value will have an effect on the reproducibility of the results
5 # returned by the function (fit, split, or any other function like k_means)
6 # 'lbfgs' solver support only L2 regularization
7 # 'multinomial' uses the cross-entropy loss for multiclass
8 LRRegression_Multi_Model = LogisticRegression(multi_class='multinomial', solver='lbfgs', random_state = 0)

```

Los parámetros pasados definen las características del modelo de regresión logística y se detallan a continuación:

- **multiclass='multinomial'**, donde el parámetro *multinomial*, que es una extensión de la regresión logística binaria que permite más de dos categorías de la variable dependiente como es el caso de este conjunto de datos en el que hay 3. Al igual que la regresión logística binaria, la regresión logística *multinomial* utiliza la estimación de máxima verosimilitud para evaluar la probabilidad de pertenencia categórica.
- **solver='lbfgs'**, que representa el algoritmo utilizado para el proceso de optimización utilizado por la librería *scikit-learn* (Pedregosa et al., 2011), que en este caso es *LBFSGS* (Yuan et al., 2011), el cual es un algoritmo *BFGS* (Fletcher, 1987) de memoria limitada para resolver ecuaciones simétricas no lineales.
- **random\_state=0**, este parámetro solo se utiliza para mantener la reproducibilidad de los resultados.



#### 4.2.5. Métricas utilizadas para evaluar el rendimiento del modelo

En todas las métricas utilizadas, el valor óptimo es 1 y el más bajo es 0, y para el cálculo de las mismas ha sido utilizada la librería (Pedregosa et al., 2011). Las utilizadas para evaluar el rendimiento del modelo son:

- La métrica **precision\_score 4.1**, que es la habilidad del clasificador de no clasificar como positiva una muestra que en realidad es negativa. Es una media útil cuando el coste debido a los falsos positivos es elevado.

$$precision = \frac{t_{positive}}{t_{positive} + f_{positive}} \quad (4.1)$$

- La métrica **recall\_score 4.2**, que es la habilidad del clasificador de encontrar todas la muestras positivas. Es importante cuando el coste de los falsos negativos es elevado.

$$recall = \frac{t_{positive}}{t_{positive} + f_{negative}} \quad (4.2)$$

- La métrica **f1\_score 4.3**, que es el media armónica de las 2 métricas anteriores, precisión y recuerdo. El *f1* para el conjunto de datos no balanceados, como es el caso, es una media conveniente ya que penaliza los valores extremos, es decir, puede haber un valor elevado de *precisión* pero si el valor de *recuerdo* es bajo, el valor de *f1* será muy bajo y viceversa.

$$f1 = \frac{2 * precision * recall}{precision + recall} \quad (4.3)$$

- La métrica **accuracy\_score 4.4**, que calcula la exactitud del modelo, es decir, el porcentaje de las clases que han sido pronosticadas correctamente respecto al total de las predicciones, sean correctas o no.

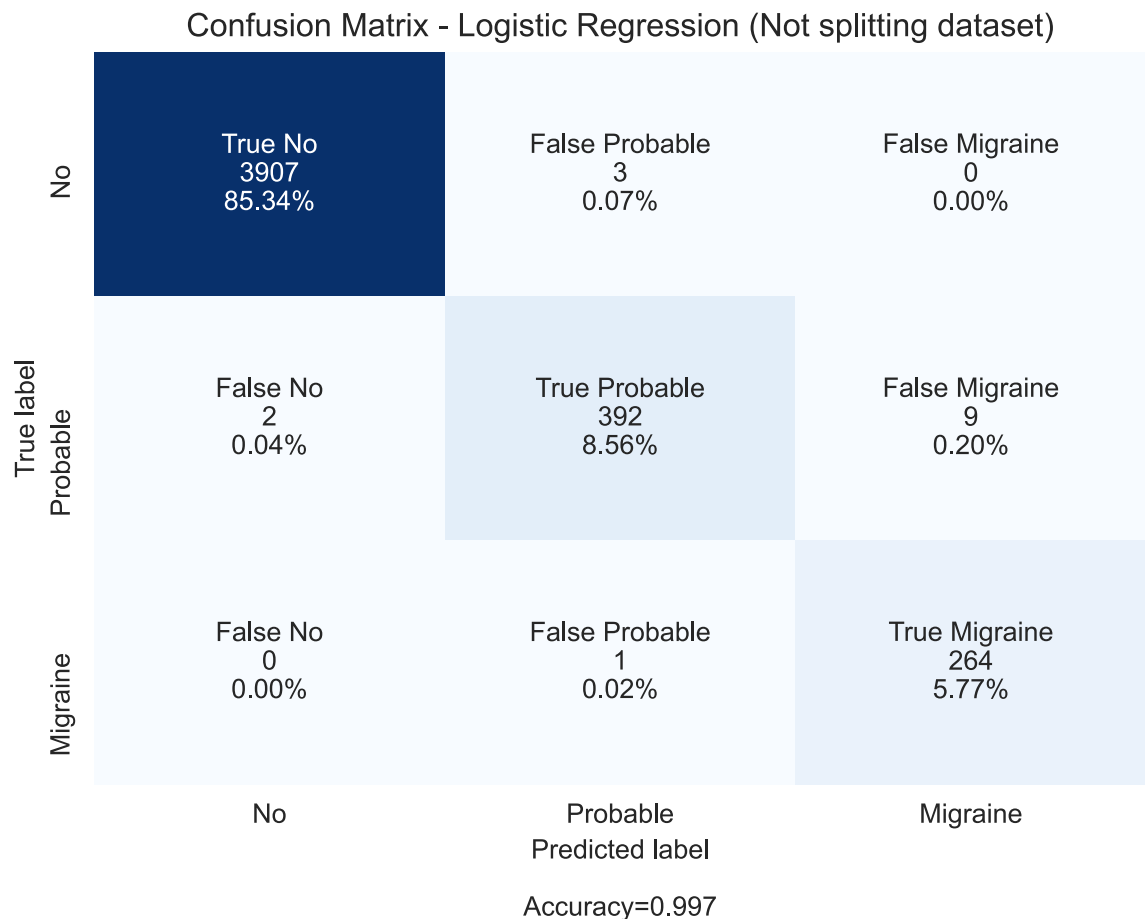
$$accuracy = \frac{t_{positive} + t_{negative}}{t_{positive} + f_{positive} + t_{negative} + f_{negative}} \quad (4.4)$$

#### 4.2.6. Todos los datos como datos de entrenamiento (Primer planteamiento)

Utilizando todos los datos disponibles como datos de entrenamiento, con el modelo de regresión logística *multinomial* 4.2.4 utilizado, se obtiene una precisión del **99.67%** como se puede ver en la siguiente [Figura 10](#), la cual es un buen valor de precisión, así como son buenas el resto de métricas obtenidas.

**Figura 10.** Matriz de confusión - Regresión Logística Multinomial - Sin separar el conjunto de datos.

```
Metric score format: [NMig% PMig% Mig%]
Precision score:     [99.95% 98.99% 96.70%]
Recall score:       [99.92% 97.27% 99.62%]
F1 score:           [99.94% 98.12% 98.14%]
Accuracy:           99.67%
```



*Nota.* Utilizando todo el conjunto de datos aparentemente las métricas son elevadas pero con un riesgo de sobre ajuste que hay que descartar.

Aunque las métricas obtenidas son muy elevadas hay que descartar que sea una medida engañosa, ya que puede ocurrir alguna de las siguientes situaciones cuando se utiliza todo el conjunto de datos para entrenar el modelo:

- Utilizando todos los datos como datos de entrenamiento se corre el riesgo bastante probable de tener sobre ajuste, es decir, que el modelo se ajuste perfectamente a los datos de entrenamiento pero cuando es probado con otros datos de prueba nuevos o con datos reales no se ajusta correctamente.
- En un conjunto de datos no balanceado, como es el caso, el modelo de aprendizaje corre el riesgo de predecir sobre todo la clase mayoritaria, en este caso concreto, la clase mayoritaria corresponde a los días con ausencia de migraña. Por lo tanto, hay que descartar que el modelo obtenido aplicando el clasificador de regresión logística no sea adecuado, y una forma inmediata de hacerlo, es dividir los datos, en datos de prueba y datos de entrenamiento.

Cuando todos los datos procedentes del conjunto de datos se usan como datos de entrenamiento, como en el apartado anterior, se puede obtener una precisión en el modelo muy elevada **99,67 %** que puede ser engañosa debido a un ajuste excesivo que provoque que el modelo siga a la perfección o casi todas las muestras de entrenamiento, y sin embargo, no se ajuste bien a nuevos datos, que realmente es el objetivo buscado.

#### **4.2.7. División de los datos, (20 %) de prueba y el resto (80 %) de entrenamiento**

Un punto de partida para comprobar si hay un problema de sobre ajuste o no, es dividir el conjunto de datos en dos partes, una de prueba y otra de entrenamiento:

- Datos de prueba, es decir, datos nuevos, para ver si el modelo se ajusta correctamente, un porcentaje de un 20 % suele ser una buena elección como punto de partida.
- El resto, como datos de entrenamiento del modelo, en este caso el 80 %.

El método de la librería *scikit-learn* (Pedregosa et al., 2011) utilizado es *train\_test\_split*, el cual realiza de manera automática una división de los datos, con el parámetro *test\_size=0.2*.

Además destacar como puede verse en la fracción de código siguiente [Listado 4.3](#) extraída del cuaderno de *Jupyter* que el parámetro *stratify=true* permite dividir los datos de manera estratificada, es decir distribuida a lo largo del conjunto de datos de manera fraccionada. El parámetro *random\_state=0* únicamente hace que el método sea repetible en múltiples ejecuciones posteriores para poder obtener unos resultados que puedan ser comparados, de

no ser así la partición de los datos cada vez que se llama al método es distinta y las métricas varían en consecuencia.

**Fragmento de código 4.3.** Método de división simple de los datos 20 % para test y el resto de entrenamiento.

```

1 # splitting 20% (test data), 80% (training data) and stratify mode:
2 # https://scikit-learn.org/stable/modules/cross_validation.html#stratification
3 # with 'random_state=0' everytime the result will be the same
4 XTrain, XTest, yTrain, yTest = train_test_split(X.values, y.values.ravel(), test_size = 0.2, stratify=y, random_state=0)

```

Las métricas obtenidas, así como la matriz de confusión de la [Figura 11](#) siguen siendo buenas a pesar de que se están utilizando solo un 80 % de los datos para entrenar el modelo, y el resto, el 20 %, corresponden a datos nuevos con los que probar el modelo.

**Figura 11.** Matriz de confusión - Regresión Logística Multinomial - 80 % de entrenamiento y 20 % de test.

```

Metric score format:      [NMig%  PMig%  Mig%]
Precision score - 80|20:  [99.87% 97.50% 96.23%]
Recall score - 80|20:    [100.00% 96.30% 96.23%]
F1 score - 80|20:       [99.94% 96.89% 96.23%]
Accuracy - 80|20:       [99.45%]

```

Confusion Matrix - Logistic Regression (Splitting dataset: Train=80%, Test=20%) - Not balanced

True label \ Predicted label	No	Probable	Migraine
No	True No 782 85.37%	False Probable 0 0.00%	False Migraine 0 0.00%
Probable	False No 1 0.11%	True Probable 78 8.52%	False Migraine 2 0.22%
Migraine	False No 0 0.00%	False Probable 2 0.22%	True Migraine 51 5.57%

Accuracy=0.995

*Nota.* Matriz de confusión con el 80 % de entrenamiento y 20 % de test para las 3 clases *migraña*, *probable migraña* y *migraña*.

**Tabla 5.** Regresión Logística Multinomial - Métricas obtenidas con el 80 % de entrenamiento y 20 % de test.

Metric	No Migraine	Probable Migraine	Migraine
Precision	99.87 %	97.5 %	96.23 %
Recall	100 %	96.3 %	96.23 %
F1	99.94 %	96.89 %	96.23 %
	Default Umbral	Default Umbral	Default Umbral
	<b>0.5</b>	<b>0.5</b>	<b>0.5</b>

*Nota.* Utilizando un 80 % para el entrenamiento del modelo y un 20 % para el test, las métricas siguen siendo bastante buenas, el riesgo de sobre ajuste es menor.

Aún así, es conveniente dividir el conjunto de datos en distintos bloques de entrenamiento y de test distribuidos a lo largo de todos los datos, para verificar que el modelo es adecuado. Con este criterio en mente, se puede desarrollar un algoritmo que vaya entrenando el modelo sobre cada uno de los bloques de datos para test y así obtener la probabilidad correspondiente.

Para realizar esta división distribuida de los datos en este trabajo se ha utilizado la función *Stratified K-fold* de la librería *scikit-learn* (Pedregosa et al., 2011).

#### 4.2.8. Aplicar una validación cruzada *Stratified K-fold Cross-Validation*

Como se ha comentado anteriormente, es conveniente verificar los resultados haciendo una validación cruzada para comprobar que los resultados son buenos dividiendo los datos de test y de entrenamiento a lo largo de todo el conjunto de datos (*8 partes*) para así obtener las métricas en promedio para cada uno de los bloques de datos. Se puede ver en el siguiente fragmento de código [Listado 4.4](#), extraído del cuaderno, un ejemplo de validación cruzada en lenguaje *Python*.

**Fragmento de código 4.4.** Método validación cruzada de los datos en 8 fragmentos y cálculo de las métricas promedio.

```

1  ...
2  number_splits = 8
3  # Provides train/test indices to split data in train/test sets, num splits = 8
4  # StratifiedKFold object is a variation of KFold that returns stratified folds.
5  # The folds are made by preserving the percentage of samples for each class.
6  skf = StratifiedKFold(n_splits=number_splits, shuffle=True, random_state = 0)
7  skf.get_n_splits(X, y)
8
9  #for train_index, test_index in kf.split(X):
10 for train_index, test_index in skf.split(X,y):
11     XTrain, XTest = X.iloc[train_index], X.iloc[test_index]
12     yTrain, yTest = y.iloc[train_index], y.iloc[test_index]
13     yMigrainePredictKFold = LRegression_Multi_Model.fit(XTrain, yTrain.values.ravel()).predict(XTest)
14     # precision_score

```

```

15     var_precision_score = precision_score(yTest, yMigrainePredictKFold, average=None)
16     prec_avg_no_migraine = prec_avg_no_migraine + var_precision_score[0]
17     prec_avg_possible_migraine = prec_avg_possible_migraine + var_precision_score[1]
18     prec_avg_migraine = prec_avg_migraine + var_precision_score[2]
19     #print("Precision score: " , var_precision_score)
20
21     # F-score
22     var_f1_score = f1_score(yTest, yMigrainePredictKFold, average=None)
23     f1_avg_no_migraine = f1_avg_no_migraine + var_f1_score[0]
24     f1_avg_possible_migraine = f1_avg_possible_migraine + var_f1_score[1]
25     f1_avg_migraine = f1_avg_migraine + var_f1_score[2]
26     #print("F1 score: " , var_f1_score)
27
28     # recall_score
29     var_recall_score = recall_score(yTest, yMigrainePredictKFold, average=None)
30     recall_avg_no_migraine = recall_avg_no_migraine + var_recall_score[0]
31     recall_avg_possible_migraine = recall_avg_possible_migraine + var_recall_score[1]
32     recall_avg_migraine = recall_avg_migraine + var_recall_score[2]
33     #print("Recall score: " , var_recall_score)
34
35     # accuracy_score
36     var_accuracy_score = accuracy_score(yTest, yMigrainePredictKFold)
37     accuracy_avg = accuracy_avg + var_accuracy_score
38     #print("Accuracy score: " , var_accuracy_score)
39
40     # precision metric average
41     prec_avg_no_migraine = prec_avg_no_migraine / number_splits
42     prec_avg_possible_migraine = prec_avg_possible_migraine / number_splits
43     prec_avg_migraine = prec_avg_migraine / number_splits
44
45     # recall metric average
46     recall_avg_no_migraine = recall_avg_no_migraine / number_splits
47     recall_avg_possible_migraine = recall_avg_possible_migraine / number_splits
48     recall_avg_migraine = recall_avg_migraine / number_splits
49
50     # f1 metric average
51     f1_avg_no_migraine = f1_avg_no_migraine / number_splits
52     f1_avg_possible_migraine = f1_avg_possible_migraine / number_splits
53     f1_avg_migraine = f1_avg_migraine / number_splits
54
55     # accuracy metric average
56     accuracy_avg = accuracy_avg / number_splits
57     ...

```

Utilizado una validación cruzada y distribuida en 8 partes se obtienen unas métricas en promedio bastante buenas como se puede ver en la [Tabla 6](#), así pues, se determina que no hay un sobre ajuste aparente y el modelo es bastante bueno utilizado la regresión logística con los parámetros utilizados. No obstante, se puede intentar mejorar el modelo con algunos de los métodos disponibles que se describen en los siguientes apartados.

**Tabla 6.** Regresión Logística Multinomial - Métricas promedio de la validación cruzada de los datos en 8 fracciones distribuidas.

Metric	No Migraine	Probable Migraine	Migraine
Precision	99.88 %	94.5 %	96.0 %
Recall	99.6 %	95.8 %	96.6 %
F1	99.7 %	95.1 %	96.2 %
	Default Umbral	Default Umbral	Default Umbral
	<b>0.5</b>	<b>0.5</b>	<b>0.5</b>

*Nota.* El modelo se ajusta bastante bien, incluso realizando una validación cruzada sin ningún tipo de optimización.

### 4.3. Técnicas de mejora del modelo de regresión logística *multinomial*

Aunque el modelo de regresión logística *multinomial* es bastante efectivo en todas las métricas utilizadas, como se ha detallado en el apartado anterior, el caso práctico y real implica tratar de mejorar su rendimiento con alguna o varias de las métricas utilizadas en este trabajo y que han sido detalladas en 4.2.5. A continuación se van a describir los diferentes métodos utilizados para la optimización de modelo, verificando si son efectivos o no, para finalmente seleccionar el método de optimización más adecuado.

#### 4.3.1. Ajustar los pesos en función de la frecuencia de las clases objetivo

Una de las maneras directas de tratar de mejorar la eficacia del modelo de regresión logística planteado, ver Listado 4.2, consiste en añadir un parámetro, concretamente ***class\_weight = 'balanced'***, ya que en un conjunto de datos no balanceado, como es el caso, puede ser interesante desplazar el límite de decisión por medio de potenciar el peso de las clases minoritarias, que en este caso son *Migraña=5.8%* y *Probable Migraña=8.8%* frente a la mayoritaria, *No Migraña=85.4%*.

Al pasar este parámetro en la construcción del modelo lo que se consigue es que los pesos se ajustan inversamente proporcional a la frecuencia de las clases, para más información consultar *sklearn.linear\_model.LogisticRegression*. Así pues, en este caso se repite el proceso de validación cruzada utilizado en Listado 4.4 pero añadiendo en la instanciación de la clase el parámetro ***class\_weight = 'balanced'*** como en el fragmento de código Listado 4.5.

**Fragmento de código 4.5.** Método validación cruzada con el parámetro `class_weight = 'balanced'` en el constructor.

```

1  ...
2  #####
3  # Multinomial 'balanced' logistic regression model instance
4  #####
5  # class_weight = 'balanced' (convenient for unbalanced dataset) -> mode uses the values of y to automatically adjust
6  # weights inversely proportional to class frequencies in the input data as n_samples / (n_classes * np.bincount(y)).
7  LRRegression_Multi_Model_Balanced = LogisticRegression(multi_class='multinomial', solver='lbfgs', random_state = 0,
8  class_weight = 'balanced')
  ...

```

Los resultados obtenidos en promedio de las diferentes métricas realizando una validación cruzada de 8 divisiones de el 80 % de entrenamiento y el 20 % de prueba con el parámetro `class_weight = 'balanced'` en el constructor del modelo se muestran en la siguiente [Tabla 7](#):

**Tabla 7.** Regresión Logística Multinomial - Métricas promedio de la validación cruzada con `class_weight = 'balanced'`.

Metric	No Migraine	Probable Migraine	Migraine
Precision	100 %	78.4 %	96.4 %
Recall	97.2 %	97.5 %	99.6 %
F1	98.6 %	86.9 %	98.0 %
	Default Umbral	Default Umbral	Default Umbral
	<b>0.5</b>	<b>0.5</b>	<b>0.5</b>

*Nota.* El modelo no se ajusta tan bien al pasar el parámetro, sobre todo por la bajada en la precisión y la F1 para la clase *probable migraña*.

Para este conjunto de datos, si bien ha mejorado la métrica *F1* del 96.2 % al 98.0 % para la clase minoritaria *Migraña*, se han producido más falsos positivos para la clase *Probable migraña*, esto es debido a que el límite de decisión entre la clase mayoritaria *No migraña* y la clase *Probable migraña* se ha visto desplazado en favor de la última en exceso, haciendo que la métrica *F1* descienda hasta 86.9 % lo cual no está muy mal, pero era bastante mejor, del 95.1 %, cuando no se había modificado el modelo con el parámetro `class_weight`.

**CONCLUSIÓN**, la modificación realizada tratando de mejorar el rendimiento por medio de potenciar los pesos de manera inversamente proporcionales a la frecuencia de las clases **queda descartada** por un excesivo desplazamiento del límite de decisión entre la clase *Probable Migraña* y la clase *No Migraña*.



### 4.3.2. Ajuste del umbral óptimo del modelo

Entre las diferentes soluciones que hay para mejorar el rendimiento del modelo obtenido en conjuntos de datos no balanceados como es el caso, existen varias aproximaciones o vías que se pueden intentar como quedan muy bien detalladas en los siguientes artículos (Rocca, 2021) y (Brownlee, 2020). Entre las diferentes soluciones existentes hay unas cuantas relacionadas con realizar modificaciones sobre los datos de partida, como las que se detallan a continuación:

- Utilizar técnicas de **sub-muestreo** de la clase predominante, en este caso es *No migraña* frente a las otras 2 que son minoritarias. Consiste básicamente en eliminar muestras mayoritarias tratando de equilibrar el conjunto de datos.
- Utilizar técnicas de **sobre-muestreo** de las clases minoritarias. Consiste en generar muestras de las clases minoritarias para equilibrar el conjunto de datos.
- Generar **nuevas muestras** de las clases minoritarias de manera artificial utilizando métodos o algoritmos como los disponibles en la librería (Lemaître et al., 2017), como por ejemplo el método **SMOTE**.

Aunque, en muchas ocasiones, una simple solución sin necesidad de generar o eliminar los datos originales, y que suele dar buenos resultados consiste en modificar el umbral utilizado por el modelo elegido para la clasificación, que por defecto es (**umbral=0.5**), con el objetivo de encontrar el umbral óptimo del modelo para obtener el mejor rendimiento del mismo. De entre los diferentes métodos para optimizar el umbral óptimo (Brownlee, 2020), los más habituales son detallados en los siguientes apartados para obtener los umbrales óptimos del conjunto de datos.

#### 4.3.2.1. Utilizar la curva ROC

La *Tasa de Falsos Positivos (FPR)* se traza en el eje x que es la inversa de la *Tasa de Verdaderos Negativos (TNR-Specificity)* mientras que la *Tasa de Verdaderos Positivos (TPR-Sensitivity-Recall)* se traza en el eje y. Es ideal cuando lo que se busca es el balance óptimo entre la *especificidad* y la *sensibilidad*.

Cuanto mayor sea el área por encima de la curva sin habilidad (diagonal de 45°) mejores resultados da el modelo.

Para el ajuste del umbral utilizando la curva **ROC** se pueden usar varios métodos (Brownlee, 2020), como los 2 siguientes que se detallan a continuación:

- La media geométrica **GMean** expresada como la raíz cuadrada de la *sensibilidad* y la *especificidad*, donde ambos parámetros se rigen según las siguientes ecuaciones:

$$Sensitivity = TruePositiveRate = \frac{TruePositive}{TruePositive + FalseNegative} \quad (4.5)$$

$$Specificity = 1 - FalsePositiveRate = \frac{TrueNegative}{FalsePositive + TrueNegative} \quad (4.6)$$

- Un método más simple computacional mente hablando es utilizar la estadística **J de Youden** que se rige con la siguiente ecuación:

$$J = Sensibility + Specificity - 1 = TruePositiveRate - FalsePositiveRate \quad (4.7)$$

Para la obtención de la curva **ROC**, en este trabajo se ha utilizado, como se muestra en el fragmento de código [Listado 4.6](#) la estadística *J* porque es más simple con resultados similares a cuando se utiliza la media geométrica.

**Fragmento de código 4.6.** Curvas ROC obtenidas utilizando la estadística *J* con un *dataset* 80 % de entrenamiento y 20 % de prueba.

```

1  ...
2  for i in range(n_classes):
3      fpr, tpr, thresholds = roc_curve(GetPredictionMigraineArray(i,yTest), y_predict_proba_80_20[:, i]) roc_auc = auc(fpr,
4          tpr)
5          #####
6          # Extrapolating a multiclass example from [12] (Brownlee, 2020).
7          # Apply the J Youden statistic method to calculate the optimal umbral
8          # https://en.wikipedia.org/wiki/Youden%27s_J_statistic
9          #####
10         J = tpr - fpr
11         ix = argmax(J)
12         if i==0:
13             thresholdROCOptmNoMigraine = thresholds[ix]
14         if i==1:
15             thresholdROCOptmProbMigraine = thresholds[ix]
16         if i==2:
17             thresholdROCOptmMigraine = thresholds[ix]
18     ...

```

*Nota.* Fragmento de código adaptado de (Brownlee, 2020).

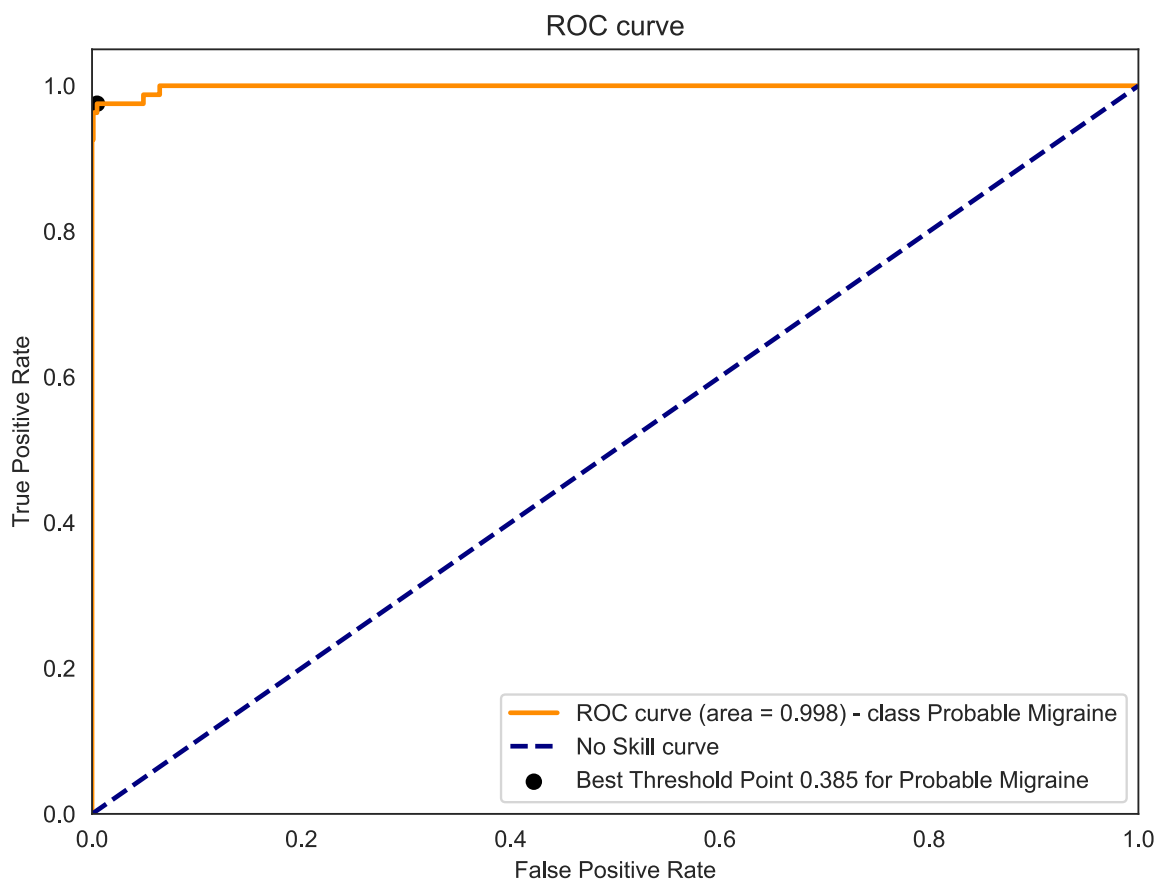
La curva *ROC* obtenida en este caso solamente para la clase *Probable Migraña* con un punto marcando el umbral óptimo para esa clase, junto con los umbrales óptimos obtenidos para las tres clases por medio de las curvas *ROC* se pueden ver en la [Figura 12](#) donde puede observarse que hay un gran área 0.998 por encima de la diagonal correspondiente a lo que sería una curva sin habilidad, con lo que se deduce que el modelo es bastante hábil.

**Figura 12.** Curva *ROC* de la clase *Probable Migraine* utilizando la estadística *J*.

ROC -> Best Threshold No Migraine Class = 0.673

ROC -> Best Threshold Probable Migraine Class = 0.385

ROC -> Best Threshold Migraine Class = 0.454



*Nota.* Para obtener la curva *ROC* se ha utilizado una distribución del conjunto de datos con un 80 % de entrenamiento y 20 % de prueba.

Una vez se han obtenido mediante las curvas *ROC* los tres umbrales óptimos, se obtienen las métricas utilizando los umbrales calculados.

En el siguiente fragmento [Listado 4.7](#), extraído del cuaderno de *Jupyter*, se describe el procedimiento utilizado para obtener las métricas con los umbrales óptimos calculados, el cual es repetido posteriormente para los siguientes métodos de optimización de los umbrales, y por lo tanto no será desarrollado posteriormente en este documento, consiste en:

- **Calcular** la matriz de probabilidades de las 3 clases a través del modelo utilizando el método *predict\_proba* (Pedregosa et al., 2011).
- **Obtener** la matriz de predicciones de cada clase mediante la comparación de la matriz de probabilidades obtenida al comparar la probabilidad con el umbral óptimo de cada clase.
- **Imprimir** los resultados de forma legible con los porcentajes de cada métrica obtenidos.

#### Fragmento de código 4.7. Método de cálculo de las métricas en función de los umbrales óptimos.

```

1  ...
2  # Predict the probabilities for Xtest (20% stratified)
3  # return matrix with probabilities for -> No migraine 0, Probable Migraine 1 and Migraine 2
4  y_predict_proba_80_20 = LRegression_Multi_Model.fit(XTrain, yTrain.ravel()).predict_proba(XTest)
5
6  # Getting the metrics with thresholds calculated from ROC curves:
7  # thresholdROCOptmNoMigraine = 0.673
8  yNoMigrainePredictThresholdROCOpt = (y_predict_proba_80_20[:, 0] >= thresholdROCOptmNoMigraine).astype(int)
9  # thresholdROCOptmProbMigraine = 0.385
10 yProbableMigrainePredictThresholdROCOpt = (y_predict_proba_80_20[:, 1] >= thresholdROCOptmProbMigraine).astype(int)
11 # thresholdROCOptmMigraine = 0.454
12 yMigrainePredictThresholdROCOpt = (y_predict_proba_80_20[:, 2] >= thresholdROCOptmMigraine).astype(int)
13
14 # Precision for optimal threshold from ROC curves
15 print('Precision with threshold optimized from ROC curve: [{0:2.2%} {1:2.2%} {2:2.2%}]'.format(round(precision_score(
    GetPredictionMigraineArray(0, yTest), yNoMigrainePredictThresholdROCOpt),4), round(precision_score(
    GetPredictionMigraineArray(1, yTest), yProbableMigrainePredictThresholdROCOpt),4), round(precision_score(
    GetPredictionMigraineArray(2, yTest), yMigrainePredictThresholdROCOpt),4)))
16 # Recall for optimal threshold from ROC curves
17 print('Recall with threshold optimized from ROC curve:  [{0:2.2%} {1:2.2%} {2:2.2%}]'.format(round(recall_score(
    GetPredictionMigraineArray(0, yTest), yNoMigrainePredictThresholdROCOpt),4), round(recall_score(
    GetPredictionMigraineArray(1, yTest), yProbableMigrainePredictThresholdROCOpt),4), round(recall_score(
    GetPredictionMigraineArray(2, yTest), yMigrainePredictThresholdROCOpt),4)))
18 # F-score for optimal threshold from ROC curves
19 print('F1 score with threshold optimized from ROC curve:  [{0:2.2%} {1:2.2%} {2:2.2%}]'.format(round(f1_score(
    GetPredictionMigraineArray(0, yTest), yNoMigrainePredictThresholdROCOpt),4), round(f1_score(
    GetPredictionMigraineArray(1, yTest), yProbableMigrainePredictThresholdROCOpt),4), round(f1_score(
    GetPredictionMigraineArray(2, yTest), yMigrainePredictThresholdROCOpt),4)))
20 ...

```

Los umbrales y las métricas calculadas por medio de las curvas ROC se exponen a continuación en la [Tabla 8](#):

**Tabla 8.** *Regresión Logística Multinomial - Métricas promedio con umbrales óptimos obtenidos de las curvas ROC.*

<b>Metric</b>	<b>No Migraine</b>	<b>Probable Migraine</b>	<b>Migraine</b>
Precision	100 %	95.2 %	94.6 %
Recall	99.7 %	97.5 %	100.0 %
F1	99.9 %	96.3 %	97.25 %
	ROC Umbral	ROC Umbral	ROC Umbral
	<b>0.673</b>	<b>0.385</b>	<b>0.454</b>

*Nota.* Las métricas para los umbrales óptimos obtenidos de las curvas ROC han sido obtenidas a partir de una distribución del conjunto de datos con un 80 % de entrenamiento y 20 % de prueba.

#### **4.3.2.2. Utilizar la curva de Recuperación de Precisión (Precision-Recall)**

En conjuntos de datos no balanceados como es este caso, una curva interesante que permite evaluar la eficacia del modelo elegido es la curva de *recuperación de precisión*, la cual puede ser utilizada en vez de la habitual curva ROC con su área inferior *AUC*, ya que la curva de *recuperación de precisión* está más enfocada y es más hábil para informar sobre el rendimiento de modelos predictivos para conjuntos de datos no balanceados como indican (Saito y Rehmsmeier, 2015). Es decir, este tipo de curvas resulta más interesante cuando lo que se trata es de optimizar las clases minoritarias buscadas, en este caso *Probable Migraña* y *Migraña*. El método seguido para obtener la curva se basa en ir calculando tanto la precisión como la recuperación para un conjunto de umbrales y así poder obtener un gráfico donde los umbrales van creciendo con la recuperación en el eje x y la precisión en el eje y.

Para relacionar la precisión y la recuperación se utiliza la denominada métrica **F1** por medio de las ecuaciones ya desarrolladas en el apartado 4.2.5 de este documento, repitiendo el mismo algoritmo del apartado anterior para las curvas ROC 4.3.2.1, pero con las pequeñas modificaciones necesarias que se detallan en el fragmento código siguiente [Listado 4.8](#) para las curvas de recuperación de precisión.

**Fragmento de código 4.8.** Fragmento de código para obtener la curva de recuperación de precisión utilizando la métrica *F1*.

```

1  ...
2  for i in range(n_classes):
3      #if (i > 0): # Probable Migraine or Migraine classes
4          precision, recall, thresholds = precision_recall_curve(GetPredictionMigraineArray(i, yTest), y_predict_proba_80_20[:,
           i])
5          average_precision = average_precision_score(GetPredictionMigraineArray(i, yTest), y_predict_proba_80_20[:, i])
6          # convert to f score
7          fscore = (2 * precision * recall) / (precision + recall)
8          # locate the index of the largest f score
9          ix = argmax(fscore)
10  ...

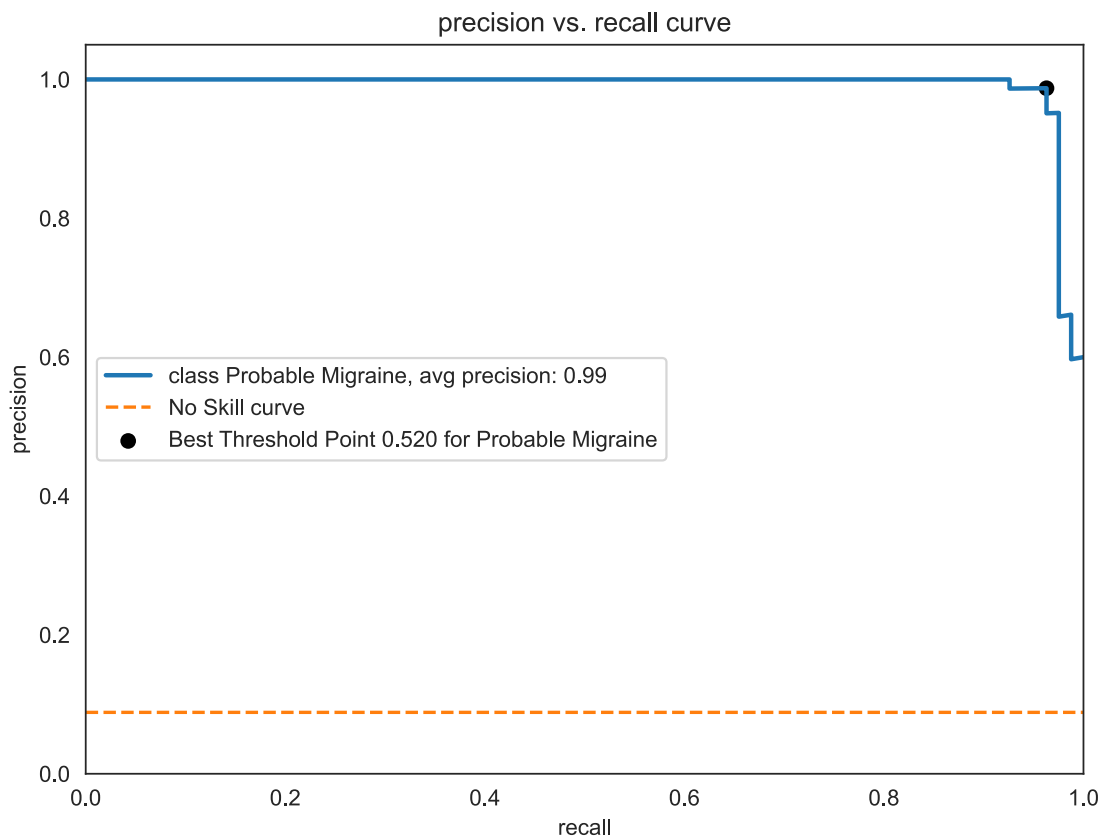
```

*Nota.* Código adaptado de (Brownlee, 2020).

La curva de *recuperación de precisión* obtenida en la [Figura 13](#), en este caso solamente para la clase *Probable Migraña*, incluye el punto marcando el umbral óptimo para la métrica *F1* que equilibra la *precisión* y la *recuperación* para esa clase, además se muestran los umbrales óptimos de la métrica *F1* obtenidos para las tres clases.

**Figura 13.** Curva *Precision-Recall* de la clase *Probable Migraine* utilizando la estadística *F1*.

Precision Recall -> Best Threshold No Migraine Class = 0.499, F-Score = 99.94%  
 Precision Recall -> Best Threshold Probable Migraine Class = 0.520, F-Score = 97.50%  
 Precision Recall -> Best Threshold Migraine Class = 0.453, F-Score = 97.25%



*Nota.* Para obtener la curva *recuperación de precisión* se ha utilizado una distribución del conjunto de datos con un 80 % de entrenamiento y 20 % de prueba.

Los umbrales y las métricas calculadas a través de la curva de *recuperación de precisión* se exponen a continuación en la [Tabla 9](#):

**Tabla 9.** *Regresión Logística Multinomial - Métricas promedio con umbrales óptimos obtenidos de la curva de recuperación de precisión.*

Metric	No Migraine	Probable Migraine	Migraine
Precision	99.87 %	98.73 %	94.64 %
Recall	100 %	96.3 %	100.0 %
F1	99.94 %	<b>97.5 %</b>	<b>97.25 %</b>
	PR Umbral	PR Umbral	PR Umbral
	<b>0.499</b>	<b>0.520</b>	<b>0.453</b>

*Nota.* Las métricas para los umbrales óptimos obtenidos de las curvas de *recuperación de precisión* han sido obtenidas a partir de una distribución del conjunto de datos con un 80 % de entrenamiento y 20 % de prueba.

Se destaca una mejora clara utilizando los umbrales optimizados para la métrica **F1** en las clases minoritarias respecto a los valores obtenidos originalmente [Figura 11](#), con el mismo conjunto de datos, pero utilizando el umbral por defecto que era **0.5**. Así pues, para la clase *Probable Migraña* se pasa del 96.89 % al 97.5 % y para la clase *Migraña* del 96.23 % al 97.25 %.

#### 4.3.2.3. **Método genérico para obtener el ajuste óptimo del umbral para mejorar la métrica F1 score**

El método genérico utilizado a continuación para el ajuste óptimo del umbral con el objetivo de mejorar la *F1* es adecuado cuando el número de datos no es muy elevado como es este caso ya que su rendimiento computacional se ve afectado cuanto mayor es el número de datos disponibles.

Este es un proceso resumido de lo que va haciendo el algoritmo utilizado y que está basado en el trabajo de ([Aprilliant, 2021](#)) y ([Brownlee, 2020](#)):

- El algoritmo utilizado, como muestra el [Listado 4.9](#), va recorriendo el rango de valores válidos de cada umbral de las 3 clases desde **0** como valor mínimo hasta **1** como valor máximo.
- El número de pasos en que es subdividido el rango del umbral, que ha sido fijado en 1000 (**step\_points=1000**), es directamente proporcional a la precisión del umbral que se busca, pero claro es que un aumento de los pasos hará que la carga computacional sea mucho mayor, obviamente hay que llegar a un compromiso entre eficacia y precisión.

- En cada paso del bucle se compara la probabilidad de la clase correspondiente extraída del vector de probabilidades con el umbral correspondiente, en todos y cada uno de los 1000 pasos y también se calcula la  $F1$  para cada paso del umbral mediante la función `f1_score` de la librería **Scikit-learn** (Pedregosa et al., 2011).
- Posteriormente se extrae la métrica con el valor máximo de  $F1$  para cada clase de la matriz obtenida.
- Por último se representa gráficamente la evolución de los umbrales para las tres clases utilizando la librería **ggplot2** (Wickham, 2016).

#### Fragmento de código 4.9. Método genérico para obtener los umbrales óptimos para la métrica $F1$ .

```

1  ...
2  #####
3  # Threshold tuning functions
4  # Adapted from: Aprilliant, A. (2021, enero 1). Optimal Threshold for Imbalanced Classification.
5  # Medium. https://towardsdatascience.com/optimal-threshold-for-imbalanced-classification-5884e870c293
6  #####
7  def getOptimalThresholdTuning(y_pred, y_test, step_points = 1000, plotting=True, title = 'Threshold Tuning Curve',
8  className = 'Migraine', precisionRecallThreshold = 0.5):
9  thresholds = np.arange(0.0, 1.0, 1/step_points) # default 1000 steps
10 fscore = np.zeros(shape=(len(thresholds)))
11 #print('Length of sequence: {}'.format(len(thresholds)))
12
13 # Fit the model
14 for index, elem in enumerate(thresholds):
15     # Corrected probabilities
16     y_pred_prob = (y_pred > elem).astype('int')
17     # Calculate the f-score
18     fscore[index] = f1_score(y_test, y_pred_prob)
19
20 # Find the optimal threshold
21 index = np.argmax(fscore)
22 thresholdOpt = round(thresholds[index], ndigits = 4)
23 fscoreOpt = round(fscore[index], ndigits = 4)
24 ...

```

*Nota.* Fragmento de código adaptado de (Aprilliant, 2021).

Los umbrales óptimos obtenidos mediante este método de ajuste del umbral paso a paso para mejorar la métrica  $F1$  se muestran a continuación en [Figura 14](#), [Figura 15](#) y [Figura 16](#).

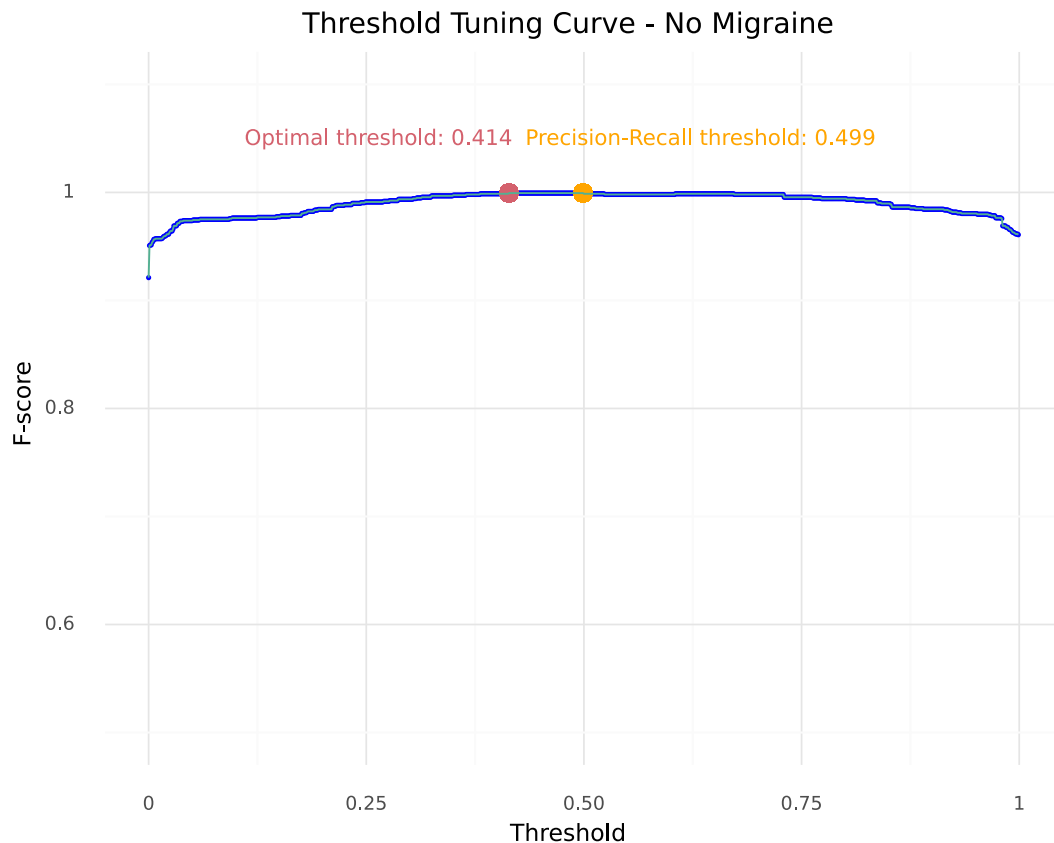


Para la clase mayoritaria *No Migraña* el umbral óptimo obtenido es **0.414** que se corresponde al mínimo valor del umbral donde la métrica *F1* es máxima, el valor máximo corresponde al umbral obtenido utilizando las curvas de *recuperación de precisión* que es **0.499**, curvas que también optimizan los umbrales para la métrica *F1* como puede verse en la [Figura 14](#) para la clase *No Migraña*.

**Figura 14.** Umbral óptimo 0.414, método paso a paso de la clase **No Migraña** utilizando la métrica *F1*.

Class: No Migraine -> Best Optimal Threshold Tuning: 0.414 with F-Score: 99.94%

Class: No Migraine -> Best Precision Recall Curve Threshold: 0.499 with F-Score: 99.94%

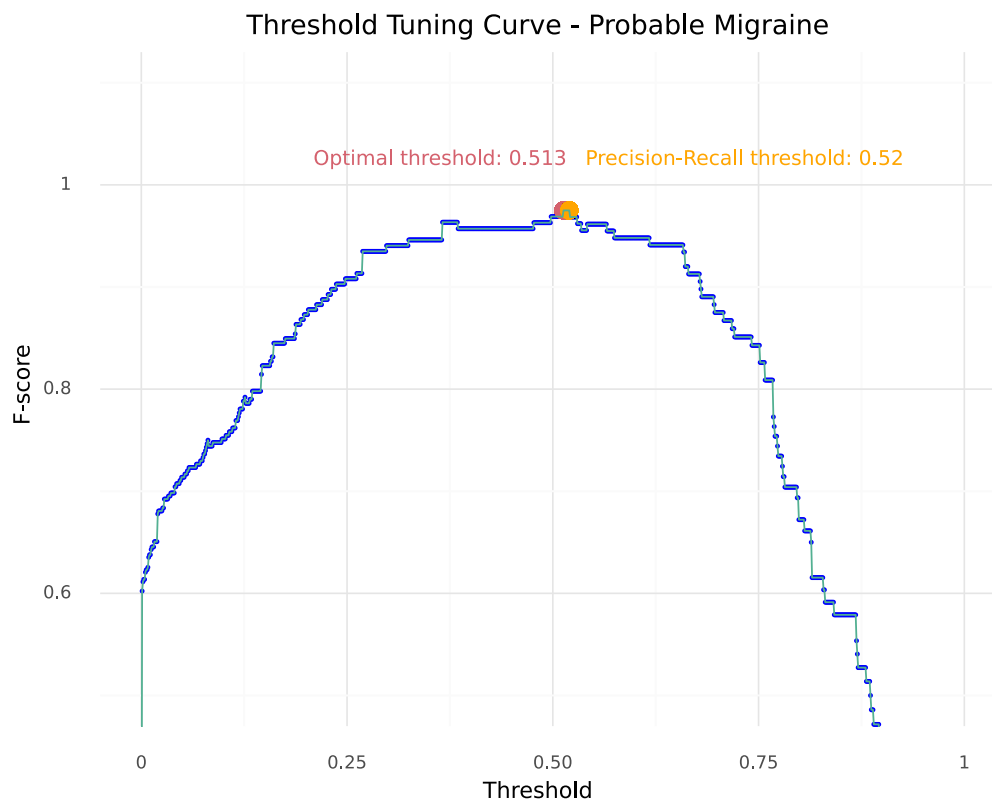


*Nota.* Para obtener el umbral óptimo por el métodos paso a paso se ha utilizado una distribución del conjunto de datos con un 80 % de entrenamiento y 20 % de prueba.

Para la clase minoritaria *Probable Migraña* el umbral óptimo obtenido es **0.513** que se corresponde al mínimo valor del umbral donde la métrica *F1* es máxima, el valor máximo corresponde al umbral obtenido utilizando las curvas de *recuperación de precisión* es **0.520**, curvas que también optimizan los umbrales para la métrica *F1* como puede verse en la [Figura 15](#) para la clase *Probable Migraña*.

**Figura 15.** Umbral óptimo, método paso a paso de la clase *Probable Migraña* utilizando la estadística *F1*.

Class: Probable Migraine -> Best Optimal Threshold Tuning: 0.513 with F-Score: 97.50%  
Class: Probable Migraine -> Best Precision Recall Curve Threshold: 0.52 with F-Score: 97.50%

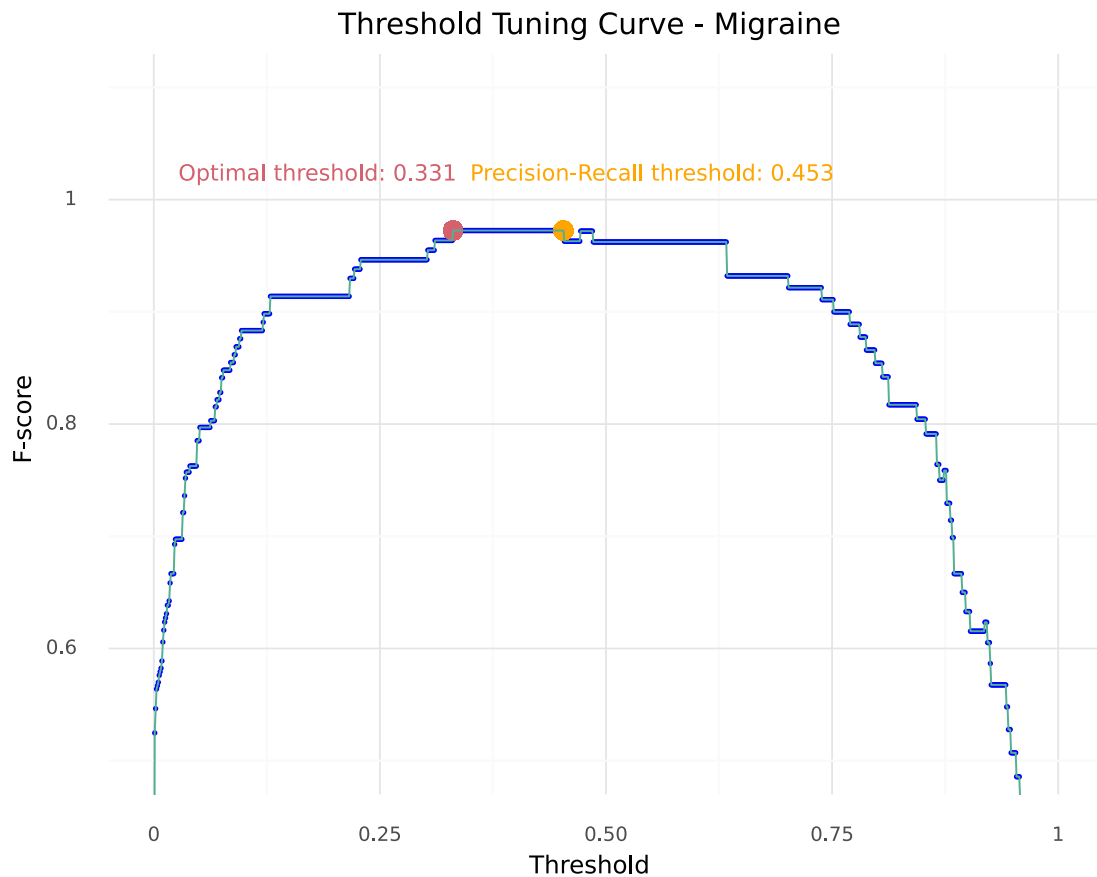


*Nota.* Para obtener el umbral óptimo por el métodos paso a paso se ha utilizado una distribución del conjunto de datos con un 80 % de entrenamiento y 20 % de prueba.

Para la clase minoritaria *Migraña* el umbral óptimo obtenido es **0.331** que se corresponde al mínimo valor del umbral donde la métrica *F1* es máxima, el valor máximo corresponde al umbral obtenido utilizando las curvas de *recuperación de precisión* es **0.453**, curvas que también optimizan los umbrales para la métrica *F1* como puede verse en la [Figura 16](#) para la clase *Migraña*.

**Figura 16.** Umbral óptimo, método paso a paso de la clase *Migraña* utilizando la métrica *F1*.

Class: Migraine -> Best Optimal Threshold Tuning: 0.331 with F-Score: 97.25%  
Class: Migraine -> Best Precision Recall Curve Threshold: 0.453 with F-Score: 97.25%



*Nota.* Para obtener el umbral óptimo por el métodos paso a paso se ha utilizado una distribución del conjunto de datos con un 80 % de entrenamiento y 20 % de prueba.

#### 4.3.2.4. ***Método genérico para obtener el ajuste óptimo del umbral para mejorar la precisión***

El método genérico utilizado a continuación para el ajuste óptimo del umbral con el objetivo de mejorar la precisión es adecuado cuando el número de datos no es muy elevado como es este caso ya que permite recorrer con un paso muy fino (0.001) todos los umbrales posibles desde 0 a 1 hasta obtener los umbrales óptimos que permiten aumentar al máximo la precisión del modelo.

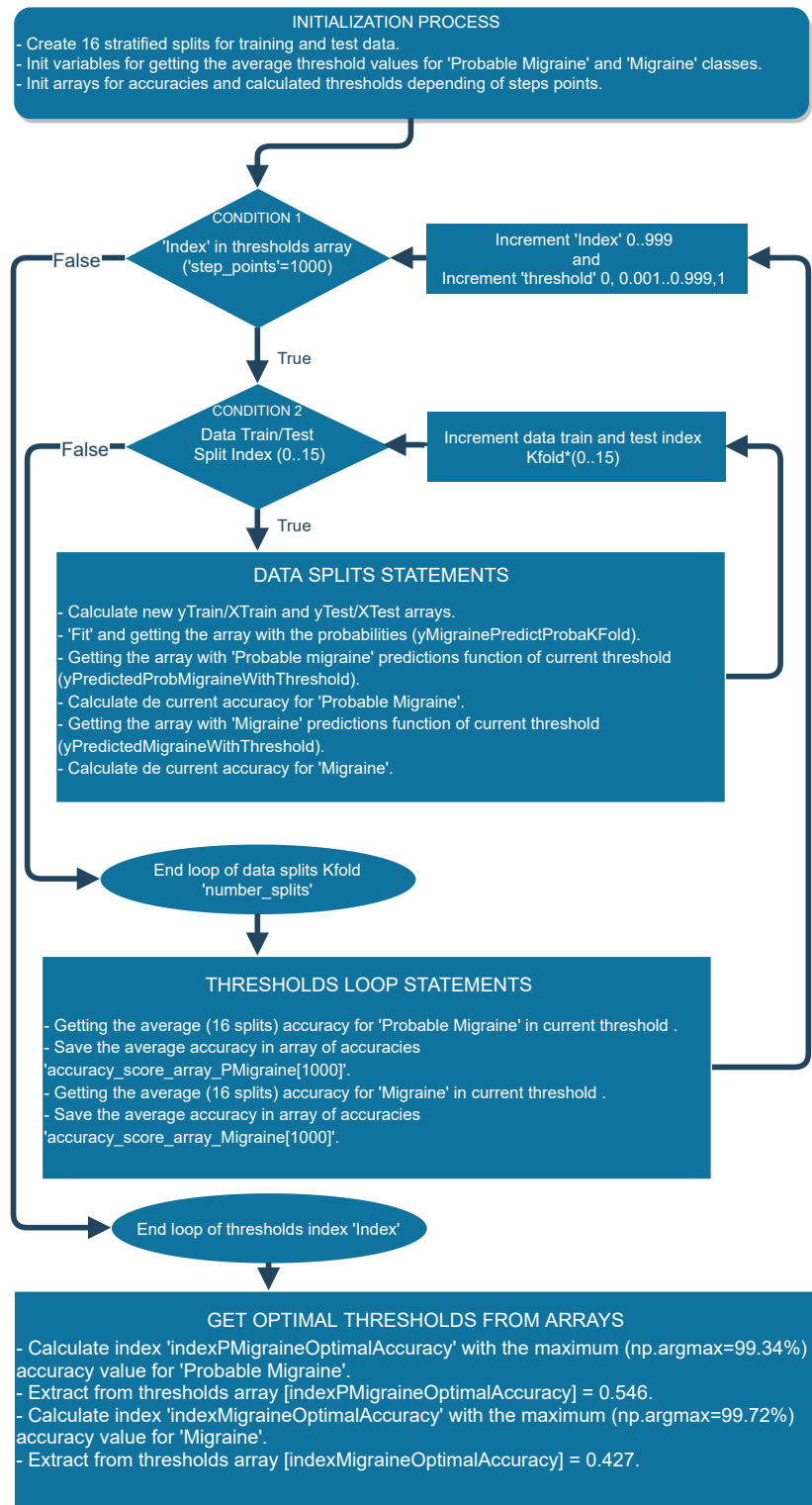
Para evitar el sobre ajuste y teniendo en cuenta que la métrica decidida a mejorar en el modelo de regresión, es la *precisión*, hay que repetir el bucle anidado con diferentes fracciones (*Kfolds*), en este caso **16**, con datos de entrenamiento y datos de test que permitan garantizar que el modelo se ajusta sobre datos nuevos cada vez.

Por lo tanto, en cada iteración del umbral (0-1000) se ejecutan 16 iteraciones de las diferentes fracciones de datos de entrenamiento y test distribuidas a lo largo del conjunto de datos obteniendo el umbral óptimo promedio de las mismas, así se genera la matriz de umbrales promedio optimizados (dim=1000) y se extrae el que ofrece una precisión mayor.

El siguiente método, ha sido diseñado específicamente en este trabajo para obtener los umbrales óptimos tanto para la clase *Probable Migraña* como la clase *Migraña*.

A continuación en la Figura 17, y para clarificar el algoritmo desarrollado, se describe el diagrama de bloques funcional.

**Figura 17.** Diagrama de bloques funcional para obtener los umbrales óptimos que mejoran el rendimiento del modelo mejorando la precisión.



*Nota.* Para obtener el umbral óptimo que mejora la *precisión* del modelo se ha utilizado una distribución de **16** bloques de datos distribuidos a lo largo del conjunto de datos, con un 80 % de datos entrenamiento y 20 % de datos de prueba para cada bloque.

A continuación en [Listado 4.10](#), un fragmento de código del método utilizado, hay que tener en cuenta que es una parte del método y que en [A Tablas de datos y Notebooks](#) está disponible el enlace para acceder al cuaderno de *Jupyter* con el código completo.

#### Fragmento de código 4.10. Método genérico para obtener los umbrales óptimos para la precisión.

```

1  ...
2  number_splits = 16
3  # Provides train/test indices to split data in train/test sets, num splits = 8
4  #kf = KFold(n_splits=number_splits, shuffle=True, random_state = 0)
5  # StratifiedKFold object is a variation of KFold that returns stratified folds.
6  # The folds are made by preserving the percentage of samples for each class.
7  skf = StratifiedKFold(n_splits=number_splits, shuffle=True, random_state = 0)
8  skf.get_n_splits(X, y)
9
10 # Variables for metrics average values
11 accuracy_avg_threshold_PMigraine = 0
12 accuracy_avg_threshold_Migraine = 0
13
14 step_points = 1000
15 thresholds = np.arange(0.0, 1.0, 1/step_points) # default 100 steps, thresholds 0, 0.01...0.99, 1.00
16 accuracy_score_array_PMigraine = np.zeros(shape=(len(thresholds)))
17 accuracy_score_array_Migraine = np.zeros(shape=(len(thresholds)))
18
19 for index, elem in enumerate(thresholds): # index 0->1000, elem 0.0, 0.001 -> 0.999, 1.00
20     accuracy_avg_threshold_PMigraine = 0
21     accuracy_avg_threshold_Migraine = 0
22     for train_index, test_index in skf.split(X, y):
23         XTrain, XTest = X.iloc[train_index], X.iloc[test_index]
24         yTrain, yTest = y.iloc[train_index], y.iloc[test_index]
25         yMigrainePredictProbaKFold = LRegression_Multi_Model.fit(XTrain, yTrain.values.ravel()).predict_proba(XTest)
26         #####
27         # Predicted array as a function of threshold for 'Probable Migraine' class
28         #####
29         #yPredictedProbMigraineWithThreshold = (yMigrainePredictProbaKFold[:,1] > elem).astype('int')
30         yPredictedProbMigraineWithThreshold = (yMigrainePredictProbaKFold[:,1] > elem).astype('int')
31         # Predicted array as a function of threshold for Migraine class
32         #yPredictedMigraineWithThreshold = (yMigrainePredictProbaKFold[:,2] > elem).astype('int')
33
34         # Accuracy function for the current fold data and current threshold
35         var_accuracy_score_threshold = accuracy_score(GetPredictionMigraineArray(1, yTest),
36             yPredictedProbMigraineWithThreshold)
37
38         # Alternative method to compare 2 arrays and return the accuracy, extracted from:
39         # https://stackoverflow.com/questions/38877301/how-to-calculate-accuracy-based-on-two-lists-python
40         #var_accuracy_score_threshold = sum(1 for a,b in zip(yPredictedWithThreshold, yTest.values.ravel()) if a == b)
41         # / len(yPredictedWithThreshold)
42         accuracy_avg_threshold_PMigraine = accuracy_avg_threshold_PMigraine + var_accuracy_score_threshold
43         #####
44         # Predicted array as a function of threshold for 'Migraine' class
45         #####
46         yPredictedMigraineWithThreshold = (yMigrainePredictProbaKFold[:,2] > elem).astype('int')
47
48         # Accuracy function for the current fold data and current threshold
49         var_accuracy_score_threshold = accuracy_score(GetPredictionMigraineArray(2, yTest),
50             yPredictedMigraineWithThreshold)
51
52         accuracy_avg_threshold_Migraine = accuracy_avg_threshold_Migraine + var_accuracy_score_threshold
53         #####
54         # End of internal loop -> Kfolds
55         #####
56
57     # accuracy metric average
58     accuracy_avg_threshold_PMigraine = accuracy_avg_threshold_PMigraine / number_splits
59     # save accuracy average in array for 'Probable Migraine' class
60     accuracy_score_array_PMigraine[index] = accuracy_avg_threshold_PMigraine

```

```

57 accuracy_avg_threshold_Migraine = accuracy_avg_threshold_Migraine / number_splits
58 # save accuracy average in array for 'Migraine' class
59 accuracy_score_array_Migraine[index] = accuracy_avg_threshold_Migraine
60
61 # Accuracy score with specific threshold
62 if (elem == 0.5): # default
63     print("Accuracy score average for 'Probable Migraine', threshold {(0:.3f)} - {1:0} Splits - Kfold: [{2:2.2%}].
        format(elem, number_splits, round(accuracy_avg_threshold_PMigraine, 4)))
64     print("Accuracy score average for 'Migraine', threshold {(0:.3f)} - {1:0} Splits - Kfold: [{2:2.2%}].format(elem,
        number_splits, round(accuracy_avg_threshold_Migraine, 4)))
65 #####
66 # End of external loop -> Thresholds
67 #####
68
69 #####
70 # Find the optimal threshold 'Probable Migraine'
71 #####
72 indexPMigraineOptimalAccuracy = np.argmax(accuracy_score_array_PMigraine)
73 thresholdPMigraineOptForAccuracy = round(thresholds[indexPMigraineOptimalAccuracy], ndigits = 4)
74 accuracyPMigraineThresholdOpt = round(accuracy_score_array_PMigraine[indexPMigraineOptimalAccuracy], ndigits = 4)
75 print('Class: {0} -> Best Optimal Accuracy: [{1:2.2%}] with Threshold: {(2:.3f)}'.format(classNames[1],
        accuracyPMigraineThresholdOpt, thresholdPMigraineOptForAccuracy))
76 ...

```

Después de unos minutos, que pueden ser más o menos dependiendo de la computadora utilizada, ya que son dos bucles de 16x1000 iteraciones, se obtienen los siguientes umbrales óptimos para el conjunto de datos recogidos en la [Tabla 10](#):

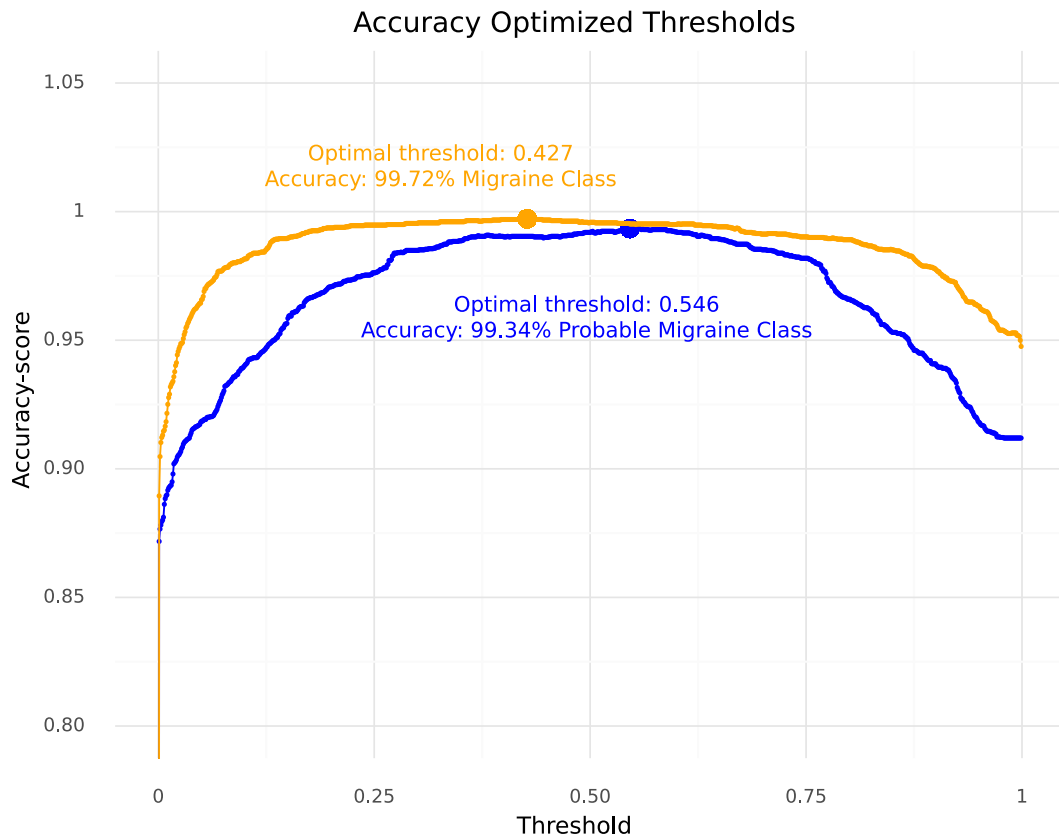
**Tabla 10.** *Regresión Logística Multinomial - Métricas promedio con umbrales óptimos para la precisión accuracy.*

Metric	Probable Migraine	Migraine
Precision	99.19 %	99.58 %
	Default Umbral	Default Umbral
	<b>0.5</b>	<b>0.5</b>
Precision Optimizada	99.34 %	99.72 %
	Opt. Umbral	Opt. Umbral
	<b>0.546</b>	<b>0.427</b>

*Nota.* Método paso a paso para obtener los umbrales óptimos promedio que permiten una precisión más elevada para las clases minoritarias *Probable Migraña* y *Migraña*.

La siguiente representación gráfica de la Figura 18, utilizando la librería **ggplot2** (Wickham, 2016), con código fuente disponible en [A.2 Notebooks y repositorios](#), permite seguir claramente la evolución de ambas métricas a lo largo de los diferentes *umbrales*, 1000 pasos desde 0->1:

**Figura 18.** Evolución de los umbrales óptimos para la precisión precisión de las clases minoritarias.



*Nota.* Para obtener los umbrales óptimos para la precisión se ha utilizado un paso de **0.001** y una validación cruzada de los datos de **16**.

#### 4.3.3. Conclusiones de lo métodos utilizados para optimizar el rendimiento del modelo

Después de los métodos utilizados para optimizar el rendimiento del modelo de regresión logística *multinomial* utilizado, se han obtenido las siguientes conclusiones para mejorarlo:

- Añadir el parámetro ***class\_weight='balanced'*** para tratar de dar más importancia o peso a las clases minoritarias respecto a la mayoritaria, **ha quedado descartado** por su pérdida de rendimiento en la clase *Probable Migraña* por un excesivo desplazamiento del límite de decisión hacia esta clase respecto a la mayoritaria *No Migraña*.
- Métodos para obtener el **umbral óptimo**, con estos métodos de optimización es factible mejorar el rendimiento del modelo, dependiendo del método utilizado, curvas **ROC**,



curvas de **recuperación de precisión**, métodos **paso a paso**, etc, se pueden mejorar las diferentes métricas según el criterio del desarrollador sobre qué métricas se van a priorizar. El objetivo que ha de decidirse a priori es cuales métricas se quieren optimizar y a partir de ahí decidir el método más conveniente en cada caso.

Relacionado con la optimización de los umbrales y cuales métricas se desean optimizar para el conjunto de datos utilizado se han obtenido las siguientes conclusiones:

1. Que si se desea optimizar la media armónica **F1** se pueden usar las curvas tanto de **recuperación de precisión**, donde el umbral obtenido corresponde al máximo del rango donde es óptimo, como utilizando el método de ajuste óptimo **paso a paso** donde el umbral obtenido corresponde al mínimo del rango donde es óptimo.

En este punto, para la optimización de la **F1**, hay que tener en cuenta que en este trabajo se han calculado las métricas para una fracción de datos de entrenamiento y de test en proporción 80/20, siempre la misma, así que si este es el objetivo elegido para intentar optimizar el *umbral* habría que distribuir los datos de entrenamiento y de test en un número distribuido de fragmentos mucho mayor a 1, para garantizar que no haya sobre ajuste y que se obtienen los umbrales promedio de todos los fragmentos distribuidos.

2. Aprovechando que el tamaño del conjunto de datos no es muy elevado, para evitar un exceso de carga computacional, se ha planteado el cálculo de los umbrales óptimos promedio de la clases minoritarias paso a paso utilizando una validación cruzada de 16 bloques distribuidos de datos de entrenamiento y test (80/20) para mejorar la precisión obteniendo los umbrales óptimos para ambas clases con el objetivo de que puedan ser usados en el proceso de predicción de las migrañas por medio de la aplicación **ADMTF**.

#### 4.4. Implementación del modelo de aprendizaje con **Flask**

A lo largo de los apartados anteriores 4.2 y 4.3 se ha verificado que el modelo de aprendizaje seleccionado de **regresión logística multinominal** tiene un buen rendimiento, comprobando los buenos valores de las diferentes métricas, como son la *precisión*, el *F1*, etc a lo largo de las diferentes fracciones de datos en los procesos de validación cruzada. Una vez elegido este modelo, llega el momento del empaquetado y posterior uso del mismo para que pueda ser usado posteriormente, por ejemplo, desde la aplicación web **ADMTF** desarrollada en este trabajo y basada en *Flask* (*Pallets/Flask*, 2021).

#### 4.4.1. Guardar y empaquetar el modelo

La validación cruzada de los datos es un procedimiento para estimar el rendimiento de un modelo concreto, en este caso el de **regresión logística multinomial**, con el objetivo de poder comparar su efectividad con otros modelos llegado el caso, y no para generar el modelo definitivo en sí.

Por lo tanto, para que los **pesos** obtenidos sean calculados sobre el mayor número de datos disponibles y habiendo comprobado la efectividad del modelo, además de los umbrales, para lograr una mayor precisión, se realiza el cálculo definitivo de los pesos del mismo con el total de los datos disponibles (Brownlee, 2017), (Bogdanovist, s.f.).

Para guardarlo se usa la librería de *python* **Joblib** (Joblib Development Team, 2021) como el siguiente fragmento Listado 4.11 donde el modelo queda empaquetado en el fichero `'ADMTF_LR_Model.pkl'` para uso en la aplicación **ADMTF**.

Fragmento de código 4.11. Empaquetado del modelo utilizando *Joblib Accuracy*.

```

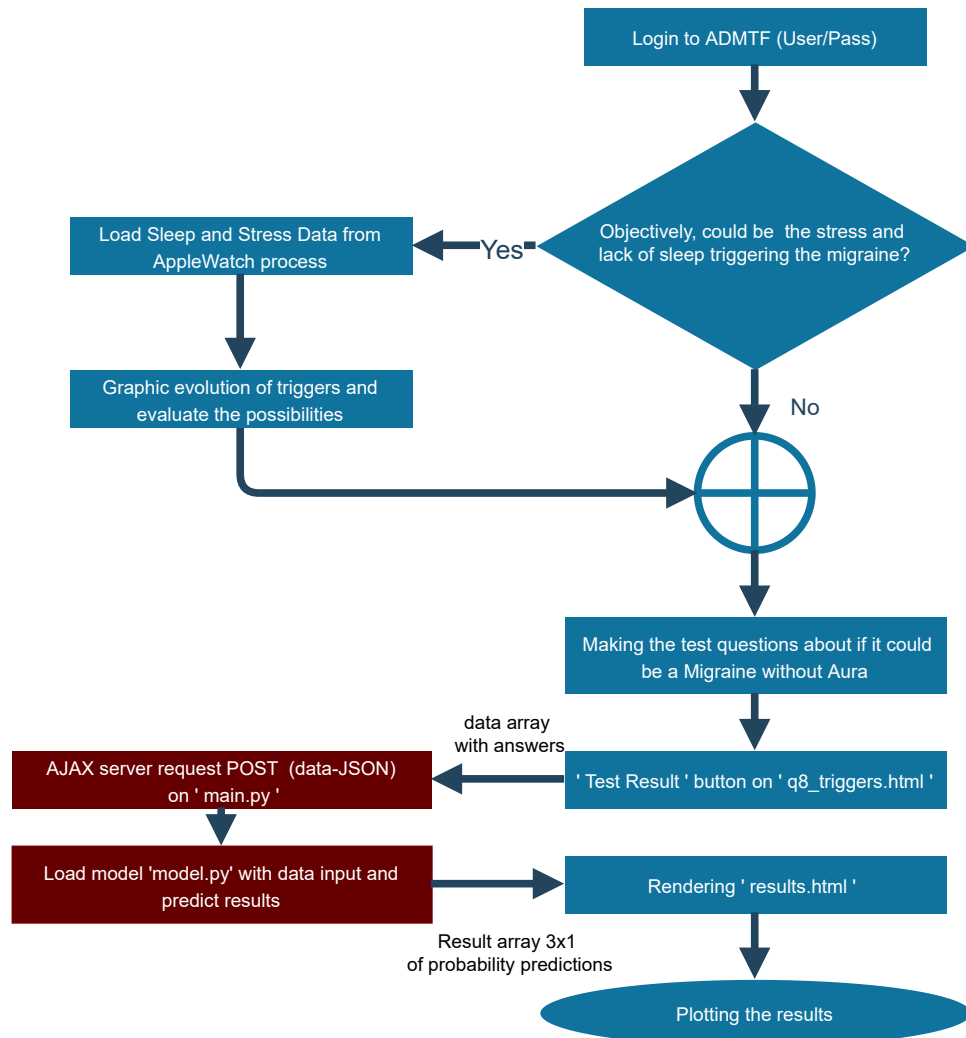
1  ...
2  import joblib
3
4  # Recalculate the model for all dataset before save it
5  LRRegression_Multi_Model.fit(X, y.values.ravel() )
6  # returns a matrix of weights (coefficients)
7  print ("Matrix of weights (Logistic regression multinomial model coefficients):\r\n")
8  print(LRRegression_Multi_Model.coef_)
9
10 # Save RL_Model to file in the current working directory
11 joblib_file = 'ADMTF_LR_Model.pkl'
12 # Save model as pickle file in the same directory
13 joblib.dump(LRRegression_Multi_Model, joblib_file)
14 ....
15 Matrix of weights (Logistic regression multinomial model coefficients):
16 [[-1.91701452e+00 -2.16310037e+00 -1.83165497e+00 -2.28469308e+00
17    -5.64104423e+00 -4.37362270e+00 -5.92449412e+00 -2.77966801e-01
18    -2.55758603e-01 -8.92914450e-02 -5.95218466e-01  2.13298765e-01
19    -3.34891466e-02 -2.00368207e-01  2.28501143e-01  1.40074345e-02
20     1.62231034e-01  1.87898422e-01 -1.17247013e-01 -3.79566384e-01
21     1.67867204e-01 -6.16831827e-02  3.14407945e-01  7.07372694e-02
22    -1.69584226e-01  9.68507250e-04  2.44079028e-01]
23 [ 4.67823413e-01  3.22401463e-01  5.42992903e-01  8.08852319e-01
24     1.85090235e-01  2.25608594e-02  4.41891694e-01  1.34423416e-01
25     6.78311168e-02 -1.02713703e-01  3.38378832e-01 -3.94458734e-01
26     3.24569672e-01 -6.22078282e-02 -1.05290115e-01  9.26386378e-02
27    -1.61077786e-01 -3.94728894e-01 -1.17827327e-01  1.48380864e-01
28     4.58689637e-02 -2.50016367e-01  3.07082429e-01  2.65952513e-01
29    -7.61380298e-02 -1.94015249e-01 -4.26172445e-01]
30 [ 1.44919111e+00  1.84069891e+00  1.28866207e+00  1.47584076e+00
31     5.45595399e+00  4.35106184e+00  5.48260243e+00  1.43543385e-01
32     1.87927487e-01  1.92005148e-01  2.56839634e-01  1.81159969e-01
33    -2.91080525e-01  2.62576035e-01 -1.23211027e-01 -1.06646072e-01
34    -1.15324819e-03  2.06830472e-01  2.35074340e-01  2.31185520e-01
35    -2.13736168e-01  3.11699550e-01 -6.21490375e-01 -3.36689782e-01
36     2.45722256e-01  1.93046741e-01  1.82093416e-01]]
37 ['ADMTF_LR_Model.pkl']
38 ...

```

#### 4.4.2. Cargar el modelo en la aplicación ADMTF (Flask)

El procedimiento seguido para cargar el modelo (bloques resaltados) en la aplicación *ADMTF* es el siguiente Figura 19:

Figura 19. Diagrama funcional de la aplicación *ADMTF* enfocado en la carga del modelo de aprendizaje.



*Nota.* Resaltados los bloques donde se hace la solicitud al servidor para que cargue el modelo, haga la predicción en *model.py* y devuelva la página de resultados *results.html* con los datos para ser mostrados gráficamente.

- Una vez lanzada la aplicación y se ha dado de alta el usuario en la base de datos por medio del proceso de inicio de sesión (*user + password*), se ejecuta el procedimiento de diagnóstico de la *migraña sin aura* como queda detallado en el apartado 4.6 respondiendo a las diferentes cuestiones planteadas e incluso verificando, si es necesaria la consulta por parte del usuario de si el estrés o la carencia de sueño han podido ser factores desencadenantes.
- Cuando se alcanza el final del test, pulsando el botón *Test Result*, lo que hace la aplicación es cargar el *html* resultado *results.html* a través de una solicitud AJAX mediante el método

POST con los datos del test como datos de entrada.

- En ese proceso los resultados de las preguntas del test se pasan en formato JSON a la aplicación principal *main.py*, método *run\_classifier()*, donde lo que se hace es básicamente es ejecutar la función *model.api\_predict(data\_json)*, en este método almacenado en el fichero *model.py* se carga el modelo utilizando la librería *JobLib* y se le pasan como parámetros de vuelta el vector de 3 números con la probabilidad de sea *No Migraña*, *Probable Migraña* y *Migraña* donde la suma de las 3 probabilidades es 1.
- Por último con el vector de resultados se obtiene la representación gráfica para visualizar los resultados.

#### 4.5. Procesar los datos de salud del *Apple Watch*

Parte del objetivo de este trabajo como se indica en 3.1 [Objetivos Generales](#) es tratar de determinar si dos de los desencadenantes naturales más prevalentes en el origen de las *migrañas sin aura*, como son, el **estrés** y la **carencia de sueño** pueden causarla analizando una serie de datos obtenidos de un reloj comercial de muñeca como es el *Apple Watch*, el cual es bastante más asequible respecto a otros métodos utilizados tradicionalmente u otros relojes de muñeca más específicos, sofisticados y con coste económico mayor como el *Empatica E4*.

Los 2 parámetros utilizadas en este trabajo y que proporciona el reloj son:

- **Variabilidad del ritmo cardiaco (HRV)**, específicamente, es la métrica en el dominio temporal que ofrece el Apple Watch desde sus datos de salud, y corresponde a la *desviación estándar de los intervalos R-R normal a normal (SDNN)* (Shaffer y Ginsberg, 2017) ofrecido en milésimas de segundo.

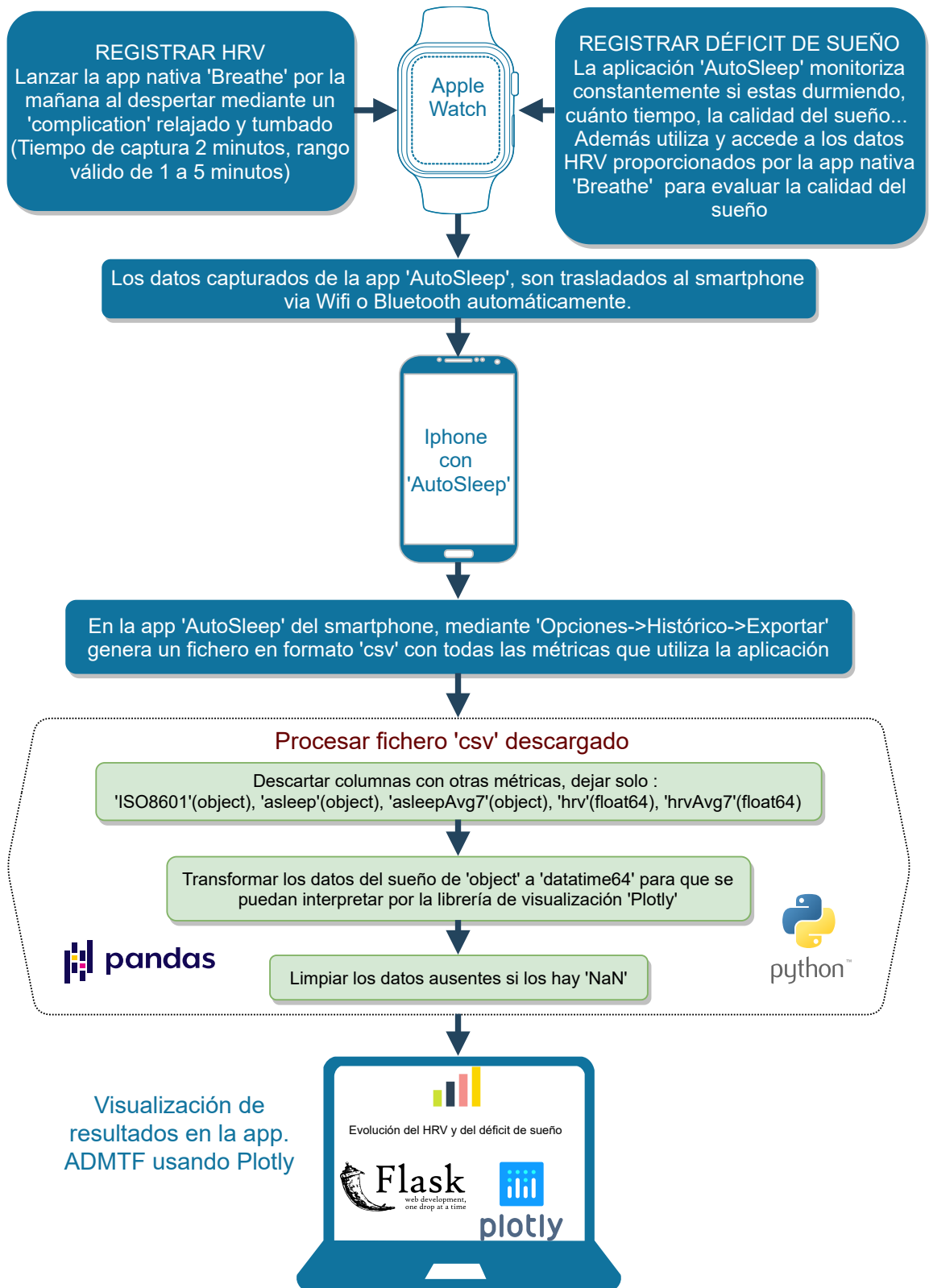
Ha quedado demostrada su relación con el **estrés** en capítulo 2 [Estado del Arte](#), y como se describe en (Castaldo et al., 2015) puede apreciarse claramente la bajada en la variabilidad del *HRV* cuando los individuos están sometidos a algún tipo de estrés.

- **Tiempo de sueño efectivo**, registro y evolución de las horas de sueño efectivo en formato (hh:mm).

El déficit de sueño es uno de los desencadenantes naturales de mayor prevalencia (J. Kim et al., 2017) (Houle et al., 2012) en la *migrañas*.

La estructura y aplicaciones para obtener los datos del *HRV* y el *déficit o carencia de sueño*, se describe de manera esquematizada en la Figura 20.

**Figura 20.** Esquema de la metodología seguida para procesar los datos del *Apple Watch*.



*Nota.* Esquema del proceso llevado a cabo para obtener las métricas que permiten al usuario comprender de manera simple como el **estrés** o el **déficit de sueño** han evolucionado mediante la captura de datos del *Apple Watch*.

La metodología seguida es la siguiente:

#### 4.5.1. Registrar el *HRV* mediante la aplicación nativa **Respirar**

La calidad de la medida en sí del *Apple Watch* ha sido contrastada en algunos estudios como (Hernando et al., 2018), donde se concluye que la comparativa de las métricas obtenidas son bastante precisas respecto a las de otros dispositivos de referencia, como en este caso es la combinación *Bland-Altman / Polar H7*.

No obstante, para que la medida del *HRV* en sí sea adecuada a cada usuario, hay que seguir una serie de pautas como son:

- Al contrario de lo que puede parecer más lógico, una menor variación del intervalo entre latidos, es decir un comportamiento más estable entre latidos implica una mayor probabilidad de estar expuesto a situaciones de *estrés*.
  - Como indican en (H.-G. Kim et al., 2018), el valor de *HRV* que para un individuo indica una situación de *estrés* no tiene porque indicar lo mismo en otro individuo con el mismo valor obtenido, ya que intervienen numerosos factores como la edad, el estado físico, el sexo, factores genéticos, el momento, etc que llevan a realizar un estudio particularizado a cada individuo.
  - Por lo tanto, como indican los autores del (*AutoSleep. Sleep Tracking, s.f.*) un buen método para obtener unos adecuados valores de *HRV* particularizados a cada individuo sería el siguiente método:
    - Al despertar, con el reloj obviamente colocado, y antes de incorporarse lanzar la aplicación de salud integrada en el *Apple Watch Respirar* por un minuto y relajarse, sin realizar movimientos bruscos, y realizando los ejercicios de respiración indicados por la aplicación mediante pequeñas vibraciones. Con este método, se logra tomar los datos de *HRV* de la misma manera todos los días intentando eliminar factores que puedan distorsionar el valor como el movimiento, etc.
    - Para facilitar el lanzamiento de la aplicación *Respirar*, al despertar, es interesante la utilización de lo que Apple denomina *Complications*, así se facilita mucho el lanzamiento de la aplicación.
- Capturar el *HRV*, cuya métrica hay que recordar que es el **SDNN** en el *Apple Watch*, se ve influenciada tanto por el *estrés mental* como por el *estrés físico* que es el no deseado en este caso, de ahí la importancia de seguir este procedimiento o rutina matinal para lograr unos valores adecuados y más relacionados al *estrés mental*.

- Así pues, con este método sencillo en unos cuantos días se pueden tener unos datos del HRV en **milésimas de segundo** muy adecuados y adaptados a cada individuo, donde valores elevados respecto al promedio indicaran días buenos para combatir el *estrés* y viceversa.

Estos datos de *HRV* son capturados por la aplicación **AutoSleep** desde la base de datos interna de salud del *Apple Watch*, que es donde los va volcando la aplicación **Respirar** obteniendo no solo el '*hrv*' al despertar sino también el promedio de evolución de la última semana '*hrvAvg7*'.

#### 4.5.2. Registrar el déficit o carencia de sueño

De las numerosas métricas que obtiene el *Apple Watch*, como son las fases de sueño, sueño ligero, profundo, etc, de manera automática, la aplicación **AutoSleep** captura muchas relacionadas con el sueño, en este trabajo, solo se utiliza el tiempo de sueño real que extrae con mucha precisión ya que incluso utiliza los datos del pulso cardiaco para determinar si el individuo está realmente dormido o no.

De aquí se extraen las 2 métricas utilizadas en este trabajo relacionadas con el **déficit del sueño** como son el '*asleep*' o número de horas de sueño y el promedio de evolución de la última semana '*asleepAvg7*'.

#### 4.5.3. Transferencia de datos desde el *Apple Watch* al smartphone *Iphone*

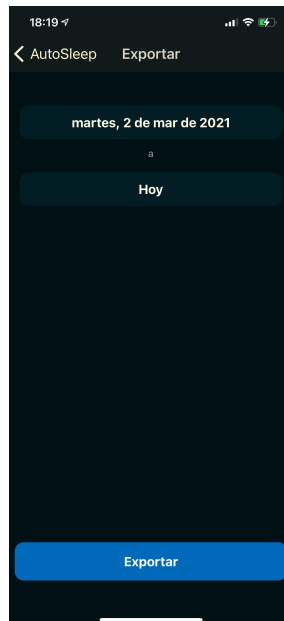
Periódicamente vía Wifi / Bluetooth se van transmitiendo estos datos a la aplicación cliente **AutoSleep** instalada en el *smartphone*. De esta manera los datos quedan almacenados y disponibles cada día en el *smartphone* para su análisis en el momento que el usuario lo necesite.

#### 4.5.4. Descarga y procesado de datos en formato csv

Cada vez que el usuario quiere actualizar los datos capturados en el *Apple Watch* y cargarlos en la aplicación creada para el diagnóstico de las *migrañas sin aura*, denominada **ADMTF**, se han de realizar las siguientes operaciones:

- Accediendo a la aplicación **AutoSleep** desde el *smartphone* mediante **Opciones > Histórico > Exportar** se accede a una pantalla como la de la [Figura 21](#) y se genera un fichero en formato **csv** con el intervalo de días con el que se desea trabajar.

**Figura 21.** Descarga de fichero de datos con la métricas de sueño y HRV desde la aplicación *AutoSleep*.



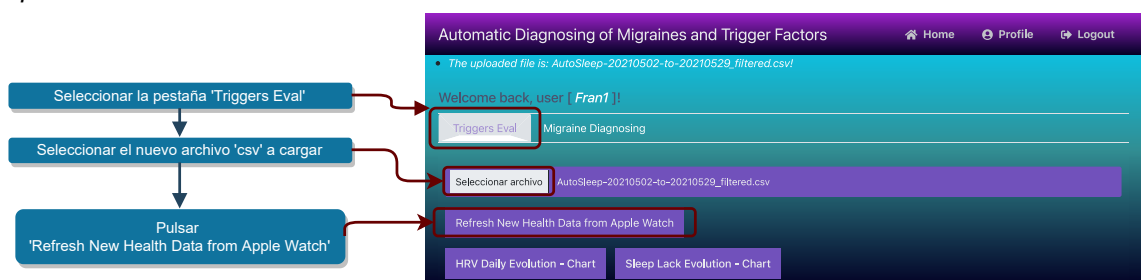
*Nota.* Imagen extraída del smartphone *iPhone*.

- Cargar el fichero en la aplicación **ADMTF**, para ello, y una vez el usuario ha accedido a la aplicación con su nombre de usuario y contraseña, se selecciona la pestaña **Triggers Eval**.

En esa situación, se puede decidir lanzar las representaciones gráficas que permiten evaluar los datos anteriores, seleccionando el botón **HRV Daily Evolution - Chart** para ver la evolución de la variabilidad del pulso cardiaco, el gráfico de evolución del sueño mediante el botón **Sleep Lack Evolution - Chart** o incluso cargar datos nuevos pulsando el botón de **Seleccionar archivo** para cargar el fichero **csv** descargado de la aplicación **AutoSleep** con los datos actualizados y al pulsar **Refresh New Health Data from Apple Watch** se cargan nuevos datos en la aplicación para su análisis.

Esto proceso esquematizado queda reflejado en la siguiente [Figura 22](#).

**Figura 22.** Procedimiento de carga de más datos de salud procedentes del *Apple Watch* en la aplicación **ADMTF**.





- La aplicación **ADMTF** procesa los datos capturados, descartando las columnas con las métricas no utilizadas, dejando solamente las siguientes:

```
'S08601' (object), 'asleep' (object), 'asleepAvg7' (object)

'hrv' (float64), 'hrvAvg7' (float64)
```

Además, para que la librería de visualización utilizada en la aplicación, que en este caso es **Plotly** (Inc., 2015), interprete correctamente las métricas de tiempo, el tipo de dato original (**object**) es necesario convertirlo a **datetime64** para que sea legible por la librería de visualización. Por último se descartan los datos nulos **NaN** que puedan haber.

A continuación, en el [Listado 4.12](#) un fragmento de código del proceso descrito.

#### Fragmento de código 4.12. Procesado de los datos del *Apple Watch* en la aplicación **ADMTF**.

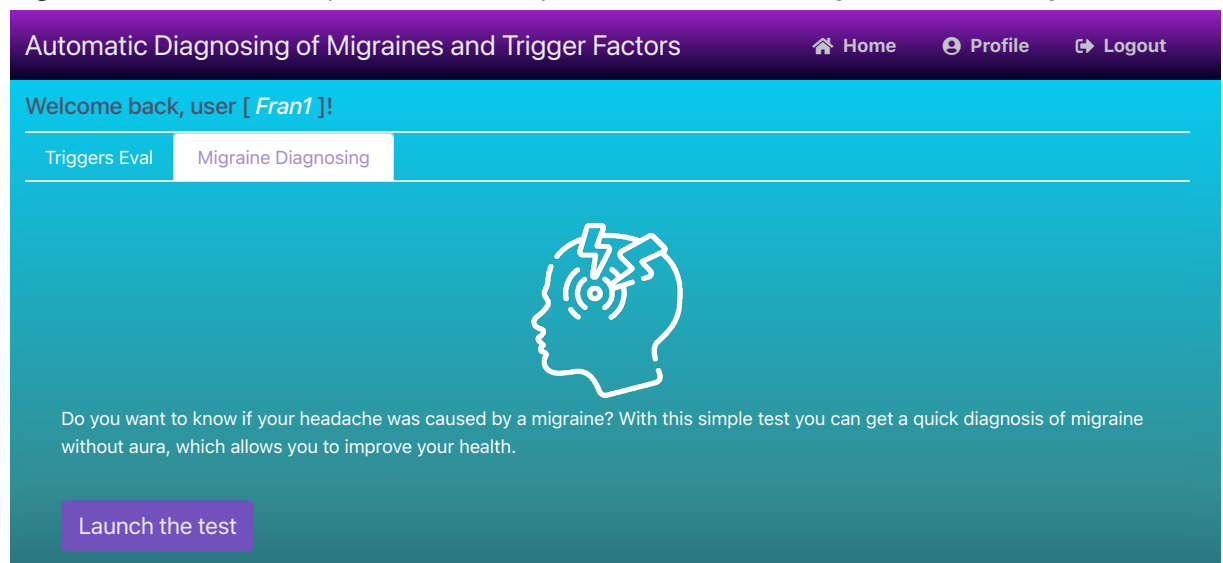
```
1 # http://localhost:5000/ADMTF/upload - this will be the home page, only accessible for loggedin users
2 @app.route('/ADMTF/upload', methods=['GET', 'POST'])
3 def upload():
4     # Check if user is loggedin
5     if loggedin():
6         if request.method == 'POST':
7             file = request.files['csvfile']
8             dframe = pd.read_csv(file, skipinitialspace=True, usecols=attributes_selected)
9             dframe[['asleep', 'asleepAvg7']] = dframe[['asleep', 'asleepAvg7']].apply(pd.to_datetime)
10            dframe['datetime'] = dframe['S08601'].apply(lambda x: datetime.strptime(x, '%Y-%m-%dT%H:%M:%S%z'))
11            if not os.path.isdir('uploads'):
12                os.mkdir('uploads')
13            dframe.to_csv('./uploads/AutoSleep-filtered.csv', encoding='utf-8', index = False)
14            flash('The uploaded file is: {}'.format(file.filename))
15            # redirect to home page after upload csv file
16            return redirect(url_for('home'))
17
18 @app.route('/ADMTF/chart-hrv-daily')
19 def chart1():
20     dframe = pd.read_csv('./uploads/AutoSleep-filtered.csv', skipinitialspace=True)
21     dropna_dframe = dframe.dropna(subset=['hrv', 'hrvAvg7'])
22     fig = px.line(dropna_dframe, x='datetime', y=['hrv', 'hrvAvg7'], title='Heart Rate Variability plot',
23                 labels={'value': 'HRV (ms)', 'datetime': 'Days register', 'variable': 'HRV vars'}, template='
24                 simple_white')
25
26     #template='plotly_dark')
27
28     fig.update_layout(xaxis=dict(zeroline=False, showgrid=False))
29     fig.update_layout(yaxis=dict(zeroline=False, showgrid=False))
30
31     graphJSON = json.dumps(fig, cls=plotly.utils.PlotlyJSONEncoder)
32     header="HRV daily and 7 days average evolution"
33     description = """
34     A decrease in the average values may indicate a greater predisposition to stress.
35     The values should always be taken when waking up, being relaxed and lying down, so the data will be adequate for
36     a correct analysis.
37     """
38     return render_template('trigger-graph-json.html', graphJSON=graphJSON, header=header, description=description)
```

- Visualización de los resultados de las métricas capturadas en la aplicación **ADMTF** mediante la librería **Plotly**, los cuales han sido detallados en [5.1 Resultados obtenidos del \*Apple Watch\*](#).

## 4.6. Diagnóstico de la *migraña sin aura*

Desde la aplicación principal **ADMTF** realizada en *Flask* y una vez, esto último a elección del usuario, se han analizado más en profundidad los desencadenantes de las migrañas mediante el análisis de los datos obtenidos desde el *Apple Watch*, que son el **estrés** y la **ausencia de sueño**. El usuario de la aplicación tiene la mayor información posible para lanzar desde la pestaña **Migraine Diagnosing** el test para el diagnóstico de la **migraña sin aura** como se muestra en la Figura 23.

**Figura 23.** Ventana de la aplicación **ADMTF** para lanzar el test de diagnóstico de las migrañas.



En el desarrollo del test se van realizando una serie de preguntas, desde la 1 a la 7, que siguen los criterios de clasificación de las migrañas, que como se ha detallado anteriormente, son los criterios utilizados por la organización **ICHD-3** correspondientes al apartado **1.1 Migraine without aura**, las cuales son:

### 1. ¿El dolor de cabeza ha durado entre 4 y 72 horas aproximadamente?

**Nota 1.** Cuando el paciente que sufre la migraña duerme durante el ataque y se despierta si él, **ICHD-3** considera que el tiempo de duración de la migrañas debe contabilizarse hasta el momento de despertar.

**Nota 2.** En niños y adolescentes (menores de 18 años), los ataques pueden durar de 2 a 72 horas (la evidencia de duraciones no tratadas de menos de dos horas en niños no se ha comprobado).

### 2. ¿El dolor es unilateral, es decir, se ubica en un lado de la cabeza?

**Nota 1.** El dolor suele ser unilateral, pero puede ser bilateral, con mayor frecuencia en

*una distribución frontal-temporal.*

3. **¿El dolor es pulsante o no?**
4. **En una escala de 1 a 3 para la intensidad del dolor (bajo, moderado o severo), ¿Considera que el nivel de dolor puede ser moderado o severo?**
5. **¿Podría aumentar el dolor debido a la actividad física de rutina (por ejemplo, caminar o subir escaleras)?**
6. **¿Ha tenido náuseas o vómitos?**
7. **¿Ha sufrido fotofobia (exceso de sensibilidad a la luz) y fonofobia (aversión a los sonidos fuertes)?**
8. **¿Ha notado que uno o varios de los siguientes factores desencadenantes le han causado dolor de cabeza?**

***Estrés**, Dormir demasiado, **Dormir poco**, Ejercicio excesivo, Ningún ejercicio, Fatiga, Cambios hormonales, Cambios emocionales, Cambios de tiempo, Excesiva luz del sol, Luz inapropiada, Ruido excesivo, Olores, Comidas irregulares, Calor excesivo, Cafeína, Fumar, Beber alcohol, Comer chocolate o queso, Viajar.*

En las figuras [Figura 24](#) correspondiente a la primera pregunta, y [Figura 25](#) correspondiente a la séptima, se puede visualizar el estilo de las ventanas de la aplicación, con el objetivo de enfocar al usuario en seleccionar la respuesta correspondiente para poder realizar el test de manera rápida y sin distracciones:

**Figura 24.** Ventana de la aplicación **ADMTF** correspondiente a la primera de las preguntas del test.

The screenshot shows the application interface for "Automatic Diagnosing of Migraines and Trigger Factors". The header is purple with navigation links for Home, Profile, and Logout. The main content area has a blue gradient background and displays "Question 1 of 8". The question text is: "Has the Headache attack lasted between 4–72 hours?" followed by explanatory text: "When the patient falls asleep during a migraine attack and wakes up without it, duration of the attack is reckoned until the time of awakening. In children and adolescents (aged under 18 years), attacks may last 2–72 hours (the evidence for untreated durations of less than two hours in children has not been substantiated)." Below the text are two buttons: "Yes" (orange) and "No" (white with blue border). A "Next" button is located below the response buttons. At the bottom left, it says "Questions updated from [ichd-3.org](http://ichd-3.org)".

**Figura 25.** Ventana de la aplicación **ADMTF** correspondiente a la séptima de las preguntas del test.

The screenshot shows the application interface for "Automatic Diagnosing of Migraines and Trigger Factors". The header is purple with navigation links for Home, Profile, and Logout. The main content area has a blue gradient background and displays "Question 7 of 8". The question text is: "Have you suffered from photophobia (excess of light sensitivity) and phonophobia (aversion to loud sounds)?" Below the text are two buttons: "Yes" (white with blue border) and "No" (orange). A "Next" button is located below the response buttons. At the bottom left, it says "Questions updated from [ichd-3.org](http://ichd-3.org)".

Previa a la visualización de los resultados del test que pueden consultarse en el 5.2, la última ventana de preguntas, es decir la octava Figura 26, engloba los desencadenantes más habituales de las migrañas, como se ha desarrollado en 2 Estado del Arte.

Si el usuario ha seleccionado el estudio en la parte de la aplicación correspondiente a los datos capturados del *Apple Watch* puede decidir con mayor seguridad la selección de las casillas de verificación correspondientes al **estrés** o la **carencia de sueño**, los demás desencadenantes quedan más a la elección del propio encuestado y en un futuro puedan ser analizados con un mayor rigor.

**Figura 26.** Ventana de la aplicación *ADMTF* correspondiente a la octava de las preguntas del test, la relacionada con los desencadenantes más habituales.

The screenshot shows the application interface for 'Automatic Diagnosing of Migraines and Trigger Factors'. At the top, there is a navigation bar with 'Home', 'Profile', and 'Logout' options. The main content area is titled 'Question 8 of 8' and contains the question: 'Have you noticed that one or several of the following triggers have caused your headache?'. Below the question, there are 16 checkboxes arranged in a grid. The selected options are 'Stress Situations', 'No Exercise', and 'Over Heating'. At the bottom of the question area, there is a 'Test Result' button. A footer note states 'Questions updated from [ichd-3.org](http://ichd-3.org)'.

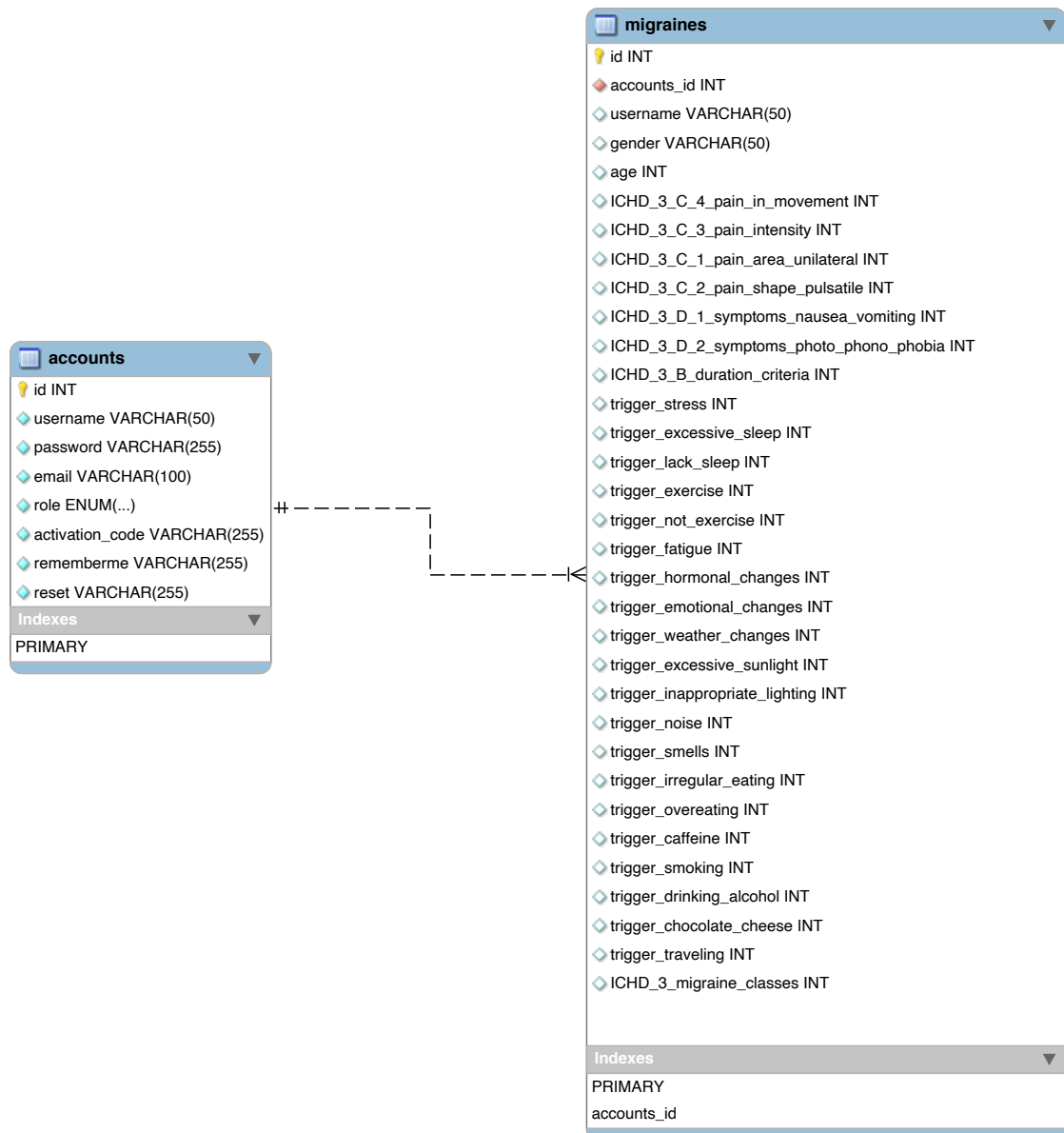
#### 4.7. Almacenar los resultados en *MySQL* para su consulta

Los resultados de los test de la aplicación asociados a cada paciente son almacenados en una base de datos de *MySQL* que consta de 2 tablas principales:

- Tabla **accounts**, donde se almacena la información relacionada con el usuario.
- Tabla **migraines**, donde se almacenan los registros con los resultados de las diferentes pruebas de diagnóstico de la *migraña si aura* asociadas al identificador del usuario.

La estructura con todos los campos, así como la relación de las tablas que conforman la base de datos de la aplicación queda detallada en la [Figura 27](#):

**Figura 27.** Estructura de la base de datos de la aplicación **ADMTF**.



*Nota.* La tabla *migraines* contiene los registros de los resultados de las pruebas de diagnóstico de las migrañas, cada registro queda asociado al usuario de la tabla *accounts* a través del campo *id*.

Después de realiza la predicción, como ha quedado detallado en el diagrama funcional de la aplicación **ADMTF** [Figura 19](#), se realiza la carga de los resultados en la base de datos de *MySQL* previa creación de la página de resultados *results.html* como se muestra en el fragmento [Listado 4.13](#) siguiente.



```

55
56     cursor.execute(sql, (row\username['id'], row\username['accounts_id'], row\username['username'], row\username['
57         gender'], row\username['age'],
58         row\username['ICHD\3\C\4\_pain\_in\_movement'], row\username['ICHD\3\C\3\_pain\_intensity'], row\username['ICHD
59         \3\C\1\_pain\_area\_unilateral'],
60         row\username['ICHD\3\C\2\_pain\_shape\_pulsatile'], row\username['ICHD\3\D\1\_symptoms\_nausea\_vomiting'], row\
61         _username['ICHD\3\D\2\_symptoms\_photo\_phono\_phobia'],
62         row\username['ICHD\3\B\_duration\_criteria'], row\username['trigger\_stress'], row\username['trigger\_excessive\
63         _sleep'], row\username['trigger\_lack\_sleep'],
64         row\username['trigger\_exercise'], row\username['trigger\_not\_exercise'], row\username['trigger\_fatigue'], row\
65         _username['trigger\_hormonal\_changes'],
66         row\username['trigger\_emotional\_changes'], row\username['trigger\_weather\_changes'], row\username['trigger\
67         _excessive\_sunlight'],
68         row\username['trigger\_inappropriate\_lighting'], row\username['trigger\_noise'], row\username['trigger\_smells'],
69         row\username['trigger\_irregular\_eating'],
70         row\username['trigger\_overeating'], row\username['trigger\_caffeine'], row\username['trigger\_smoking'], row\
71         _username['trigger\_drinking\_alcohol'],
72         row\username['trigger\_chocolate\_cheese'], row\username['trigger\_traveling'], row\username['ICHD\3\_migraine\
73         _classes'],))
74
75     # commit new test result to database table
76     mysql.connection.commit()
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

*Nota.* Los registros con los resultados son creados en el fichero principal *main.py* de la aplicación y se almacenan en la tabla *migraines*

La base de datos de la aplicación *ADMTF* se va completando con registros asociados a cada usuario mediante el campo *ID* para que se puedan cargar los datos posteriormente, cuando se desee, filtrando por usuario y de una manera visual e intuitiva como se puede ver en la [Figura 33](#) del apartado [5.3 Resultados obtenidos de la base de datos de la aplicación](#).



## Capítulo 5

# Resultados

### 5.1. Resultados obtenidos del *Apple Watch*

De los datos obtenidos, y procesados como se ha descrito en el capítulo 4 Metodología de trabajo del *Apple Watch* mediante la aplicación de salud (*AutoSleep. Sleep Tracking, s.f.*) se obtienen una serie de características muy interesantes, tanto del *estrés* como de la *carencia de sueño*, seleccionables en un rango de días desde la aplicación instalada en el *iPhone* que está enlazado con el reloj, como ejemplo, en la siguiente [Figura 28](#) se obtiene el marco de datos solo con los campos necesarios para ser cargados posteriormente en la aplicación:

**Figura 28.** Fragmento de datos obtenidos después del procesarlos, destinados al análisis del estrés y la carencia de sueño de la aplicación (*AutoSleep. Sleep Tracking, s.f.*).

ISO8601	asleep	asleepAvg7	hrv	hrvAvg7	datetime
2021-05-02T20:59:59+02:00	2021-06-25 10:05:00	2021-06-25 05:07:30	16.0	19.0	2021-05-02 20:59:59+02:00
2021-05-03T20:59:59+02:00	2021-06-25 08:41:00	2021-06-25 05:50:12	22.0	20.0	2021-05-03 20:59:59+02:00
2021-05-04T20:59:59+02:00	2021-06-25 05:30:00	2021-06-25 05:46:50	18.0	20.0	2021-05-04 20:59:59+02:00
2021-05-05T20:59:59+02:00	2021-06-25 06:15:00	2021-06-25 05:50:51	10.0	18.0	2021-05-05 20:59:59+02:00
2021-05-06T20:59:59+02:00	2021-06-25 04:15:00	2021-06-25 05:48:17	15.0	18.0	2021-05-06 20:59:59+02:00
2021-05-07T20:59:59+02:00	2021-06-25 06:41:00	2021-06-25 06:15:51	14.0	18.0	2021-05-07 20:59:59+02:00
2021-05-08T20:59:59+02:00	2021-06-25 05:15:00	2021-06-25 06:40:17	13.0	15.0	2021-05-08 20:59:59+02:00
2021-05-09T20:59:59+02:00	2021-06-25 07:13:00	2021-06-25 06:15:42	13.0	15.0	2021-05-09 20:59:59+02:00
2021-05-10T20:59:59+02:00	2021-06-25 08:43:00	2021-06-25 06:16:00	16.0	14.0	2021-05-10 20:59:59+02:00
2021-05-11T20:59:59+02:00	2021-06-25 05:30:00	2021-06-25 06:16:00	17.0	14.0	2021-05-11 20:59:59+02:00
2021-05-12T20:59:59+02:00	2021-06-25 05:44:00	2021-06-25 06:11:34	11.0	14.0	2021-05-12 20:59:59+02:00
2021-05-13T20:59:59+02:00	2021-06-25 05:45:00	2021-06-25 06:24:25	21.0	15.0	2021-05-13 20:59:59+02:00
2021-05-14T20:59:59+02:00	2021-06-25 04:14:00	2021-06-25 06:03:25	10.0	15.0	2021-05-14 20:59:59+02:00
2021-05-15T20:59:59+02:00	2021-06-25 05:30:00	2021-06-25 06:05:34	35.0	18.0	2021-05-15 20:59:59+02:00
2021-05-16T20:59:59+02:00	2021-06-25 09:23:00	2021-06-25 06:24:08	29.0	20.0	2021-05-16 20:59:59+02:00
2021-05-17T20:59:59+02:00	2021-06-25 07:00:00	2021-06-25 06:09:25	17.0	20.0	2021-05-17 20:59:59+02:00
2021-05-18T20:59:59+02:00	2021-06-25 04:30:00	2021-06-25 06:00:51	29.0	22.0	2021-05-18 20:59:59+02:00
2021-05-19T20:59:59+02:00	2021-06-25 05:30:00	2021-06-25 05:58:51	14.0	22.0	2021-05-19 20:59:59+02:00
2021-05-20T20:59:59+02:00	2021-06-25 05:12:00	2021-06-25 05:54:08	24.0	22.0	2021-05-20 20:59:59+02:00
2021-05-21T20:59:59+02:00	2021-06-25 04:00:00	2021-06-25 05:52:08	11.0	23.0	2021-05-21 20:59:59+02:00

*Nota.* La tabla contiene los datos filtrados y procesados con las 4 características extraídas, como son el *HRV* diario, el *HRV* promedio semanal, las horas efectivas dormidas y promedio semanal de horas dormidas.

De las columnas con las métricas obtenidas en la [Figura 28](#), la aplicación utiliza para la representación de los resultados las descritas en la [Tabla 11](#):

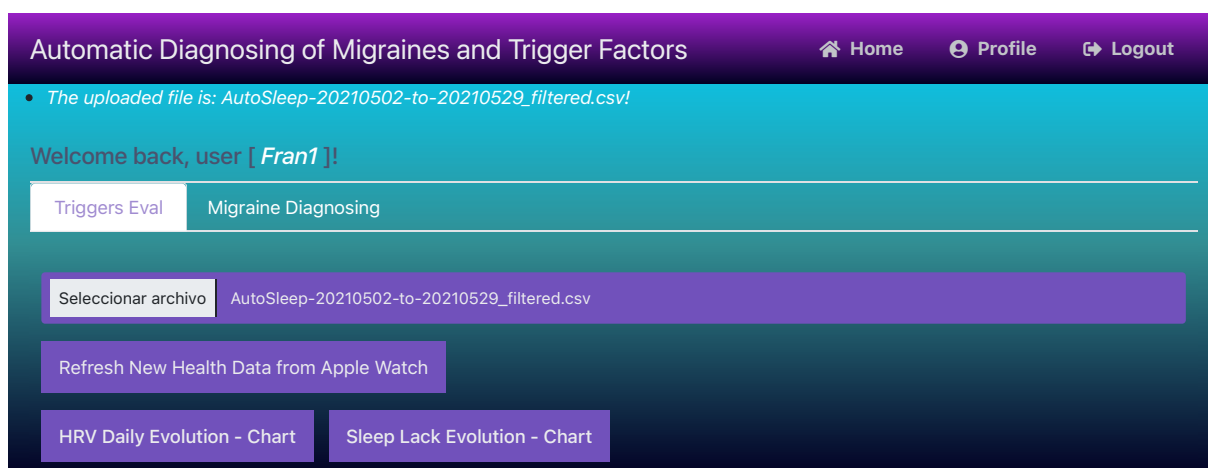
**Tabla 11.** Métricas obtenidas del Apple Watch mediante al aplicación AutoSleep.

<b>Relacionadas con el estrés</b>	
Eje X - Día	Día del registro de los datos en formato <b><i>datetime</i></b> .  La métrica que se extrae del Apple Watch correspondiente a la <i>variabilidad del pulso cardíaco</i> , en este caso es el <b>SDNN</b> , que como se describe en (Shaffer y Ginsberg, 2017), corresponde a la desviación estándar del intervalo entre latidos medido en milésimas de segundo. Donde “normal” indica que se han eliminado los intervalos anormales. El tiempo estimado para la captura es de 5 min, y también es efectivo para intervalos de 1 o 2 minutos.
Eje Y - hrv	Valor promedio de los últimos 7 días del <b>HRV</b> .
<b>Relacionadas con la carencia de sueño</b>	
Eje X - Día	Día del registro de los datos en formato <i>datetime</i> .
Eje Y - asleep	Horas y minutos de sueño efectivo registrados en formato <b>(hh:mm)</b> .
Eje Y - asleepAvg7	Promedio horas y minutos por noche de los últimos 7 días del en formato <b>(hh:mm)</b> .

*Nota.* Descripción de las métricas utilizadas para analizar el estrés y la carencia de sueño extraídos del Apple Watch.

Una vez cargados los datos en la aplicación mediante la pestaña de la [Figura 29](#):

**Figura 29.** Captura de la aplicación en el proceso de cargar de datos para el análisis del estrés y la carencia de sueño.



Se pueden obtener las dos siguientes gráficas, que permiten analizar individualmente la evolución del estrés y la carencia de sueño.

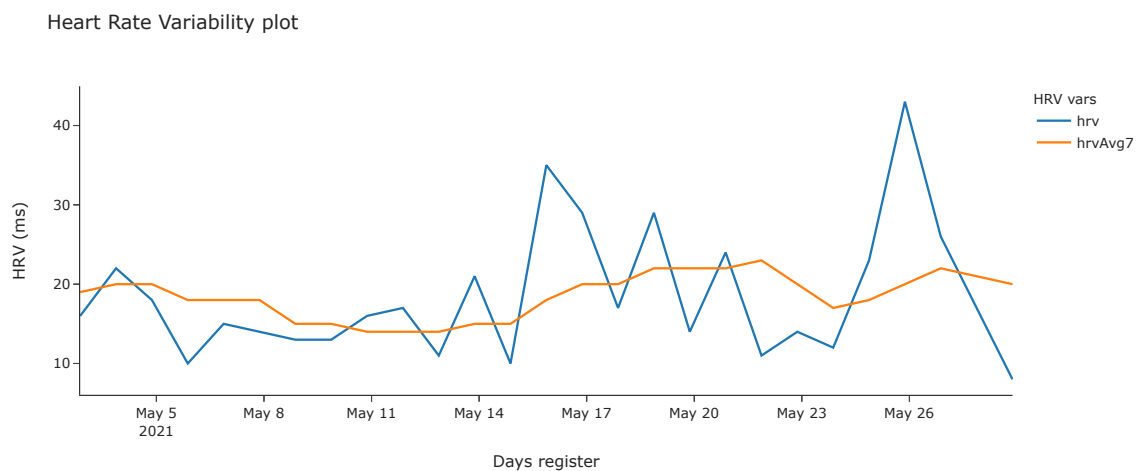
### 5.1.1. Gráfica de la evolución del HRV

Como se puede ver en la [Figura 30](#), a simple vista, el usuario puede considerar que entre el 8 y el 15 de Mayo unos valores promedio relativamente bajos del HRV podrían indicar una mayor predisposición a haber sufrido algún tipo de estrés respecto a otros días.

**Figura 30.** Evolución del HRV obtenida de la aplicación.

## HRV daily and 7 days average evolution

[Return to home page](#)



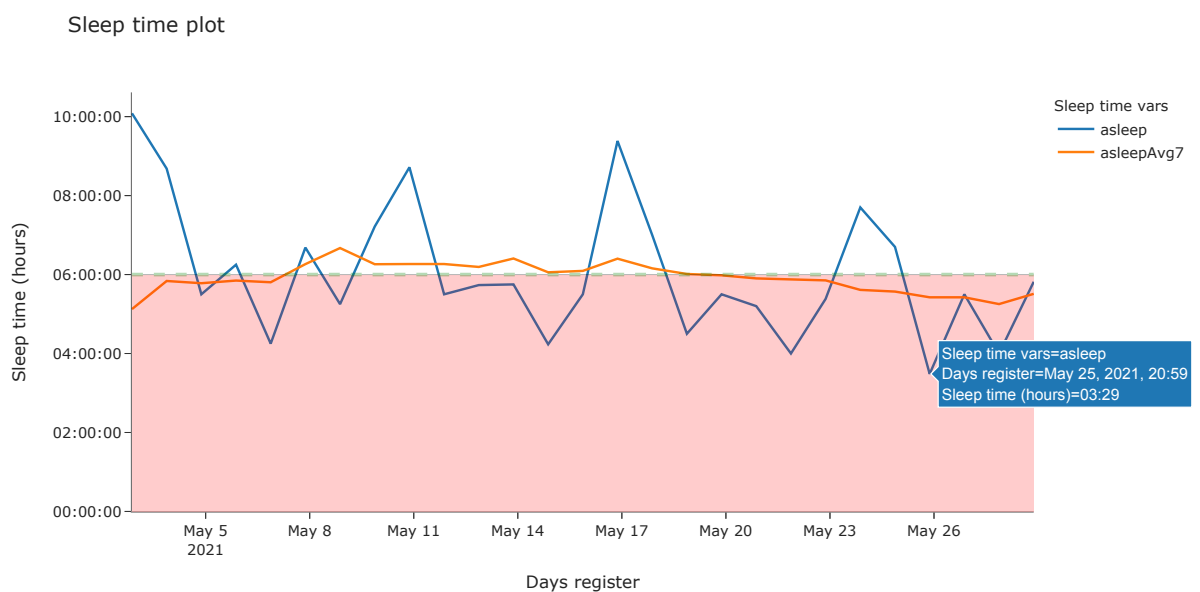
A decrease in the average values may indicate a greater predisposition to stress. The values should always be taken when waking up, being relaxed and lying down, so the data will be adequate for a correct analysis.

### 5.1.2. Gráfica de la evolución de la ausencia de sueño

El **38 %** de las personas que sufren *migrañas* según (Houle et al., 2012) reportan problemas para conciliar el sueño con promedios de **6 horas** o inferiores , de ahí la importancia de que en los resultados obtenidos se fije un umbral visual de **6 horas** que permite al usuario determinar rápidamente que días puede haber problemas debidos a la carencia de sueño.

En la **Figura 31**, el usuario puede comprobar que puede tener un ataque de migrañas con mayor prevalencia a partir del 23 de Mayo donde se ve una evolución negativa en cuanto a las horas necesarias de sueño.

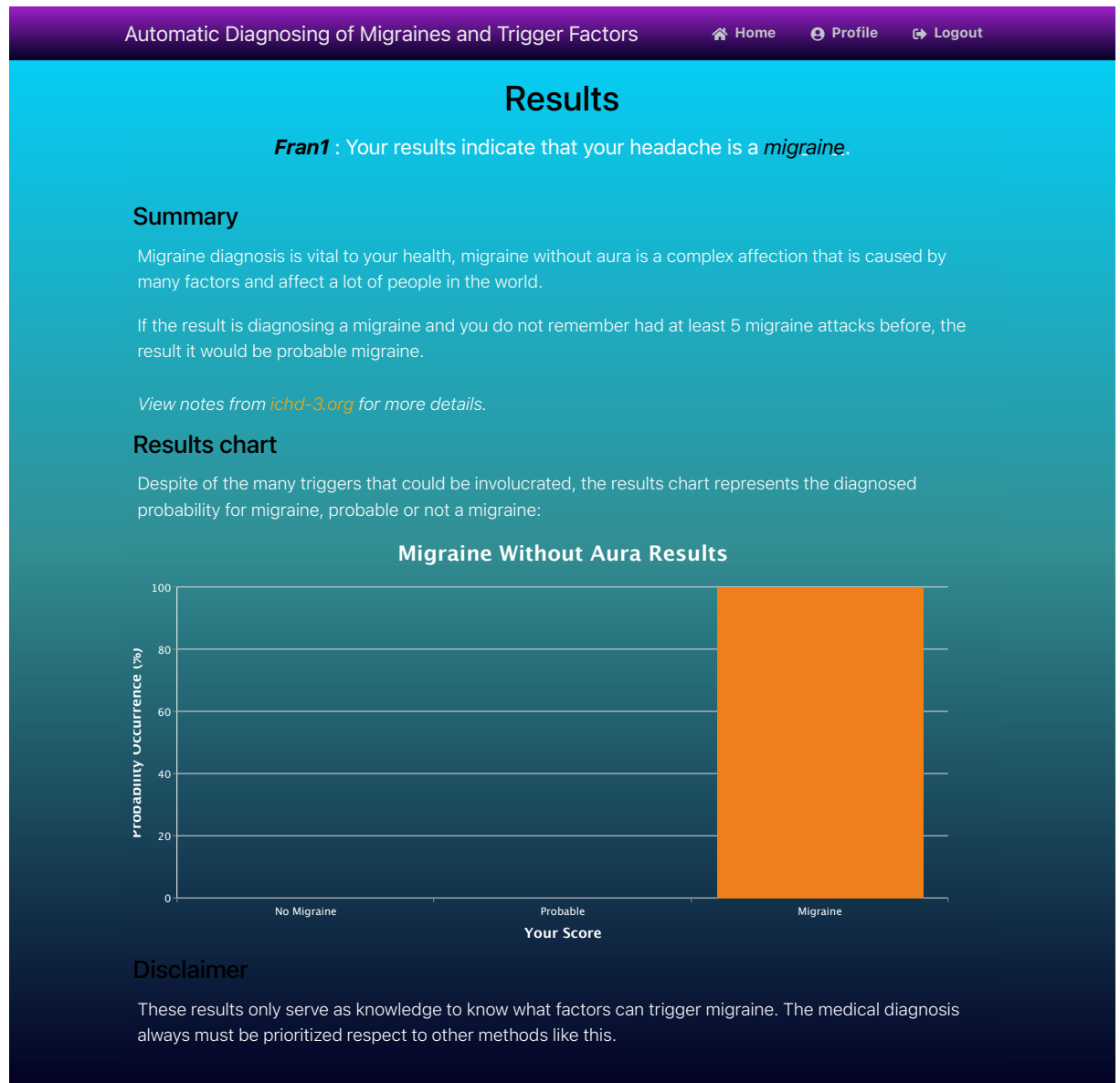
**Figura 31.** Evolución del déficit de sueño obtenida de la aplicación.



## 5.2. Resultados obtenidos de la aplicación

La aplicación **ADMTF**, permite ver los resultados del test de forma inmediata, además graba en la tabla **migraines** de la base de datos *MySql* los resultados de cada test asociados a cada usuario. En la **Figura 32** se muestra un ejemplo de visualización simple del resultado del test en porcentaje de probabilidad de si ha podido ser una migraña o no.

**Figura 32.** Ventana de la aplicación **ADMTF** correspondiente al resultado del test de detección de la migraña sin aura.



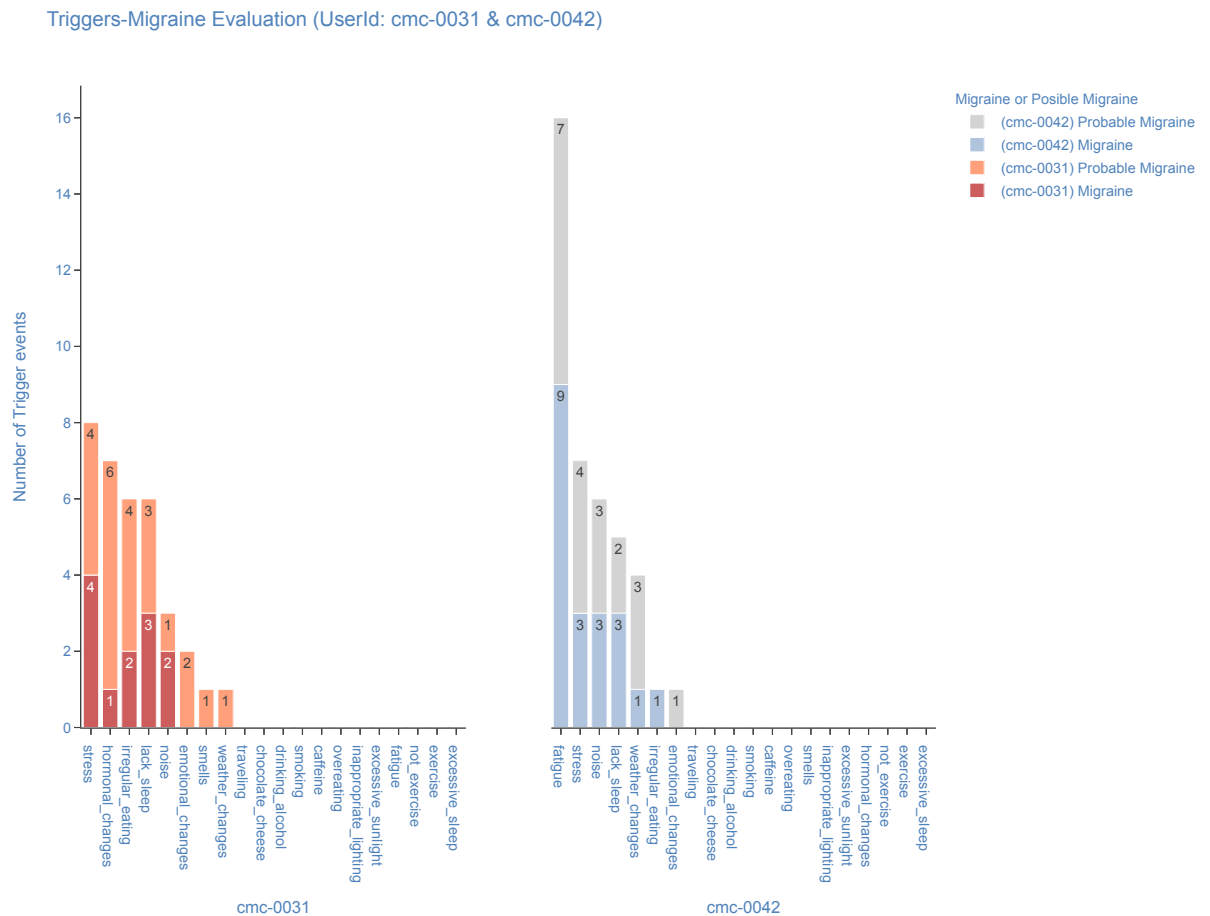
### 5.3. Resultados obtenidos de la base de datos de la aplicación

Los datos de las diferentes sesiones para el diagnóstico de los posibles episodios de *migraña* son almacenados en la base de datos para consultarlos en cualquier momento. Al ser específicos a cada usuario mediante las diferentes sesiones, la base de datos va incrementándose, por consiguiente, mientras los diferentes usuarios la van utilizando para detectar las migrañas se va creando dinámicamente una base de conocimiento que permite obtener los siguientes resultados.

#### 5.3.1. Visualización de los desencadenantes más habituales particularizados al usuario

De los resultados obtenidos de la aplicación, se pueden visualizar rápidamente que desencadenantes tienen la mayor prevalencia para cada usuario. En este caso de la base de datos se extraen los desencadenantes de las diferentes sesiones que corresponden a un resultado positivo, es decir, la aplicación da como resultado un episodio de *migraña sin aura* o de *probable migraña sin aura*.

En la siguiente [Figura 33](#), se muestra un ejemplo de los resultados positivos (migrañas o posibles migrañas) y los desencadenantes con mayor prevalencia obtenidos para los usuarios *cmc-0031* y *cmc-0042*:

**Figura 33. Desencadenantes más habituales de las migrañas para los usuarios cmc-0031 y cmc-0042.**

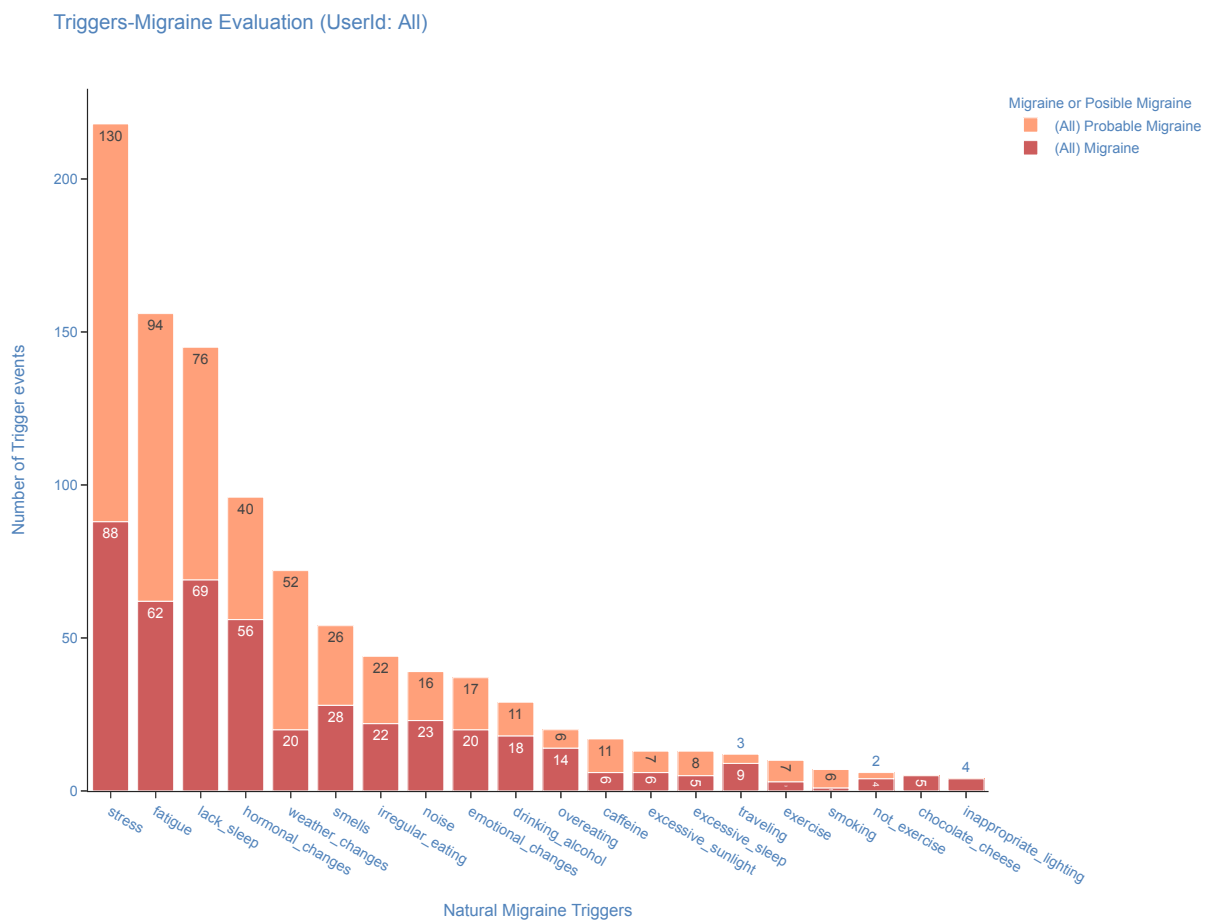
*Nota.* Representación gráfica utilizando la librería *Plotly* (Chris P, s.f.), de dos usuarios de la base de datos, ordenando los desencadenantes de mayor a menor en los episodios con migrañas o posibles migrañas.

Del análisis rápido de la representación gráfica anterior se deduce que para el usuario *cmc-0031* el **estrés** y los **cambios hormonales** son los que tienen una mayor prevalencia frente a otro tipo de desencadenantes. Sin embargo, para el usuario *cmc-0042* es la **fatiga** con diferencia es el principal desencadenante de sus ataques de migraña.

### 5.3.2. Desencadenantes para todos los usuarios

De los resultados anteriores, particulares a cada usuario, es inmediato obtener una visualización global como la de la figura [Figura 34](#), con los desencadenantes *naturales* con más prevalencia para todos los usuarios de la base de datos, que por otra parte, es uno de los objetivos de este trabajo, el ir acumulando datos para conocer que desencadenantes son los de mayor prevalencia.

**Figura 34.** Desencadenantes más habituales de las migrañas para todos los usuarios de la base de datos.



*Nota.* Representación gráfica utilizando la librería *Plotly* (Chris P, s.f.).

Como indica (Marmura, 2018), es evidente el interés de obtener unos resultados que permitan determinar que desencadenantes naturales son los que tienen mayor prevalencia, porque al *ser muy específicos* a cada persona que sufre migrañas, el mal diagnóstico de los mismos, puede provocar un efecto negativo y frustrante en el intento de evitarlos.



## Capítulo 6

# Contribución

Este trabajo, ha tratado de aportar un poco de información que permita mejorar y reducir, si es posible, los episodios de una de las causas de cefaleas de mayor prevalencia en el mundo como es la *migraña*, el hecho de no ser una enfermedad grave, provoca que no se le de la gran importancia que tiene, su impacto social, económico y laboral en el rango de población de edad intermedia.

El problema, ha sido abordado mediante el desarrollo de una aplicación web denominada *ADMTF*, que permite hacer un diagnóstico sencillo del tipo de *migraña* más común, como es la *migraña sin aura* mediante un sencillo cuestionario basado en los criterios de la última versión del clasificador de trastornos de cabeza *ICHD-3* del organismo *The Classification Committee of The International Headache Society*.

La aplicación desarrollada, permite facilitar el diagnóstico rápido de las *migrañas sin aura* y tratar de asociar los episodios, si es posible, a una serie de factores *desencadenantes* específicos a cada paciente.

Es conveniente, indicar una serie de recomendaciones a tener en cuenta para este tipo de trabajos que tienen como objetivo el desarrollo de una aplicación que permita el diagnóstico de algún tipo de dolencia, como son:

- Intentar mantener la integridad de los datos originales para no tergiversar los resultados.
- Hay que tener en cuenta el tipo de usuario, que sufre la dolencia, para facilitarle de una manera rápida un resultado fácil de interpretar.
- Indicar claramente, por ejemplo, en los resultados, que el diagnóstico médico es siempre prioritario respecto al ofrecido desde la aplicación.
- Mantener la privacidad de los datos de unos usuarios respecto a otros, por ejemplo, con un sistema de acceso restringido.

## 6.1. Utilización y protección de los datos

Los datos utilizados como base de conocimiento para el desarrollo de la aplicación han sido extraídos del artículo *Analysis of Trigger Factors in Episodic Migraineurs Using a SmartphoneHeadache Diary Applications* (Park et al., 2016) y están disponibles bajo licencia *Creative Commons Attribution License*, la cual, permite el uso, distribución y reproducción sin restricciones en cualquier medio, siempre que se acredite el autor y la fuente originales, como así se ha hecho.

Los términos de la licencia establecen que es necesario incluir el enlace a la misma, el cual ha sido incluido en el párrafo anterior, así como una descripción de los cambios realizados.

Hay que indicar que **no se han realizado** cambios significativos sobre los datos originales correspondientes a la base de conocimiento, ni sobre los datos nuevos correspondientes a las pruebas de diagnóstico, manteniendo la privacidad y protección de los mismos como establece el **RGPD**, es decir:

- Se mantiene el anonimato de los 62 pacientes originales extraídos de (Park et al., 2016) y solo ha sido utilizada una parte de los datos sin modificación de los mismos.
- No se muestra información sensible.
- El acceso a los datos correspondientes al diagnóstico están restringidos a cada usuario.
- Los datos utilizados han sido almacenados en una base de datos para ser almacenados como base de conocimiento y exclusivamente para realizar los test de pruebas de la aplicación desarrollada.

## Capítulo 7

# Conclusiones y trabajo futuro

### 7.1. Conclusiones

La aplicación **ADMTF** desarrollada, permite a la persona que sufre migrañas, una manera sencilla de determinar la probabilidad de que la cefalea que ha sufrido corresponde a una *migraña sin aura* o no, utilizando una base de datos, como base de conocimiento de 62 personas que han sufrido migrañas, y para ello, se han utilizado parte de los datos disponibles en (Park et al., 2016) como se ha descrito en [6.1 Utilización y protección de los datos](#).

Para obtener dicha probabilidad se han utilizado técnicas de aprendizaje automático para obtener el modelo que mejor responde a la base de datos utilizada, una vez elegido el modelo de **regresión logística multinomial** comprobando su eficacia con diferentes bloques de datos de entrenamiento y de test, ha sido optimizado mediante diferentes técnicas para mejorar una de las métricas más habituales como es la precisión.

Una de las características más importantes de esta aplicación es que al utilizar la base de conocimiento de (Park et al., 2016) permite además ir creando una base de datos de los factores desencadenantes de las migrañas que se ha demostrado son los más habituales [Tabla 4](#), y así poder lograr a largo plazo:

- Un histórico por paciente de los desencadenantes más habituales en sus migrañas. Esta información permite al usuario disponer de la información básica necesaria para tratar de eludir esos desencadenantes en lo posible en un futuro y así tratar de evitar la frustración que conlleva en muchas ocasiones tratamientos erróneos por falta de información.
- Una histórico general que va creando una base de conocimiento que determine que desencadenantes son más habituales en las personas que sufren migrañas.

Otra característica añadida, es que la aplicación permite determinar de la manera más

objetiva posible, es decir que dependa lo menos posible del criterio del paciente, si dos de los desencadenantes más habituales como son el **estrés** y la **carencia de sueño** han podido desencadenar la migraña mediante el análisis de los datos extraídos de un reloj de uso comercial como es el **Apple Watch**.

Para realizar este análisis objetivo de estos dos factores, los datos obtenidos y analizados desde el *Apple Watch* son los siguientes:

- La métrica utilizada es la variabilidad del ritmo cardiaco (*HRV*) en el dominio temporal, concretamente la desviación estándar en mili-segundos. La utilización de un reloj comercial limita el número de sensores disponibles que permitan determinar con una mayor probabilidad el echo de que se haya producido un episodio de estrés o no, pero también hay que decir que si las medidas son realizadas de manera metódica con el procedimiento descrito en 4.5.1 es muy buen punto de partida para determinar los niveles de estrés de manera simple y económica.
- Registro de las horas promedio de sueño en los últimos días y comparativa con respecto al promedio que se concibe como rango mínimo de sueño aceptable (6horas). Permite determinar de una manera muy simple si el paciente ha dormido las horas necesarias y como han ido evolucionando esas horas en los últimos 7 días.

Tratar de aclarar cuál o cuales desencadenantes son los factores que han podido influir en las migrañas de un determinado paciente es uno de los objetivos finales de este trabajo, obviamente, solo es una introducción que trata de aportar algo para mejorar el tratamiento y diagnóstico de este tipo de cefaleas sin perder la perspectiva de la dificultad que implica todo buen diagnóstico.

## 7.2. Trabajo futuro

Este trabajo ha sido un breve introducción en el campo del diagnóstico de las migrañas sin aura y sus desencadenantes más habituales. La aplicación desarrollada en el mismo, a pesar de su sencillez, permite a una persona que sufre migrañas sin aura llevar un registro de las mismas, para obtener información de manera fehaciente, es decir, como le afectan determinados desencadenantes, si es posible limitar el efecto de los mismos a la hora de causarla, etc.

Este trabajo en un futuro debería desarrollarse para darle un valor añadido, en al menos los siguientes aspectos reseñables:

- El número de desencadenantes de las migrañas es fijo, y han sido elegidos como los más habituales desde diferentes fuentes, pero sobre todo de la base de datos de 62 pacientes utilizada (Park et al., 2016). Un opción clara de mejora es la de añadir la posibilidad de la creación dinámica de más desencadenantes, si el usuario no encuentra respuesta en los más habituales que le permitan añadir nuevos con el objetivo de que la base de datos vaya creciendo en cuanto al número de los mismos.
- El rendimiento del modelo de aprendizaje utilizado, *regresión logística multinominal* para el diagnóstico de las migrañas sin aura es efectivo, con buenos rendimientos obtenidos después de los diferentes procedimientos de verificación realizados en 4.3, pero habría que comparar su eficacia y rendimiento con otros modelos de aprendizaje de buena eficacia en este tipo de conjunto de datos, como por ejemplo los árboles de decisión, utilizando la misma librería **Scikit-learn** (Pedregosa et al., 2011) o las redes neuronales, por ejemplo, utilizando la librería de aprendizaje profundo **Keras** (Chollet, 2015).
- El modelo de aprendizaje utilizado ha sido generado utilizando la base de datos original, con los 62 pacientes contrastados, se podría tratar de crear de manera dinámica cada cierto tiempo el modelo en función del incremento de la base de datos de la aplicación en *MySQL* para así crear cada vez el modelo con más datos disponibles y por lo tanto con menos probabilidad de *overfitting*.
- Los datos extraídos del *Apple Watch* para determinar si se está más expuesto a situaciones de estrés o no, se basan en la variabilidad del ritmo cardiaco (*HRV*) en el dominio temporal, el diagnóstico del **estrés** es un campo ampliamente mejorable, de entre las diferentes posibilidades están las siguientes:
  - Utilizar un reloj, profesional, enfocado a la captura de datos de salud, con un número

de sensores mayor, como ejemplo de referencia está el *Empatica E4* que dispone de sensor de actividad electrodérmica (*EDA*) que añade mayor precisión en el diagnóstico de procesos de estrés.

- Extraer el *HRV* de otros conjuntos de datos como el descrito en el apartado 2.1.2.1 *Introducing WESAD, a Multimodal Dataset for Wearable Stress and Affect Detection* utilizando la librería **Neurokit** (Makowski et al., 2020), el análisis y desarrollo de los datos obtenidos de los diferentes sensores en esta interesante base de datos **WESAD** permiten obtener datos muy precisos de los episodios de estrés de las 13 personas que fueron sometidas a situaciones de estrés.
- Para objetivamente evaluar si la carencia de sueño es un desencadenante de los datos extraídos del *Apple Watch*, simplemente se obtienen datos de las horas dormidas por día y el promedio de horas efectivas dormidas los últimos 7 días, si se obtienen los datos de la frecuencia cardiaca mientras se duerme que también están disponibles, se puede verificar no solo el número de horas dormido sino la calidad del mismo.

## Capítulo 8

# Bibliografía

### Referencias

- Ahmed, F. (2012, agosto). Headache disorders: Differentiating and managing the common subtypes. *Br J Pain*, 6(3), 124--132. <https://doi.org/10.1177/2049463712459691>
- Akben, S., Tuncel, D., y Alkan, A. (2016, abril). Classification of multi-channel EEG signals for migraine detection. *Biomedical Research*, 27, 743--748. Descargado 2021-05-21, de <https://www.biomedres.info/abstract/classification-of-multichannel-eeeg-signals-for-migraine-detection-4561.html>
- Apple Watch*. (s.f.). Descargado 2021-04-25, de <https://www.apple.com/watch/>
- Aprilliant, A. (2021, enero). *Optimal Threshold for Imbalanced Classification*. Descargado 2021-04-03, de <https://towardsdatascience.com/optimal-threshold-for-imbalanced-classification-5884e870c293>
- AutoSleep. Sleep Tracking*. (s.f.). Tantsissa. Descargado 2021-06-24, de <https://autosleepapp.tantsissa.com/home>
- Biosignalsplux | respiBAN Pro*. (s.f.). Descargado 2021-04-29, de <https://biosignalsplux.com/products/wearables/respiban-pro.html>
- Bogdanovist. (s.f.). *How to choose a predictive model after k-fold cross-validation?* Cross Validated. Descargado de <https://stats.stackexchange.com/q/52277>
- Brownlee, J. (2017, marzo). *How to Train a Final Machine Learning Model*. Descargado 2021-08-23, de <https://machinelearningmastery.com/train-final-machine-learning-model/>
- Brownlee, J. (2020, febrero). *A Gentle Introduction to Threshold-Moving for Imbalanced Classification*. Descargado 2021-07-05, de <https://machinelearningmastery.com/threshold-moving-for-imbalanced-classification/>
- Can, Y. S., Chalabianloo, N., Ekiz, D., y Ersoy, C. (2019, enero). Continuous Stress Detection

- Using Wearable Sensors in Real Life: Algorithmic Programming Contest Case Study. *Sensors*, 19(8), 1849. <https://doi.org/10.3390/s19081849>
- Castaldo, R., Melillo, P., Bracale, U., Caserta, M., Triassi, M., y Pecchia, L. (2015, abril). Acute mental stress assessment via short term HRV analysis in healthy adults: A systematic review with meta-analysis. *Biomedical Signal Processing and Control*, 18, 370--377. <https://doi.org/10.1016/j.bspc.2015.02.012>
- Celik, U., Yurtay, N., y Pamuk, Z. (2014, enero). Migraine Diagnosis by Using Artificial Neural Networks and Decision Tree Techniques. *AJIT-e: Bilişim Teknolojileri Online Dergisi*, 5(14), 79--90. <https://doi.org/10.5824/1309-1581.2014.1.005.x>
- Cervantes, J., Li, X., Yu, W., y Li, K. (2008, enero). Support vector machine classification for large data sets via minimum enclosing ball clustering. *Neurocomputing*, 71(4), 611--619. <https://doi.org/10.1016/j.neucom.2007.07.028>
- Chollet, F. (2015). *Keras*. Descargado de <https://keras.io>
- Chris P. (s.f.). *Plotly: An open-source, interactive data visualization library for Python*. Descargado 2020-08-30, de <https://plotly.com/python/>
- E4 wristband | Real-time physiological signals | Wearable PPG, EDA, Temperature, Motion sensors*. (s.f.). Descargado 2021-04-25, de <https://www.empatica.com/research/e4>
- Flask*. (s.f.). Descargado 2021-02-15, de <https://palletsprojects.com/p/flask/>
- Fletcher, R. (1987). *Practical methods of optimization; (2nd ed.)*. USA: Wiley-Interscience. Descargado de <https://dl.acm.org/doi/10.5555/39857>
- Garcia-Ceja, E., Osmani, V., y Mayora, O. (2016, julio). Automatic Stress Detection in Working Environments From Smartphones' Accelerometer Data: A First Step. *IEEE J Biomed Health Inform*, 20(4), 1053--1060. <https://doi.org/10.1109/JBHI.2015.2446195>
- GBD Compare | IHME Viz Hub*. (s.f.). Descargado 2021-04-19, de <https://vizhub.healthdata.org/gbd-compare>
- Göbel, C. H., Karstedt, S. C., Münte, T. F., Göbel, H., Wolfrum, S., Lebedeva, E. R., . . . Royl, G. (2020, enero). ICHD-3 is significantly more specific than ICHD-3 beta for diagnosis of migraine with aura and with typical aura. *J Headache Pain*, 21(1), 2. <https://doi.org/10.1186/s10194-019-1072-2>
- Headache Classification Committee of the International Headache Society (IHS) The International Classification of Headache Disorders, 3rd edition. (2018, enero). *Cephalalgia*, 38(1), 1--211. <https://doi.org/10.1177/0333102417738202>
- Headache disorders*. (s.f.). Descargado 2021-04-21, de <https://www.who.int/news-room/fact-sheets/detail/headache-disorders>



- Hernando, D., Roca, S., Sancho, J., Alesanco, Á., y Bailón, R. (2018, agosto). Validation of the Apple Watch for Heart Rate Variability Measurements during Relax and Mental Stress in Healthy Subjects. *Sensors (Basel)*, 18(8). <https://doi.org/10.3390/s18082619>
- Homan, R. W. (1988, diciembre). The 10-20 Electrode System and Cerebral Location. *American Journal of EEG Technology*, 28(4), 269--279. <https://doi.org/10.1080/00029238.1988.11080272>
- Houle, T. T., Butschek, R. A., Turner, D. P., Smitherman, T. A., Rains, J. C., y Penzien, D. B. (2012, diciembre). Stress and Sleep Duration Predict Headache Severity in Chronic Headache Sufferers. *Pain*, 153(12), 2432--2440. <https://doi.org/10.1016/j.pain.2012.08.014>
- Inc., P. T. (2015). *Collaborative data science*. Montreal, QC: Plotly Technologies Inc. Descargado de <https://plot.ly>
- Institute for Health Metrics and Evaluation. (s.f.). Descargado 2021-06-08, de <http://www.healthdata.org/institute-health-metrics-and-evaluation>
- The International Classification of Headache Disorders, 3rd edition (beta version). (2013, julio). *Cephalalgia*, 33(9), 629--808. <https://doi.org/10.1177/0333102413485658>
- Joblib Development Team. (2021). *Joblib: Running Python functions as pipeline jobs — joblib 1.1.0.dev0 documentation*. Descargado 2021-07-07, de <https://joblib.readthedocs.io/>
- Kelman, L. (2007, mayo). The triggers or precipitants of the acute migraine attack. *Cephalalgia*, 27(5), 394--402. <https://doi.org/10.1111/j.1468-2982.2007.01303.x>
- Kim, H.-G., Cheon, E.-J., Bai, D.-S., Lee, Y. H., y Koo, B.-H. (2018, marzo). Stress and Heart Rate Variability: A Meta-Analysis and Review of the Literature. *Psychiatry Investig*, 15(3), 235--245. <https://doi.org/10.30773/pi.2017.08.17>
- Kim, J., Cho, S.-J., Kim, W.-J., Yang, K. I., Yun, C.-H., y Chu, M. K. (2017, abril). Insufficient sleep is prevalent among migraineurs: A population-based study. *The Journal of Headache and Pain*, 18(1), 50. <https://doi.org/10.1186/s10194-017-0756-8>
- Kwon, S., Kim, H., y Yeo, W.-H. (2021, mayo). Recent advances in wearable sensors and portable electronics for sleep monitoring. *iScience*, 24(5), 102461. <https://doi.org/10.1016/j.isci.2021.102461>
- La inversión en el tratamiento de la depresión y la ansiedad tiene un rendimiento del 400%. (s.f.). Descargado 2021-09-19, de <https://www.who.int/es/news/item/13-04-2016-investing-in-treatment-for-depression-and-anxiety-leads-to-fourfold-return>
- Lemaître, G., Nogueira, F., y Aridas, C. K. (2017). Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine*

- Learning Research*, 18(17), 1--5. Descargado 2021-07-05, de <http://jmlr.org/papers/v18/16-365.html>
- Makowski, D., Zen, Pham, T., Brammer, J. C., Lespinasse, F., Hùng), H. P. P. T., . . . Gupta, V. (2020, agosto). *Neuropsychology/NeuroKit: 0.0.40*. Zenodo. <https://doi.org/10.5281/zenodo.3972510>
- Marmura, M. J. (2018). Triggers, Protectors, and Predictors in Episodic Migraine. *Current pain and headache reports*, 22(12), 1--9. <https://doi.org/10.1007/s11916-018-0734-0>
- NumPy*. (s.f.). Descargado 2020-11-18, de <https://numpy.org/Pallets/flask>
- Pallets/flask*. (2021, febrero). The Pallets Projects. Descargado 2021-02-15, de <https://github.com/pallets/flask>
- Pandas - Python Data Analysis Library*. (s.f.). Descargado 2020-09-01, de <https://pandas.pydata.org/>
- Park, J.-W., Chu, M. K., Kim, J.-M., Park, S.-G., y Cho, S.-J. (2016, febrero). Analysis of Trigger Factors in Episodic Migraineurs Using a Smartphone Headache Diary Applications. *PLoS ONE*, 11(2). <https://doi.org/10.1371/journal.pone.0149577>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(85), 2825--2830. Descargado 2021-02-08, de <http://jmlr.org/papers/v12/pedregosa11a.html>
- Plass-Oude Bos, D. (2006, enero). EEG-based Emotion Recognition. *The Influence of Visual and Auditory Stimuli*.
- Renaud, P., y Blondin, J.-P. (1997, septiembre). The stress of Stroop performance: Physiological and emotional responses to color-word interference, task pacing, and pacing speed. *International Journal of Psychophysiology*, 27(2), 87--97. [https://doi.org/10.1016/S0167-8760\(97\)00049-4](https://doi.org/10.1016/S0167-8760(97)00049-4)
- Rocca, B. (2021, febrero). *Handling imbalanced datasets in machine learning*. Descargado 2021-07-05, de <https://towardsdatascience.com/handling-imbalanced-datasets-in-machine-learning-7a0e84220f28>
- Romero De la Hoz, Z., y Rúa Ascar, J. M. (2014). Análisis de modelos de redes neuronales artificiales, para un sistema de diagnósticos de migrañas con aura y sin aura. *Universidad Simón Bolívar*. Descargado 2021-05-19, de <https://bonga.unisimon.edu.co/handle/20.500.12442/2112>
- Roy, K., Kar, S., y Das, R. N. (2015, enero). Chapter 6 - Selected Statistical Methods in QSAR. En K. Roy, S. Kar, y R. N. Das (Eds.), *Understanding the Basics of QSAR for*

- Applications in Pharmaceutical Sciences and Risk Assessment* (pp. 191--229). Boston: Academic Press. <https://doi.org/10.1016/B978-0-12-801505-6.00006-5>
- Ruiz, M., León, C., Castillo, J., Martínez, M., Sánchez, S., y Quintela, E. (2010, enero). Distribución por diagnósticos de las cefaleas que acuden a los servicios de urgencias de atención primaria. *Semergen*, 36(1), 10--15. <https://doi.org/10.1016/j.semerg.2009.10.002>
- Saito, T., y Rehmsmeier, M. (2015, marzo). The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. *PLoS One*, 10(3), e0118432. <https://doi.org/10.1371/journal.pone.0118432>
- Salai, M., Vassányi, I., y Kósa, I. (2016). Stress Detection Using Low Cost Heart Rate Sensors. *J Healthc Eng*, 2016. <https://doi.org/10.1155/2016/5136705>
- Sanchez-Sanchez, P. A., García-González, J. R., y Rúa Ascar, J. M. (2020, julio). Automatic migraine classification using artificial neural networks. *F1000Res*, 9, 618. <https://doi.org/10.12688/f1000research.23181.2>
- Schmidt, P., Reiss, A., Duerichen, R., Marberger, C., y Van Laerhoven, K. (2018). Introducing WESAD, a Multimodal Dataset for Wearable Stress and Affect Detection. En *Proceedings of the 2018 on International Conference on Multimodal Interaction - ICMI '18* (pp. 400--408). Boulder, CO, USA: ACM Press. <https://doi.org/10.1145/3242969.3242985>
- Seddik, A. H., Branner, J. C., Ostwald, D. A., Schramm, S. H., Bierbaum, M., y Katsarava, Z. (2020, diciembre). The socioeconomic burden of migraine: An evaluation of productivity losses due to migraine headaches based on a population study in Germany. *Cephalalgia*, 40(14), 1551--1560. <https://doi.org/10.1177/0333102420944842>
- Shaffer, F., y Ginsberg, J. P. (2017, septiembre). An Overview of Heart Rate Variability Metrics and Norms. *Front Public Health*, 5, 258. <https://doi.org/10.3389/fpubh.2017.00258>
- Spilsbury, M. J., y Euceda, A. (2016, diciembre). Transformada Rápida de Fourier. *Revista de la Escuela de Física*, 4(2), 45--52. <https://doi.org/10.5377/ref.v4i2.8276>
- Stevens, S., y Clark, G. (2004, enero). CHAPTER 6 - POLYSOMNOGRAPHY. En D. Stevens (Ed.), *Sleep Medicine Secrets* (pp. 45--63). Hanley & Belfus. <https://doi.org/10.1016/B978-1-56053-592-8.50010-5>
- Tarvainen, M. P., Niskanen, J.-P., Lipponen, J. A., Ranta-aho, P. O., y Karjalainen, P. A. (2014, enero). Kubios HRV – Heart rate variability analysis software. *Computer Methods and Programs in Biomedicine*, 113(1), 210--220. <https://doi.org/10.1016/j.cmpb.2013.07.024>
- Tu, J. V. (1996, noviembre). Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. *Journal of Clinical*

- Epidemiology*, 49(11), 1225--1231. [https://doi.org/10.1016/S0895-4356\(96\)00002-9](https://doi.org/10.1016/S0895-4356(96)00002-9)
- Vollmer, M. (2019, diciembre). HRVTool - an Open-Source Matlab Toolbox for Analyzing Heart Rate Variability.. <https://doi.org/10.22489/CinC.2019.032>
- Wickham, H. (2016). *Ggplot2: Elegant graphics for data analysis*. Springer-Verlag New York. Descargado de <https://ggplot2.tidyverse.org>
- Yuan, G., Wei, Z., y Lu, S. (2011, julio). Limited memory BFGS method with backtracking for symmetric nonlinear equations. *Mathematical and Computer Modelling*, 54(1), 367--377. <https://doi.org/10.1016/j.mcm.2011.02.021>
- Zou, Q., Xie, S., Lin, Z., Wu, M., y Ju, Y. (2016, septiembre). Finding the Best Classification Threshold in Imbalanced Classification. *Big Data Research*, 5, 2--8. <https://doi.org/10.1016/j.bdr.2015.12.001>

# Apéndices

## Apéndice A

# Tablas de datos y Notebooks

A continuación se expone una breve descripción de los ficheros disponibles en el repositorio correspondientes a las **tablas de datos** así como los **cuadernos** en *Jupyter* utilizados para el desarrollo de este trabajo.

El objetivo de este anexo, es facilitar su localización y consulta, para ello, indicar que todos están disponibles en el siguiente enlace de *GitHub*:

**<https://github.com/fjarroyovalle/public-ADMTF-data-notebooks>**

### A.1. Ficheros de tablas y datos

La estructura de los ficheros y la descripción de los mismos en la siguiente [Tabla A1](#):

**Tabla A1.** Descripción de los ficheros de datos y tablas disponibles en *GitHub*.

---

<b><i>/DMS_datasets/S1_File_Preprocesed.xlsx</i></b>
<ul style="list-style-type: none"><li>■ Tabla de datos de Google Sheet con datos originales de <i>Analysis of Trigger Factors in Episodic Migraineurs Using a Smartphone Headache Diary Applications</i>.</li><li>■ URL específica de la descarga del archivo original de Excel aquí: <i>Dataset of 62 patients, 1099 Headache Diary Records, and 4579 Diary Records</i>.</li><li>■ Este archivo se procesó con macros de Google Sheets para obtener el <i>Tab-totalDiary4578Filtered</i>.</li></ul>
<b><i>/DMS_datasets/totalDiary4578Filtered.csv</i></b>
<ul style="list-style-type: none"><li>■ Archivo en formato CSV exportado desde la tabla original <i>S1_File_Preprocesed.xlsx</i> (<i>Tab-totalDiary4578Filtered</i>).</li></ul>

---

---

**migraines\_db\_table.csv**

---

- Archivo en formato CSV con todos los datos de las migrañas obtenidos de la tabla original */DMS\_datasets/totalDiary4578Filtered.csv* para que pueda ser cargado en la base de datos *MySql* de la aplicación principal.

## A.2. Notebooks y repositorios

En este repositorio están los cuadernos *Jupyter* utilizados para el desarrollo de este trabajo, los cuales se describen en la siguiente [Tabla A2](#):

**Tabla A2.** Descripción de los cuadernos de *Jupyter*.

---

**ADMTF ML Models for Diagnosing Migraine Situations (DMS).ipynb**

---

- Este *cuaderno* incluye el estudio realizado para obtener el modelo de aprendizaje más adecuado, en este caso ha sido elegido un modelo de **Regresión Logística Multinomial** con un ajuste óptimo del **umbral** para mejorar el rendimiento del modelo.

---

**ADMTF\_Stress\_LackSleep\_Eval (SLSE).ipynb**

---

- Este *cuaderno* incluye el procesado de los ficheros de datos descargados desde el reloj comercial de pulsera **Apple Watch** para obtener las gráficas con la evolución del **estrés** y de la **carencia de sueño** como principales posibles desencadenantes de las migrañas.

---

**DFrame\_To\_MySQLDBase\_TableMigraines (DFTDB).ipynb**

---

- Este *cuaderno* incluye el procesado de los datos de las migrañas para prepararlo para importar en la base de datos de **MySql** de la aplicación principal.

---

**Plot\_Triggers\_Migraines\_Results (PTM).ipynb**

---

- Este *cuaderno* incluye el las visualizaciones obtenidas utilizando la librería *Plotly* (Chris P, s.f.) para comparar de manera visual los factores desencadenantes que han sido más importantes a la hora de causar las migrañas sin aura tanto para cada persona de manera específica como para todo el conjunto de individuos que componen la base de datos.

### A.3. Fichero con el modelo obtenido

En la [Tabla A3](#) una breve descripción del fichero utilizado modelo de aprendizaje:

**Tabla A3.** Descripción del fichero con el modelo de aprendizaje empaquetado.

---

***DMTF\_LR\_Model.pkl***

---

- Fichero empaquetado en formato *pkl* por medio de la librería ([Joblib Development Team, 2021](#)) para poder cargar el modelo de aprendizaje en la aplicación principal realizada en ([Flask, s.f.](#)).