UNIVERSIDAD DE VALLADOLID

FACULTAD DE MEDICINA
ESCUELA DE INGENIERÍAS INDUSTRIALES

TRABAJO DE FIN DE GRADO
GRADO EN INGENIERÍA BIOMÉDICA

Aplicación de algoritmos de machine learning para la predicción automática del grado de sedación y analgesia en el paciente de UCI

Autor/a:
D.ª Laura Pinilla Domingo

Tutor/a:
Dr. D. Javier Gómez Pilar
Dra. D. ª Elena Bustamante Munguira

Valladolid, septiembre de 2025

TÍTULO:	**Aplicación de algoritmos de Machine Learning para la predicción automática del grado de sedación y analgesia en el paciente de UCI**
AUTOR/A:	**D.ª Laura Pinilla Domingo**
TUTOR/A:	**Dr. D. Javier Gómez Pilar** **Dra. D. ª Elena Bustamante Munguira**
DEPARTAMENTO:	**Teoría de la Señal y las Comunicaciones e Ingeniería Telemática** **Medicina, Dermatología y Toxicología**

TRIBUNAL

PRESIDENTE:	**Javier Gómez Pilar**
SECRETARIO:	**Gonzalo C. Gutiérrez Tobal**
VOCAL:	**Elena Bustamante Munguira**
SUPLENTE 1:	**Jesús Poza Crespo**
SUPLENTE 2:	**Carlos Gómez Peña**

FECHA:	**Septiembre 2025**
CALIFICACIÓN:	

A todos los que han estado a mi lado,
en las buenas y en las malas.

Agradecimientos

Primeramente, deseo expresar mi más sincero agradecimiento a todo el personal sanitario de la Unidad de Cuidados Intensivos del Hospital Clínico Universitario de Valladolid. Muy especialmente, a mi tutora, Elena, por su cercanía y la confianza depositada en mí desde el primer momento. Gracias también a Guille, por su tiempo, su paciencia diaria y su dedicación. Estos dos meses con vosotros han sido enriquecedores, tanto profesional como personalmente. Es un orgullo saber que contamos con profesionales como vosotros y una tranquilidad saber que estamos en las mejores manos.

Seguidamente, quiero agradecer este trabajo a mi tutor, Javier Gómez. Gracias por tu atención y, sobre todo, por haber apostado por mí y reconocido mi potencial. Siempre estaré agradecida por la confianza mostrada. Ha sido un privilegio haber podido contar contigo. Como estudiante, no hay nada más gratificante que poder aprender de alguien que disfruta de su trabajo y lo transmite de la misma manera.

De forma muy especial, quiero agradecer a mi familia el apoyo incondicional que me han brindado durante estos cuatro años. A mis padres, por haberme educado en valores y por recordarme que, pase lo que pase, siempre podré contar con vosotros. A mi padre, por impulsarme a luchar por mis sueños, por creer en mí y por acompañarme en los momentos de incertidumbre. A mi madre, por cuidarme, protegerme y quererme como solo una madre sabe hacerlo. Nunca permitas que el miedo se apodere de ti. A mi hermana, por ser, como tú dirías, *el ejemplo del mal ejemplo*. Aunque yo siempre he pensado, que eres el mejor referente que se puede tener. Estoy orgullosa de ti.

Gracias a mis amigas del Ave, por estar siempre. Pase el tiempo que pase, sea donde sea, sé que siempre podré contar con vosotras. Las buenas amistades perduran toda la vida, y a nosotras nos quedan muchos años.

A mis amigas de la universidad, por compartir lágrimas, alegrías y muchos desayunos. Han sido cuatro años muy felices, y sin vosotras no habrían sido lo mismo. No todos tienen la suerte de encontrar a alguien con el que simplemente ser, y aquí hemos encontrado diez piezas de puzle que encajan a la perfección.

Y, por supuesto, gracias a Anita. Ojalá todos tuvieran la suerte de contar con una amiga como tú. Gracias por estar, por hacerme la vida más fácil y por ser simplemente tú. Soy muy afortunada.

Por último, quiero dedicar este trabajo, y todos estos años de esfuerzo, a mí misma. Por superarme día a día y por aprender a creer en mí. Por tropezarme y levantarme. Por aprender a *disfrutar de los momentos felices en honor a los tiempos difíciles.*

Gracias a todos.

Resumen

El ajuste adecuado de la analgesia y la sedación es fundamental en el cuidado del paciente crítico. No solo ayuda a prevenir complicaciones, sino también es un elemento clave para asegurar su bienestar y conseguir una pronta recuperación. Sin embargo, lograr un control preciso de estos parámetros no es una tarea sencilla. Las condiciones fisiopatológicas de cada paciente, sumadas a la dependencia de escalas subjetivas para evaluar el nivel de sedación dificultan significativamente la labor.

Ante esta problemática, se han investigado métodos de automatización de la evaluación del nivel de sedación mediante el análisis de la señal de EEG. No obstante, su implementación en UCI sigue siendo limitada debido al elevado volumen de datos que genera y a la complejidad inherente de su interpretación, que generalmente requiere la intervención de un neurofisiólogo. Por otro lado, una de las herramientas objetivas más utilizadas para estimar el grado de conciencia del paciente es el BIS. A pesar de su utilidad, su uso se reserva para casos muy concretos debido al coste y carga de trabajo adicional que implicaría al personal sanitario.

En este contexto, el objetivo de este Trabajo Fin de Grado es desarrollar un modelo predictivo que permita clasificar automáticamente a los pacientes en tres grupos: infrasedados, sobresedados y sedación adecuada. Para ello, se analizarán distintas variables clínicas y sociodemográficas mediante diversas técnicas de aprendizaje automático. Además, con el fin de buscar potencial nuevo conocimiento relacionado con la adecuación de la dosis de sedación, se identificarán las características más relevantes para la predicción de dichas clases a través de técnicas de Inteligencia Artificial Explicable.

La base de datos utilizada se construyó a partir de información recopilada de 100 pacientes ingresados en la UCI del Hospital Clínico Universitario de Valladolid. El nivel de sedación fue evaluado por dos profesionales médicos, cuyas valoraciones sirvieron como *gold standard* para el entrenamiento de los modelos. Previo a este proceso, se llevó a cabo el preprocesamiento del conjunto de datos, incluyendo la limpieza, imputación de valores faltantes y análisis estadístico descriptivo. Posteriormente, se aplicó *Fast Correlation-Based Filter* para la selección de características, y se desarrollaron seis modelos de clasificación basados en regresión logística, análisis discriminante lineal y *Adaboost*. Finalmente, sobre el modelo de mejor desempeño, se calcularon los *shapley values* para interpretar la importancia de cada variable en las predicciones realizadas.

Los resultados indicaron que el modelo *AdaBoostM2* entrenado con el *target* 2 era capaz de clasificar a los pacientes con una *accuracy* del 66% y un índice *Kappa* del 0.33, lo que supone una mejora de 0.14 respecto al grado de concordancia entre las dos valoraciones de los profesionales. A demás, este modelo utilizó únicamente seis variables y no requirió del uso del BIS, lo que demuestra su potencial para automatizar la clasificación del nivel de sedación sin depender de este dispositivo. El análisis de los *shapley values* reveló que bastaba con cinco variables para realizar predicciones acertadas, siendo el peso del paciente la más relevante. Este hallazgo resulta interesante, ya que, si bien el peso corporal puede influir en la farmacocinética de los sedantes, no suele considerarse una de las variables más determinantes en la práctica clínica para evaluar el nivel de sedación.

Gracias a la elaboración de este trabajo, se ha conseguido automatizar la clasificación de pacientes en función del nivel de sedación. El modelo desarrollado no solo supera el grado de concordancia entre clínicos, sino que también prescinde del uso del BIS, ofreciendo así una herramienta accesible y eficiente. Este avance puede facilitar un control más preciso de las dosis administradas, reducir los riesgos asociados a una sedación inadecuada y contribuir a una atención personalizada y más segura para los pacientes críticos.

Palabras clave

Sedación, analgesia, evaluación del nivel de sedación, Unidad de Cuidados Intensivos, paciente crítico, *Machine Learning*, modelos predictivos, Inteligencia Artificial Explicable.

Abstract

Proper adjustment of analgesia and sedation is essential in the care of critically ill patients. Not only does it help prevent complications, but it is also key to ensuring their well-being and achieving a speedy recovery. However, achieving precise control over these parameters is not an easy task. Each patient's pathophysiological conditions, along with the reliance on subjective scales to assess sedation levels, significantly complicate the process.

In response to this challenge, automated methods for evaluating sedation levels through EEG signal analysis have been explored. Nevertheless, their implementation in ICU remains limited due to the large volume of data generated and the intrinsic complexity of the signal, which generally requires interpretation by a neurophysiologist. On the other hand, one of the most widely used objective tools to estimate the patient's level of consciousness is the BIS. Despite its usefulness, its application is limited to very specific cases due to its cost and the additional workload it places on healthcare personnel.

In this context, the aim of this Final Degree Project is to develop a predictive model capable of automatically classifying patients into three categories: under-sedated, over-sedated, and adequately sedated. To achieve this, various clinical and sociodemographic variables will be analysed using different machine learning techniques. Additionally, in order to discover potential new insights related to optimal sedation dosage, the most relevant features for predicting theses classes will be identified through Explainable Artificial Intelligence techniques.

The dataset used was built from information collected from 100 patients admitted to the ICU of the Hospital Clínico Universitario de Valladolid. Sedation levels were assessed by two medical professionals, whose evaluations served as the gold standard for model training. Prior to this process, the dataset underwent preprocessing, including data cleaning, missing value imputation, and descriptive statistical analysis. Afterwards, Fast Correlation-Based Filter method was applied for feature selection, and six classification models were developed based on logistic regression, linear discriminant analysis, and AdaBoost algorithms. Finally, for the best-performing model, Shapley values were calculated to interpret the importance of each variable in the model's predictions.

The results showed that the AdaBoostM2 model trained with target 2 was able to classify patients with an accuracy of 66% and a Kappa index of 0.33, representing an improvement of 0.14 over the level of agreement between the two professionals' evaluations. Moreover, this model required only six variables and did not rely on the BIS, demonstrating its potential to automate sedation level classification without reliance on this device. The Shapley value analysis revealed that just five variables were needed to make accurate predictions, with patient weight being the most important clinical feature. This finding is particularly interesting, as although body weight can influence the pharmacokinetics of sedatives, it is not usually considered one of the most decisive variables in clinical practice when evaluating sedation levels.

Thanks to this work, it has been possible to automate the classification of patients based on their level of sedation. The resulting model not only surpasses the level of agreement between clinicians, but also dispenses with the use of BIS, thus providing an accessible and efficient tool. This advancement could enable more precise control of administered

doses, reduce the risks associated with inadequate sedation, and contribute to personalised and safer care for critically ill patients

Keywords

Sedation, analgesia, assessment of sedation level, Intensive Care Unit, critical patient, Machine Learning, predictive models, Explainable Artificial Intelligence.

Índice general

Índice de Figuras

Índice de tablas

Glosario de abreviaturas y acrónimos

Acc ..... *Accuracy*
ACV ..... *Accidente Cerebrovascular*
AdaBoost ..... *Adaptive Boosting*
AIED ..... *Asociación Internacional para el Estudio del Dolor*
ANFIS ..... *Sistemas Neuro-Difusos*
APACHE II ..... *Acute Physiology and Chronic Health Evaluation II*
AUC ..... *Área bajo la curva*
BDZ ..... *Benzodiacepinas*
BIPAP ..... *Presión Positiva Binivel de las Vías Respiratorias*
BIS ..... *Índice Biespectral*
CPAP ..... P*resión Positiva Continua en la Vía Respiratoria*
DM ..... *Diabetes Miellitus*
EEG ..... *Electroencefalograma*
EPOC ..... *Enfermedad Pulmonar Obstructiva Crónica*
ESCID ..... *Escala de Conductas Indicadoras de Dolor*
EVN ..... *Escala Verbal Numérica*
EVA ..... *Escala Visual Analógica*
FC ..... *Frecuencia Cardiaca*
FCBF ..... *Fast Correlation-Based Filter*
FN ..... *Falsos negativos*
FP ..... *Falsos positivos*
FR ..... *Frecuencia Respiratoria*
GCS ..... *Escala de Coma de Glasgow*
HTA ..... *Hipertensión arterial*
IA ..... *Inteligencia Artificial*
ICCA ..... *IntelliSpace Cuidados Críticos y Anestesia*
IMC ..... *Índice de masa corporal*
KNN ..... *K-Nearest Neighbors*
LDA ..... *Análisis del discriminante lineal*
LIME ..... *Local Interpretable Model-agnóstic Explanations*
LORE ..... *LOcal Rule-based Explainer*
LR ..... *Razón de verisimilitud o Likelihood Ratio*
LR- ..... *Razón de verisimilitud negativa*
LR+ ..... *Razón de verisimilitud positiva*
MI ..... *Motivo de ingreso*
MICE ..... *Multivariate Imputation by Chained-Equations*
ML ..... *Machine Learning*
NPV ..... *Valor predictivo negativo*
PCA ..... *Análisis de Componentes Principales*
PCs ..... *Componentes Principales*
PPV ..... *Valor predictivo positivo*

RASS ..... *Escala de agitación-sedación de Richmond*
RL ..... *Regresión Logística*
SAS ..... *Escala de sedación-agitación*
Se ..... *Sensibilidad*
SEMICYUC .. Sociedad Española de Medicina Intensiva, Crítica y Unidades Coronarias
SHAP ..... *SHapley Additive exPlanations*
SNC ..... *Sistema Nervioso Central*
SOFA ..... *Sequential Organ Failure Assessment*
Sp ..... *Especificidad*
SU ..... *Symmetrical Uncertainty*
SVM ..... *Máquinas vector soporte*
TAM ..... *Tensión arterial media*
TCE ..... *Traumatismo craneoencefálico*
TFG ..... *Trabajo Fin de Grado*
TPR ..... *Tasa de verdaderos positivos*
UCI ..... *Unidad de Cuidados Intensivos*
VM ..... *Ventilación mecánica*
VN ..... *Verdaderos negativos*
VP ..... *Verdaderos positivos*
XAI ..... *Inteligencia Artificial Explicable*

Capítulo 1. Introducción

1.1 Introducción

La Unidad de Cuidados Intensivos (UCI) es el área hospitalaria encargada del cuidado y la atención continua del paciente en estado crítico. Se considera paciente crítico a aquel que padece una disfunción o enfermedad de tal gravedad que representa una amenaza para su vida (Aguilar García et al., 2017).

El ajuste adecuado de la analgesia y sedación en cada momento de la evolución clínica del paciente es imprescindible para poder garantizar su comodidad y seguridad (Schweickert & Kress, 2008). La gravedad de su afección, la variedad de procedimientos invasivos a los que se someten para su tratamiento —como la ventilación mecánica— y el ambiente, a veces hostil, de la UCI, pueden causar altos niveles de estrés (Schweickert & Kress, 2008)

Tanto la sedación excesiva (disminución de la motilidad gastrointestinal, debilidad muscular, mayor riesgo de delirio, mayor duración de los periodos de respiración mecánicas y periodos prolongados de hospitalización) como la sedación insuficiente (que puede llevar a la autoextracción de tubos y catéteres, a un comportamiento agresivo hacia el personal sanitario y una sincronización inadecuada con el respirador) suponen un riesgo para la recuperación del paciente (Brochard, 2008; Sessler et al., 2002). A pesar de ello, las impredecibles complicaciones clínicas hacen de esta tarea un desafío resultando en que una proporción significativa de los pacientes de UCI experimenta niveles inadecuados de sedación (Jackson et al., 2009).

Ante esta situación, el *Machine Learning (ML)* se ha convertido en una herramienta clave para mejorar la precisión y automatización del ajuste de dichos fármacos, Estudios recientes han aplicado algoritmos *ML* no lineales para predecir el nivel de sedación mediante señales de electroencefalograma (EEG) (Ramaswamy et al., 2022). Otros estudios se centran en la monitorización automática de los tres niveles de anestesia (sedación ligera, sedación moderada y sedación profunda) mediante el análisis de señales del EEG (González Rubio et al., 2019).

Como vemos, la mayor parte de estos enfoques se fundamentan en el análisis del EEG. Se trata de una señal que resulta de gran ayuda, ya que ofrece la monitorización continua del paciente (Kubota et al., 2018). Sin embargo, su disponibilidad en la práctica es muy limitada. El elevado volumen de datos que genera y la complejidad intrínseca de la señal requieren, con frecuencia, la intervención de un neurofisiólogo para su correcta interpretación. Este factor, junto con el incremento de la carga asistencial que conlleva, supondría un aumento de la carga de trabajo el personal sanitario dificultando su implementación en entornos exigentes como el de la UCI (Rubiños & Godoy, 2020).

Desde este punto de vista, en el presente documento proponemos un enfoque donde no sea necesario el uso de esta señal. Para ello se evaluará si la combinación de variables sociodemográficas, clínicas, escalas de valoración y dosis de sedantes permiten construir algoritmos que sean capaces de predecir el nivel de sedación (infrasedación, sobresedación o sedación adecuada). Este enfoque proporcionaría al personal sanitario predicciones continuas del nivel de consciencia de sus pacientes, simplificando la toma de decisiones, facilitando y mejorando indirectamente su pronóstico.

1.2 Analgesia

La Asociación Internacional para el Estudio del Dolor (AIED) define el dolor como una "experiencia sensorial y emocional desagradable asociada con un daño tisular real o potencial, o descrita en términos de dicho daño" *(International association for the study of pain, 2020)*. La respuesta al dolor es variable entre individuos, así como en la misma persona en momentos diferentes. Sin embargo, en el entorno de la UCI, muchos pacientes no pueden expresar su dolor, ya que habitualmente se encuentran intubados o bajo los efectos de diversos sedantes o bloqueadores neuromusculares (Barr et al., 2013).

El dolor en estos pacientes no solo afecta a su bienestar, sino que también desencadena una respuesta de estrés y activación del sistema nervioso simpático, lo que puede provocar taquicardia, hipertensión y un mayor consumo de oxígeno en el miocardio, aumentando así el riesgo de isquemia en pacientes vulnerables. Las causas del dolor son variadas, desde la enfermedad que motivó el ingreso, como traumatismos o cirugías, hasta los procedimientos invasivos requeridos para el soporte intensivo. Además, factores como el cuidado rutinario, la higiene y las complicaciones derivadas de la hospitalización prolongada (atrofia muscular, úlceras por presión, pie equino adquirido o infecciones) pueden contribuir significativamente al dolor del paciente (Guerrero Gutiérrez Manuel et al., 2023).

A pesar de los continuos avances en investigación, el control del dolor sigue siendo un problema significativo. Diversos estudios confirman que aproximadamente la prevalencia del dolor en pacientes ingresados en UCI es del 69% (Ramírez et al., 2018). Según Hurtado Oliver Beatriz et al. la intensidad del dolor en los pacientes críticos es significativamente mayor que en los pacientes con trauma quirúrgico. La edad (pacientes jóvenes), el sexo femenino o la presencia de depresión o ansiedad constituyen una serie de factores predisponentes a tener un dolor más intenso (Hurtado Oliver et al., 2022).

1.2.1 Medicamentos analgésicos en pacientes críticos

Los fármacos utilizados para la analgesia son principalmente opioides. Se tratan de compuestos con una relación estructural similar a sustancias presentes en el opio (Cardoso-Ortiz et al., 2020). Estos fármacos son capaces de inhibir la transmisión de una entrada nociceptiva gracias a su afinidad selectiva por los receptores opioides centrales y periféricos (Cardoso-Ortiz et al., 2020).

Existe una gran variedad de fármacos opioides. Algunos de los más utilizados son:

- La morfina. Se trata del opioide más antiguo en uso médico, habiéndose empleado por primera vez hace más de dos siglos (Bels et al., 2023). Su acción se inicia entre 5 y 10 minutos después de la administración, y su vida media oscila entre 3 y 4 horas. (Bels et al., 2023).

- El fentanilo. Opioide sintético liposoluble y analgésico más utilizado en la UCI. Se trata de un medicamento de gran importancia. Es de 50 a 100 veces más potente que la morfina y tiene un vida media de 2-3 horas. Presenta mínimos efectos depresores sobre el miocardio y tiene la capacidad de reducir las dosis requeridas de anestésicos inhalados (Bels et al., 2023; Guerrero Gutiérrez Manuel et al., 2023).

- El sufentanilo. Derivado del fentanilo con una potencia de 5 a 10 veces mayor que este. Las características de inicio de acción y de tiempo de duración en cambio son parecidas (Bels et al., 2023). Es muy utilizado en pacientes críticos con anestesia general prolongadas. Generalmente se combina con el anestésico midazolam («Sufentanilo», 2015).

1.2.2 Evaluación del dolor en pacientes críticos

El primer paso en el tratamiento del dolor es su correcta detección. Para una identificación adecuada, no se debe depender exclusivamente de indicadores conductuales (como apretar los dientes, arrugar la frente, llorar, etc.), fisiológicos (aumento de la frecuencia cardiaca, tensión arterial, frecuencia respiratoria, etc.) ni síntomas vágales (dilatación pupilar, palidez, sudoración, etc.), ya que estos son signos tardíos (Hurtado Oliver et al., 2022). Es necesario aplicar escalas validadas, cuya elección dependerá del nivel de consciencia y la capacidad de comunicación del paciente.

En pacientes comunicativos, encontramos generalmente dos tipos de escala:

- Escala Visual Analógica (Figura 1)

  La escala visual analógica (EVA) representa la intensidad del dolor en una línea recta, habitualmente de 10 cm de longitud. En el extremo izquierdo, el valor 0 indica la ausencia de valor, mientras que, en el extremo opuesto, el valor 10 representa el mayor grado que el paciente puede soportar. Actualmente se trata de la forma de valoración del dolor más utilizada en pacientes comunicativos. El paciente indica un punto dentro de esta recta y se mide para sacar el EVA. Después clasificamos el dolor en base a la puntuación. Un EVA inferior a 4 puede clasificarse como leve, de 4 a 6 como dolor moderado y superior a 6 implica un dolor muy intenso (Pardo et al., 2008).

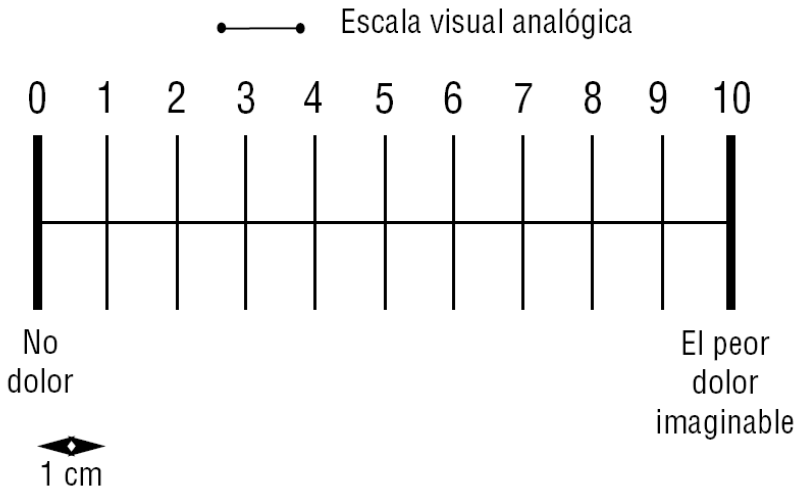

Figura 1. *Esquema simple de la Escala Visual Analógica: de 1 a 3 dolor leve-moderado, de 4 a 6 dolor moderado-grave y más de 6 dolor muy intenso. Figura adaptada de (Pardo et al., 2008).*

- Escala Verbal Numérica

  La Escala Verbal Numérica (EVN) constituye una forma de identificación del dolor mediante la asignación verbal de un número del 0 al 10. La ausencia de dolor corresponde al número 0 y el mayor dolor soportable por el paciente corresponde al 10. La EVN muestra una alta correlación con la EVA, pero con una menor proporción de pacientes que no responden a la evaluación (2 % en comparación con el 11 %). Por esta razón, se considera una de las herramientas más útiles para la valoración del dolor en pacientes en estado crítico (Pardo et al., 2008).

En pacientes no comunicativos, como personas sedadas o en coma, no podemos contar con las habilidades comunicativas del paciente, por lo que se recurren a herramientas indirectas. En este caso la más utilizada es la Escala de Conductas Indicadoras de Dolor (ESCID).

La ESCID es un criterio de valoración creado en España. Como vemos en la Tabla 1, clasifica el dolor basándose en cinco ítems: expresión facial, el tono muscular, la adaptación a la ventilación mecánica, la tranquilidad del paciente y la confortabilidad. Cada indicador toma un valor de 0 al 2, siendo 0 indicador de bajo dolor y 2 de alto. Finalmente se suman las puntuaciones de los cinco ítems, pudiéndose obtener una puntuación final del 0 al 10 (Navarro, 2023).

***Tabla 1**. Criterios de puntuación de la Escala de Conductas Indicadoras del Dolor. Datos adaptados de (Navarro, 2023).*

INDICADOR	DESCRIPCIÓN	PUNTUACIÓN
Expresión facial	Relajada	0
	Tensión, ceño fruncido, gesto de dolor	1
	Ceño fruncido siempre o dientes apretados	2
Tranquilidad	Tranquilo, relajado, movimientos normales	0
	Movimientos ocasionales de inquietud o posición	1
	Movimientos frecuentes, incluyendo cabeza y extremidades.	2
Tono muscular	Normal	0
	Aumentado, flexión de dedos de manos o pies	1
	Rígido	2
Adaptación a ventilación mecánica	Tolerando ventilación mecánica	0
	Tose, pero tolera la ventilación mecánica	1
	Lucha con el respirador	2
Confortabilidad	Tranquilo, confortable	0
	Se tranquiliza al tacto o con la voz, fácil de distraer	1
	Difícil de confortar al tacto o hablándole	2
	PUNTUACIÓN FINAL	

1.3 Sedación

La sedación es un procedimiento rutinario en el área de cuidados intensivos, ya que permite una mejor adaptación del paciente a tratamientos que pueden resultar incómodos. Los fármacos sedantes pueden administrarse bien por vía endovenosa, o bien por vía inhalatoria, siendo la primera la más utilizada (Caballero López et al., 2020).

El uso de sedantes en estos pacientes tiene como objetivo proporcionar confort, reducir la ansiedad y prevenir complicaciones derivadas de la agitación. En algunos casos, también contribuye a disminuir el consumo de oxígeno, favoreciendo la interacción con el ventilador. Sin embargo, su empleo prolongado puede generar efectos adversos, como mayor tiempo de ventilación mecánica, estancias hospitalarias prolongadas y un aumento en la incidencia de delirium, trastornos depresivos y estrés postraumático (Olmos et al., 2019).

1.3.1 Fármacos empleados en sedación

Existen numerosos estudios sobre el uso de diversos sedantes en el ámbito de la medicina intensiva, y a pesar de ello, en general, ningún fármaco sedante es claramente superior a todos los demás (Reade & Finfer, 2014).

En la práctica clínica, se dispone de diversos tipos de sedantes en función de la vía de administración. Dentro de los agentes sedantes intravenosos encontramos las benzodiacepinas (BDZ), el propofol, los agonistas α-2 adrenérgicos y los barbitúricos (Estébanez-Montiel et al., 2008).

- Benzodiacepinas

  Las BDZ son los fármacos más utilizados para la sedación de pacientes con ventilación mecánica. Su mecanismo de acción se basa en la inhibición del sistema nervioso central (SNC) mediada por el complejo receptor GABA, que regula un canal de cloro en la membrana celular. Al incrementar el flujo de cloro hacia la célula, las células nerviosas se hiperpolarizan y el umbral de excitabilidad aumenta (Mattia et al., 2006). De esta forma se consigue efectos hipnóticos, ansiolíticos, anticonvulsivantes y relajantes musculares. En la actualidad, los médicos de la UCI intentan reducir el uso de BDZ, ya que pueden prolongar el coma inducido (Bels et al., 2023).

  Dentro de este grupo encontramos distintos fármacos empleados en la sedación de pacientes críticos como, por ejemplo:

  - Midazolam: de acción rápida (5 minutos) y de corta duración (2 horas). Es un fármaco hidrofílico que se vuelve liposoluble en sangre. Se metaboliza en el hígado y se excreta por los riñones. Actualmente es la BDZ más comúnmente utilizada para la sedación moderada (Bels et al., 2023; Mattia et al., 2006).
  - Lorazepam: inicio más lento, pero interacciona menos con otros medicamentos. Su acción lenta hace que el lorazepam sea menos útil para el tratamiento de la agitación aguda. La vida media es de 12 a 15 horas, por lo que no es fácil valorar una infusión (Cooke et al., 2002).

- Propofol

El propofol es un medicamento de acción corta que produce una disminución del nivel de consciencia y con propiedades hipnóticas. Se cree que actúa, al menos en parte, a través de un receptor de GABA (Bels et al., 2023). Es altamente liposoluble y atraviesa rápidamente la barrera hematoencefálica, con la consiguiente rápida acción y redistribución periférica (aproximadamente de 1 minuto). Gracias a ello, el propofol permite una rapidez de recuperación superior al midazolam, incluso en caso de sedación prolongada de más de 72 h. Diversos estudios clínicos han confirmado eficacia del propofol para la sedación del paciente en la UCI (Barr et al., 2013).

- Agonistas alfa-2

Los agonistas alfa-2 tienen una variedad de efectos, entre ellos sedación, analgesia y ansiolíticos. Estos fármacos suelen utilizarse para la sedación de pacientes adultos que requieran un nivel de sedación no más profundo que despertarse en respuesta a la estimulación verbal. Dentro de ellos destacan la clonidina y la dexmedetomidina, siendo este último el más empleado (Bels et al., 2023).

  - Clonidina: actúa estimulando los receptores adrenérgicos alfa-2 presinápticos en el tronco encefálico, lo que reduce la liberación de noradrenalina y potencia la actividad parasimpática.
  - Dexmedetomidina: agonista alfa-2 adrenérgico con un mecanismo de acción particular. Proporciona sedación y ansiolisis a través de receptores en el locus coeruleus, analgesia mediante receptores en la médula espinal y modula la respuesta al estrés, todo esto sin causar una depresión respiratoria significativa.

- Barbitúricos

Los barbitúricos son un grupo de fármacos derivados del ácido barbitúrico que actúan como depresores del SNC. Dependiendo de la dosis y del tipo, pueden provocar efectos que van desde sedación leve hasta anestesia general profunda. Este grupo es el menos empleado debido a la cantidad de efectos adversos (depresión respiratoria grave, hipotensión y depresión miocárdica, riesgo elevado de sobredosis y dependencia) (Pérez-Bárcena et al., 2005)

En cuanto a la sedación inhalatoria, se utilizan fármacos halogenados en estado líquido, los cuales, mediante un vaporizador, pasan a estado gaseoso (Gil-Castillejos et al., 2025). Podrían ser los sedantes ideales para la UCI, debido a su fácil eliminación pulmonar, su escasa metabolización, su efecto broncodilatador y sus efectos cardioprotectores. Sin embargo, no se utilizan ampliamente ya que la mayoría de los ventiladores modernos de esta unidad no se adaptan fácilmente a los vaporizadores (Estébanez-Montiel et al., 2008).

Dentro de este grupo encontramos al sevoflurano. Se trata de un fármaco anestésico utilizado tanto en adultos como en niños. El mecanismo de acción no se comprende completamente, pero se sabe que actúan como moduladores positivos del receptor GABA y el receptor de glicina, lo que produce inhibición neuronal y sedación. A diferencia de otros sedantes, también actúan como antagonistas de receptores excitadores, lo que les da propiedades únicas y puede prevenir la tolerancia y la dependencia (Gil-Castillejos et al., 2025).

El metabolismo es principalmente pulmonar, con poca acumulación en los tejidos. El sevoflurano tiene un metabolismo hepático del 5%, lo que puede aumentar los niveles de flúor inorgánico en el paciente. En UCI, el uso de sevoflurano debe limitarse a 5 días, con monitoreo de la función renal y los iones en sangre, según las recomendaciones de la Sociedad Española de Medicina Intensiva, Crítica y Unidades Coronarias (SEMICYUC) (Caballero López et al., 2020).

1.3.2 Evaluación del nivel de sedación en pacientes críticos

Para minimizar estos riesgos, diversos estudios han demostrado que la implementación de protocolos de sedación mejora los resultados clínicos. Estas guías establecen evaluaciones periódicas del nivel de sedación mediante escalas validadas, lo que permite ajustar las dosis de los fármacos de manera precisa y evitar tanto la infra como la sobre sedación, optimizando así la recuperación del paciente (Olmos et al., 2019). A continuación, se muestran algunos de los métodos más utilizados:

- Escala de sedación-agitación

  La escala de sedación-agitación o *Sedation-Agitation Scale* (SAS) se desarrolló para evaluar la consciencia y la agitación de los pacientes adultos ingresados en la UCI. Esta escala puntúa el nivel de consciencia y agitación de un paciente a partir de una lista de siete ítems que describen el comportamiento (Tabla 2).

  La utilidad de la SAS es limitada debido a sus inadecuadas pruebas de validez y fiabilidad, a que combina más de un aspecto de la evaluación de la sedación en un solo ítem y a que se centra únicamente en dos aspectos de la sedación (conciencia y agitación). Sin embargo, el mayor número de niveles de la SAS permite un cambio más gradual de los niveles de consciencia y agitación a lo largo de la escala (Chulay, 2004).

***Tabla 2**. Escala de agitación-sedación. Datos adaptados de (Frade Mera et al., 2009)*

ESCALA DE SEDACIÓN-AGITACIÓN		
7	Agitación peligrosa	Arrancándose el tubo endotraqueal, tirando de los catéteres, agrediendo al personal.
6	Muy agitado	No está tranquilo, no se calma a la orden verbal, requiere sujeción física, muerde el tubo.
5	Agitado	Ansioso, pero se tranquiliza con las ordenes verbales.
4	Tranquilo y cooperador	Tranquilo, se despierta con facilidad al tacto o voz, obedece órdenes sencillas.
3	Sedado	Tendencia al sueño. Se despierta con ordenes verbales, pero se vuelve a dormir. Responde a ordenes sencillas.
2	Muy sedado	Responde a estímulos físicos, incapaz de comunicarse u obedecer órdenes, tiene movimientos espontáneos.
1	Arreactivo	Mínima o nula respuesta al dolor, no se comunica ni obedece órdenes.

Tabla 3. *Escala De Sedación-Agitación De Richmond. Datos adaptados de (Frade Mera et al., 2009).*

ESCALA DE SEDACIÓN-AGITACIÓN DE RICHMOND		
-5	No despertable	No responde a la voz ni a estímulos físicos
-4	Sedación profunda	Se mueve o abre los ojos a estímulos físicos, no a la voz
-3	Sedación moderada	Movimientos de apertura de ojos a la voz, no dirige mirada.
-2	Sedación ligera	Despierta a la voz, mantiene contacto visual menos de 10 segundos
-1	Somnolencia	Se mantiene despierto más de 10 segundos, pero no está alerta
0	Despierto y tranquilo	
1	Inquieto	Ansioso, sin movimientos desordenador, ni agresivo ni violento
2	Agitado	Se mueve de forma desordenada, lucha con el respirador
3	Muy agitado	Agresivo, se intenta arrancar tubos y catéteres
4	Combativo	Violento, representa un riesgo para el personal

- Escala de Agitación-Sedación de Richmond

La escala de agitación-sedación de Richmond (*RASS*, por sus siglas en inglés), examina la conciencia y la agitación a través de 10 niveles (Tabla 3). Estos niveles van de -5 (sedado profundamente y no despertable), a 0 (despierto y tranquilo), a +4 (agitado) (Chulay, 2004).

- Escala de Coma Glasgow

La Escala de Coma de Glasgow (*Glasgow Coma Scale*, *GCS*) es una herramienta utilizada para evaluar de manera objetiva el nivel de conciencia en pacientes con traumatismos o enfermedades graves (Jain & Iverson, 2025). Esta escala se basa en la valoración de tres parámetros:

- Respuesta ocular: se puntúa de 1 a 4, donde 1 indica ausencia de respuesta y 4 corresponde a la apertura ocular espontánea o en respuesta a estímulos adecuados.

- Respuesta verbal: se evalúa de 1 a 5, asignando 5 puntos a los pacientes que están completamente orientados y 1 punto a aquellos que no presentan respuesta verbal.

- Respuesta motora: se califica de 1 a 6, donde 1 representa la ausencia total de respuesta a estímulos físicos, mientras que 6 indica que el paciente obedece órdenes con movimientos voluntarios.

  En la Tabla 4, podemos observar la puntuación total de la GCS. Este resultado se obtiene sumando las evaluaciones individuales de cada parámetro, proporcionando una valoración rápida y estandarizada del estado neurológico del paciente. Finalmente, la puntuación varía entre 3 y 15, siendo 3 el nivel más bajo de conciencia y 15 el más alto (Jain & Iverson, 2025).

***Tabla 4**. Escala Glasgow. Datos adaptados de (Jain & Iverson, 2025)*

INDICADOR	DESCRIPCIÓN	PUNTUACIÓN
Respuesta ocular	No abre los ojos	1
	Tras estímulo en la punta del dedo	2
	Tras dictar una orden	3
	Abre antes del estímulo	4
Respuesta verbal	No proporciona ningún tipo de respuesta	1
	Solo gemidos o quejidos	2
	Palabras sueltas inteligibles	3
	No está orientado, pero se comunica perfectamente	4
	Da correctamente el nombre, lugar y fecha	5
Respuesta motora	No hay ningún tipo de movimiento	1
	Extiende el brazo	2
	Dobla brazo sobre codo	3
	Dobla brazo sobre codo rápidamente	4
	Lleva la mano por encima de la clavícula al estimularse el cuello	5
	Obedece la orden con ambos lados	6
	PUNTUACIÓN FINAL	

Estas escalas permiten llevar un control adecuado sobre la sedación basándose en la observación del comportamiento del paciente, evaluando respuestas motoras y verbales a estímulos. La principal desventaja de estos métodos de evaluación es que, como vemos, son subjetivos y muchas veces, valoran más la respuesta a estímulos dolorosos que la propia sedación en sí misma (Chamorro et al., 2008).

Además, estas escalas son insensibles a pacientes en tratamiento con bloqueantes neuromusculares. Estos fármacos paralizan los músculos esqueléticos, impidiendo cualquier movimiento voluntario o reflejo y pudiendo dar una falsa impresión de sedación profunda. En estos casos, y en pacientes críticos con sedación profunda, es necesario otros tipos de métodos de monitorización objetivos. Dentro de este campo, el más utilizado en el índice Biespectral (BIS).

El BIS es una tecnología basada en el análisis del EEG que convierte la actividad cerebral en un índice numérico, lo que facilita la interpretación del estado de consciencia del paciente. Este parámetro asigna un valor entre 0 y 100, donde 90-100 indica que el paciente está completamente despierto, 60-70 corresponde a una sedación ligera, 40-60 representa un nivel adecuado para anestesia y valores por debajo de 40 indican sedación profunda o inconsciencia (Navarro Suay et al., 2016).

Para su funcionamiento, se utilizan cuatro electrodos en la frente del paciente, los cuales registran la actividad eléctrica cerebral. Esta información es procesada por un algoritmo especializado que traduce las señales del EEG en un valor numérico que indica el nivel de sedación. Esta tecnología facilita el monitoreo en tiempo real del estado del paciente, eliminando la necesidad de interpretar señales eléctricas complejas.

Uno de los principales inconvenientes del BIS radica en que, tanto su algoritmo de procesamiento, como el equipo utilizado, son de carácter propietario. Esto significa que el método exacto mediante el cual se transforma la señal del EEG en un valor numérico de sedación no es accesible públicamente. La falta de transparencia en el funcionamiento del algoritmo impide una comprensión detallada de los criterios que emplea para generar el índice, lo que representa una limitación importante desde el punto de vista científico y clínico. En consecuencia, los resultados obtenidos no pueden ser validados ni reproducidos de forma independiente, lo que dificulta su comparación con otros métodos de monitoreo.

En pacientes normales, la puntuación del BIS es una medida fiable de la sedación y actualmente se utiliza de forma rutinaria en los quirófanos. Según el estudio *Bispectral Index monitoring correlates with sedation scales in brain-injured patients* existe una correlación estadísticamente significativa entre los valores del BIS y las puntuaciones de la *RASS*, la *SAS* y la *GCS* en pacientes con lesión cerebral en estado crítico, con y sin sedación (Deogaonkar et al., 2004).

1.4 Machine Learning

La Inteligencia Artificial o IA es la tecnología que aporta a los ordenadores la capacidad para imitar el comportamiento humano (Monostori, 2019). De esta forma, estos sistemas adquieren la capacidad de percibir, relacionarse con el entorno, siendo capaces de resolver problemas. Hoy en día, la IA se puede considerar un campo de la ingeniería que implementa herramientas innovadoras para resolver desafíos complejos.

El *ML* es una rama de la IA que permite a los sistemas aprender sin que haya sido explícitamente programado para ello (Mahesh, 2020). Con la creciente disponibilidad de grandes volúmenes de datos, la demanda de aprendizaje automático ha aumentado significativamente. Muchas industrias lo aplican para extraer información relevante y optimizar procesos. Su implementación en una amplia gama de aplicaciones ha impactado profundamente tanto en la ciencia como en la sociedad, abarcando casi todos los dominios científicos (Qiu et al., 2016).

1.4.1 Clasificación del Machine Learning

Los algoritmos de *ML* se dividen en cuatro categorías principales en función de cómo el modelo aprenda de los datos: supervisados, no supervisados, semisupervisado y por refuerzo. Cada uno de ellos tiene sus ventajas e inconvenientes, pudiendo utilizarlos para aplicaciones muy específicas.

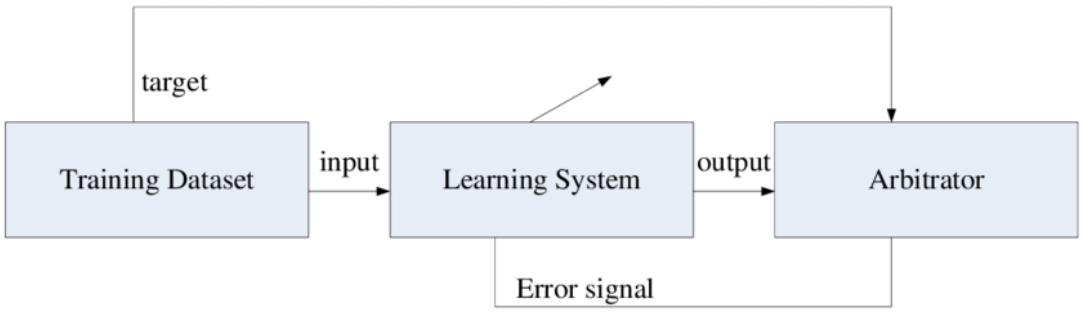

Figura 2. *Diagrama de bloques que explica el funcionamiento del aprendizaje supervisado. Figura adaptada de (Sapon et al., 2011).*

1.4.1.1 Aprendizaje supervisado

El aprendizaje supervisado (*Supervised Learning*) es una técnica de *ML* que utiliza conjuntos de datos previamente etiquetados para entrenar algoritmos. El funcionamiento se puede explicar en varias etapas, como muestra la Figura 2. El proceso comienza con un conjunto de datos de entrenamiento que contiene pares de entrada y salida esperada (*target*). Las entradas son suministradas al sistema de aprendizaje, el cual genera una salida (output) basada en sus parámetros actuales. Esta salida es evaluada por el componente llamado "*arbitrator*", que la compara con la salida esperada (*target*) para calcular la diferencia entre ambas, conocida como la señal de error. Esta señal se retroalimenta al sistema de aprendizaje para ajustar sus parámetros, permitiéndole mejorar su precisión a lo largo del tiempo mediante un proceso iterativo.

Dentro del aprendizaje supervisado, encontramos dos categorías en función de la salida u objetivo del algoritmo (Nasteski, 2017):

- Algoritmos de clasificación, cuyo objetivo es organizar los datos según diferentes categorías. Analizan los datos de entrenamiento, identifica patrones y determina la clase a la que pertenece dicha instancia de entre un conjunto finito de categorías discretas. La clasificación puede ser binaria (solo hay 2 categorías posibles) o multiclase (varios grupos). La salida que proporcionan es una variable discreta (etiqueta).
- Algoritmos de regresión, cuyo objetivo es crear modelos capaces de estimar una variable continua a través de las relaciones entre diversas variables y dicha variable objetivo. En este caso, la salida predicha es un valor real.

Algunos de los ejemplos de algoritmos de aprendizaje supervisado más utilizados en la práctica son:

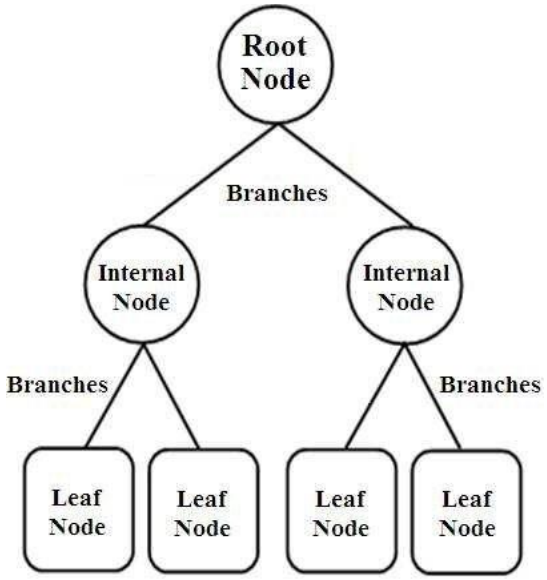

***Figura 3**. Estructura de un árbol de decisión. Figura adaptada de (Sá et al., 2016)*

- Árboles de decisión: se trata de un tipo de algoritmo de aprendizaje supervisado, utilizado tanto para clasificación como para regresión. Su estructura se asemeja a la de un árbol, de ahí su nombre. En la Figura 3 podemos ver la estructura de este algoritmo y sus partes: un nodo raíz, ramas, nodos internos (también llamados nodos de decisión, donde se hacen preguntas sobre los datos) y las hojas (asignan una categoría o una probabilidad). Esta organización permite estructurar la información de manera jerárquica. Los árboles de decisión se utilizan para la clasificación cuando la variable es categórica y para regresión cuando la variable es continua (Nasteski, 2017; Ray, 2019).

  Para mejorar el rendimiento predictivo y la robustez de los árboles de decisión, se desarrollaron técnicas de ensamblado. Dos de estos métodos son la agregación de *bootstrap* (*bagging*) y la familia de algoritmos *Adaboost* (*boosting*).

  En el *bagging*, se crean distintos subconjuntos de entrenamiento con una réplica Bootstrap del conjunto de entrenamiento original. A cada subconjunto se le aplica el mismo algoritmos y finalmente se combinan las predicciones de todos los subconjuntos para obtener una predicción final. Un ejemplo destacado de esta técnica es el algoritmo *Random Forest.* Se trata de un conjunto de árboles de decisión que permite reducir el sobreajuste u *overfitting*. Cada árbol que forma parte del algoritmo se construye a partir de una muestra extraída con reemplazo de los datos de estada. La idea principal se basa en la creación de un conjunto de árboles a través de distintos subconjuntos de muestras y luego combinar los resultados para obtener una predicción final (Machova et al., 2006; Mienye & Jere, 2024).

  En el *boosting*, en vez de entrenar los modelos paralelamente, se entrenan secuencialmente. Cada nuevo árbol se construye prestando especial atención a los ejemplos que el anterior clasificó mal, asignándoles mayor peso. De esta manera, se favorece la corrección de los errores cometidos. El algoritmo de *boosting* más utilizado es *AdaBoost*. En este caso, se ajusta automáticamente los parámetros a los datos basándose en el rendimiento de la iteración actual, de modo que el conjunto

corrige sucesivamente los errores anteriores y reduce el sesgo global del ensamblado (Mayr et al., 2014).

- Regresión logística (RL): es un algoritmo que identifica la relación entre una variable dependiente categórica y una (o varias) variables independientes, permitiendo clasificar los datos en distintas categorías. Para ello, estima la probabilidad de que ocurra un evento en función de un conjunto de variables independientes (Peng et al., 2002).
- Análisis del discriminante lineal (LDA): se basa en la proyección de los datos en un espacio de menor dimensión. Para ello, primero calcula la varianza entre clases (distancia entre las medias de las clases). Después, calcula la varianza intraclases (distancia entre la media y las muestras dentro de cada clase). Por último, se proyectan los datos en un espacio donde se maximice la distancia entre clases y se minimice la distancia intraclases (Tharwat et al., 2017).
- *K*-Nearest Neighbors (KNN): se trata de uno de los algoritmos de aprendizaje supervisado más utilizados en la minería de datos. El algoritmo durante la fase de entrenamiento almacena todos esos datos y determina el parámetro *k* (el número de vecinos que se va a tener en cuenta para la predicción). Durante la fase de predicción, calcula las distancias entre el nuevo dato que se quiere clasificar y todos los datos almacenados anteriormente. KNN busca los *k* puntos más cercanos en el conjunto de entrenamiento. La clase asignada al ejemplo que se quiere clasificar será la clase que más se repita en esos *k* ejemplos (Martin & Idoate, 2015; Zhang, 2022).
- Máquinas vector soporte (*SVM*): es un algoritmo tanto de clasificación como de regresión. El algoritmo aprende a definir una frontera de decisión, separando a dos clases distintas. En el caso de que solo haya una clase, la *SVM* crea un límite alrededor de los datos de entrenamiento sin necesidad de información sobre lo que hay fuera de esa frontera. Para lograr una mejor separación entre clases, los datos se transforman mediante un kernel a un espacio de características de mayor dimensión, como vemos en la Figura 4. En este nuevo espacio, el algoritmo busca la máxima separación entre las clases. Después, la *SVM* transforma esa separación nuevamente al espacio original, logrando clasificar los datos en distintos grupos bien organizados (de la Hoz Manotas et al., 2013).

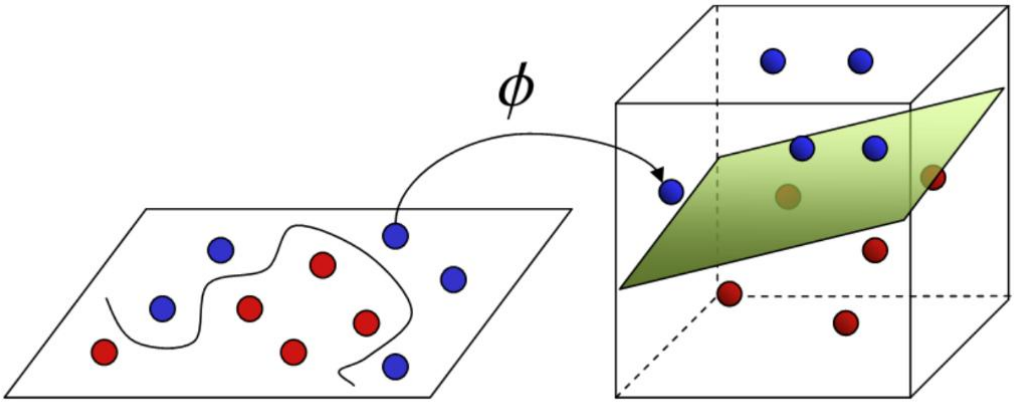

Figura 4. *Representación algoritmo SVM. Figura adaptada de (de la Hoz Manotas et al., 2013)*

- Redes neuronales: son modelos de aprendizaje automático inspirados en el funcionamiento del sistema nervioso de los seres vivos. Están compuestas por nodos o neuronas artificiales que procesan información de manera interconectada, permitiendo el análisis de datos y la generación de respuestas. En el cerebro, las neuronas se organizan en tres componentes fundamentales: los receptores, que captan estímulos del entorno o del interior del organismo; el sistema nervioso, encargado de procesar y almacenar la información; y los órganos efectores, que transforman esta información en respuestas físicas o químicas, como el movimiento de los músculos. Siguiendo esta organización, las redes neuronales artificiales también se estructuran en tres niveles de capas. La capa de entrada, que recibe los datos iniciales; la capa oculta, que procesa la información a través de una o más capas intermedias; y la capa de salida genera el resultado final.

1.4.1.2 Aprendizaje no supervisado

El aprendizaje no supervisado o *unsupervised learning* es un tipo de *ML* que analiza datos no etiquetados, identificando patrones, estructuras o relaciones ocultas dentro de los mismos. En este caso, a diferencia del aprendizaje supervisado, este enfoque no requiere que una persona proporcione respuestas correctas o etiquetas previas; es decir, los modelos aprenden directamente de los datos sin supervisión humana, extrayendo conocimiento sin intervención externa. El objetivo de este modelo es encontrar grupos de datos con características similares, para lo que utilizan una medida de similitud. Generalmente, podemos subdividir los algoritmos de aprendizaje no supervisado en dos clases:

- Algoritmos de *clustering*: agrupa la información de forma autónoma formando grupos llamados *clusters*. Se debe definir una medida de similitud. Generalmente, se utiliza la medida de la distancia entre los datos (Naeem et al., 2023). Un ejemplo claro de esta clase es el algoritmo *K-means.* Se trata de un método iterativo utilizado para dividir un conjunto de datos en un número determinado de *clusters*. Para aplicarlo, primero se debe definir la cantidad de *clusters* deseados (*K*) y luego inicializar los centroides, es decir, seleccionar *K* puntos como los centros iniciales de cada grupo. Este algoritmo consta de dos fases principales: la fase de asignación, en el que cada punto del conjunto de datos se asigna al *cluster* cuyo centroide esté más cercano, utilizando la distancia euclídea como métrica; la fase de actualización, en la que se recalculan los centroides de los *clusters* en función de la nueva distribución de los puntos asignados. El proceso se repite de manera iterativa hasta que los centroides dejan de cambiar significativamente o se alcanza el número máximo de iteraciones (Oti et al., 2021).
- Los algoritmos de reducción de dimensionalidad se suelen utilizar para disminuir el tiempo de entrenamiento de los modelos y reducir así el coste computacional. El algoritmo más utilizado se llama Análisis de Componentes Principales (PCA, por sus siglas en inglés). Se trata de un método que transforma un conjunto de datos de alta dimensión en un nuevo conjunto de variables llamadas Componentes Principales (PCs). Estas nuevas variables son funciones lineales del conjunto de datos originales y están ordenadas de manera que el primer PCs captura la mayor cantidad posible de varianza en los datos, el segundo captura la siguiente mayor cantidad de varianza, y así sucesivamente. Finalmente, se puede reducir la cantidad de variables (dimensionalidad) descartando las PCs con menos información (menos variabilidad).

Esto permite simplificar los datos minimizando la información relevante perdida (Eckhardt et al., 2023; Sánchez Mangas, 2012).

1.4.1.3 Aprendizaje semisupervisado

El aprendizaje semi supervisado es una combinación de los anteriores. En este caso, disponemos de un conjunto habitualmente pequeño de datos de entrada etiquetados y otro conjunto de datos más grande de datos no etiquetados.

Primero, se entrena un modelo utilizando un conjunto de datos etiquetados. Una vez tenemos el modelo entrenado, se incorporan nuevos datos sin etiquetar para que el modelo prediga las etiquetas de dichos datos, generando así datos pseudo-etiquetados. Por último, se combinan los datos etiquetados originales con los pseudo-etiquetados para reentrenar el modelo, obteniendo un nuevo modelo mejorado.

1.4.1.4 Aprendizaje por refuerzo

En el aprendizaje por refuerzo, el modelo aprende sobre la marcha gracias a las interacciones de un agente con el entorno. Aunque esto se parece al aprendizaje supervisado, hay una diferencia clave: la retroalimentación no es una etiqueta o valor de verdad absoluta, sino una medida de qué tan buena fue la acción según una función de recompensa. A través de ensayo error o planificación, el agente aprende qué acciones tomar para maximizar su recompensa y mejorar con el tiempo (Raschka & Mirjalili, 2019).

1.4.2 Metodología para el desarrollo de sistemas de Machine Learning basados en Feature Engineering

En esta sección se detalla una metodología centrada en el desarrollo de sistemas de *ML* utilizando técnicas de *feature engineering*. Si bien existen otros enfoques como el *Deep Learning*, que han demostrado un gran potencial en múltiples áreas, su aplicación efectiva requiere habitualmente grandes volúmenes de datos y una infraestructura computacional significativa. En el presente trabajo, se ha optado por una aproximación basada en *feature engineering* debido a la disponibilidad limitada de datos y recursos, lo cual permite construir modelos robustos y eficientes a partir de técnicas más interpretables y manejables.

El desarrollo de un sistema de *ML* implica una serie de etapas estructuradas que garantizan la calidad de los datos, la eficiencia del modelo y su capacidad de generalización. A continuación, se describen los pasos genéricos que habitualmente se utilizan para el preprocesado anterior a construir un sistema de *ML* utilizando técnicas de *feature engineering*:

I. Adquisición de los datos

Deberemos crear un *dataframe*, es decir, una matriz de datos donde cada fila corresponde a cada muestra (en este caso, un paciente) y cada columna a cada una de las variables bajo estudio, o características. Una correcta organización inicial facilita todo el proceso posterior de análisis y modelado.

II. Preprocesado de los datos

Cuando trabajamos con conjuntos de datos muy grandes, debemos verificar que nuestro *dataframe* esté completo, estructurado y adecuado para asegurar el rendimiento óptimo del algoritmo. El rendimiento del modelo depende en gran medida de la calidad de los datos utilizados durante el entrenamiento. Para ello es necesario preprocesarlos. A continuación, vamos a detallar todos los pasos que se realizan en el preprocesado (Ashmore et al., 2022):

- Limpieza de los datos: identificación y corrección de datos anómalos. Pueden ser valores inconsistentes, nulos o duplicados.
- Codificación de variables categóricas: para que el algoritmo pueda trabajar con este tipo de características, es necesario convertir las variables categóricas a formato numérico. Dentro de estas técnicas encontramos la codificación *One-Hot* o la codificación *Label Encoding.*
- Imputación de los datos: consiste en el proceso completar la base de datos, es decir, de sustituir los valores *missing* o faltantes con valores conocidos a través de técnicas como el algoritmo de KNN o con la media o la mediana del resto de datos de la característica.
- Normalización y estandarización de variables numéricas. En este proceso, se transforman los datos para que estos tengan unas características determinadas. Una opción habitual es escalar los datos para que se encuentren en un rango entre 0 y 1. Otra opción cuando se dispone de datos continuos es transformar los datos para obtener una distribución de media 0 y desviación típica 1 (z-score).

III. Extracción de características

Consiste en la creación de nuevas características o variables a partir del conjunto de características originales. En este estudio, se utilizaron datos tabulados, es decir, no fue necesaria una etapa de extracción propiamente dicha, sino que se utilizaron directamente los datos obtenidos tras el preprocesado.

IV. Selección de características

Una vez preprocesados los datos, se pasa a la etapa de identificación y selección de características relevantes. El proceso de selección de características proporciona únicamente características únicas que contribuyen en mayor medida a los resultados de la predicción, eliminando el ruido y las características irrelevantes. Este proceso busca mejorar la representación de los datos para facilitar el aprendizaje del modelo. (Htun et al., 2023; Murel Ph.D. & Kavlakoglu, 2024).

Dentro de este grupo, encontramos distintas técnicas de selección de características:

- Métodos de filtrado o *filter*: clasifican las variables en base a su relevancia para los algoritmos a la hora de hacer las predicciones. Para ello, utilizan criterios de correlación y distancia, como el coeficiente de correlación de Spearman o la correlación de Pearson o la distancia euclídea.
- Métodos de envoltura o *wrapper*: tienen integrada la selección de características dentro del proceso de aprendizaje del propio algoritmo de ML. El objetivo de estos métodos es buscar un subconjunto de características que ofrezcan el mejor rendimiento de la predicción.

- Métodos integrados o *embedded*: combinan las cualidades de los dos métodos anteriores. En este caso, realizan la selección de característica en el proceso de entrenamiento con las herramientas descritas en los métodos de filtrado. Dentro de este grupo encontramos algoritmos como *Random Forest*, SVM y LR.
- Métodos basados en la teoría de la información: utilizan la información mutua para obtener una puntuación sobre la importancia de cada característica.

V. Clasificación / Regresión

Una vez concluidos los pasos anteriores, se procede a construir un modelo predictivo que permita resolver el problema planteado, ya sea de clasificación (asignar categorías) o de regresión (predecir valores numéricos continuos). Este proceso, a la vez, comprende tres subetapas:

- División del conjunto de datos en dos subgrupos aleatoriamente: conjunto de entrenamiento (para entrenar y optimizar el modelo) y conjunto de prueba o test (para evaluar el rendimiento del modelo).
- Entrenamiento y selección de un modelo predictivo: se elige un algoritmo de *ML* apropiado para el tipo de problema y se entrena con los datos de entrenamiento. Cada algoritmo presenta ciertas ventajas y limitaciones. Por ello, es común entrenar varios modelos y comparar sus resultados para seleccionar el que mejor se desempeñe con los datos disponibles.
- Evaluación del modelo: se mide el rendimiento del modelo con métricas estadísticas para determinar su capacidad predictiva. En el caso de que realicemos problemas de clasificación calcularemos la *accuracy*, la Kappa de Cohen, el *F1-score*, entre otras. En caso de tener un problema de regresión, calcularemos el error absoluto medio, el error cuadrático medio o el coeficiente de determinación (Ekundayo, 2020).

VI. Implementación y reevaluación

El modelo se despliega en entornos reales y se monitorea continuamente para evaluar su funcionamiento. Si es necesario, se ajusta y reentrena con nuevos datos para mejorar su desempeño.

1.5 Machine Learning en medicina

La rápida evolución de la medicina hace complicada la tarea de mantenerse actualizado por parte del profesional clínico. Es aquí donde el potencial de la IA resulta una herramienta perfecta para resolver problemas de manera automática en el ámbito médico (Lanzagorta-Ortega et al., 2023).

Dentro de este campo, cabe destacar la funcionalidad del *ML*. Uno de los principales beneficios de su uso es su capacidad para facilitar el diagnóstico de enfermedades. Los algoritmos de *ML* han demostrado una gran capacidad para reconocer patrones y mejorar diagnósticos dudosos en diversas áreas de la salud. En dermatología, una red neuronal artificial clasificó lesiones como benignas o malignas con una precisión similar a la de un comité de 21 dermatólogos (Chen & Asch, 2017). En psiquiatría, el uso de *ML* permitió reducir los criterios diagnósticos del trastorno del espectro autista de 29 a 8, con una precisión del 100 % en 612 pacientes (Wall et al., 2012).

El uso de *ML* también ha permitido el descubrimiento de fármacos y la predicción de su comportamiento en el organismo. Su implementación ha marcado un cambio radical en la industria farmacéutica, optimizando la predicción de toxicidad, la minería del genoma y las aplicaciones quimiogenéticas (Handelman et al., 2018).

En radiología, al manejar principalmente datos digitales, el *ML* ha tenido un gran desarrollo, especialmente en el análisis automatizado de imágenes. Se han desarrollado sistemas de detección asistida por computadora para la segmentación de órganos, la identificación de anomalías y la clasificación de imágenes médicas (Handelman et al., 2018).

Como vemos, la IA puede ser una herramienta valiosa en la práctica clínica. Nos ayuda a reducir el número de errores médicos, lo que conlleva una mejora de la seguridad del paciente, asi como una reducción del gasto hospitalario indirectamente. Sin embargo, debemos ser prudentes a la hora de interpretar y aplicar estos avances. Para que los algoritmo de *ML* sean útiles, deben ser reproducibles en la población general. Los estudios con un número reducido de pacientes, en poblaciones específicas o con sesgos de selección no permiten generalizar sus hallazgos (Paixão et al., 2022).

Uno de los grandes debates que surge con el desarrollo de estas tecnologías es el temor o la posibilidad de reemplazo del personal. Hoy en día, esta afirmación no se ha justificado. Ningún software ha logrado sustituir el juicio clínico y la experiencia subjetiva del médico en la toma de decisiones (Paixão et al., 2022).

Rechazar el progreso tecnológico puede ser tan dañino como depender completamente del aprendizaje automático para el cuidado del paciente. La combinación de *ML* con el juicio clínico ha demostrado mejores resultados que su uso por separado (Paixão et al., 2022).

La implementación de técnicas de *ML* en medicina ha pasado de ser solo una idea, a convertirse en una realidad. Aunque su desarrollo sigue en curso, los estudios han demostrado su aplicabilidad clínica, con un impacto significativo en el diagnóstico y la evaluación del pronóstico de los pacientes (Paixão et al., 2022)

1.5.1 Machine Learning y sedación

En los últimos años, ha crecido el interés por aplicar técnicas de *ML* en la optimización del control de la sedación de forma más individualizada, precisa y basada en datos objetivos. En este ámbito, cabe destacar el estudio realizado por Ramaswamy et al., que investigó la predicción del nivel de sedación utilizando señales de EEG frontal y algoritmos de aprendizaje automático no lineales. El estudio se basó en 204 grabaciones de EEG recogidas de voluntarios sometidos a cuatro sesiones de anestesia con distintas combinaciones de fármacos. El objetivo era comparar el rendimiento de cuatro modelos de clasificación binaria para discriminar entre estados de vigilia y sedación: RL, *SVM* con kernel gaussiano, *Random Forest* y árbol de conjunto con *Bagging*. Este último obtuvo el mejor rendimiento, logrando un área bajo la curva o *AUC* por sus siglas en inglés, superior a 0.85 al utilizar un conjunto amplio de 44 características extraídas del EEG (Ramaswamy et al., 2022).

El estudio realizado por González Rubio et al., en cambio se centró en la automatización de predicción de los distintos niveles de sedación anestésica (sedación ligera, sedación

moderada y sedación profunda) mediante el análisis de señales del EEG utilizando técnicas de IA (González Rubio et al., 2019). Para el estudio se seleccionaron a 27 pacientes sometidos a cirugía y se excluyeron a pacientes con antecedentes de epilepsia, enfermedades cerebrovasculares y otras enfermedades neurológicas Los registros de los datos EEG adquiridos se tomaron desde el canal frontal F4, del cual se extrajeron nueve parámetros espectrales. Sobre estos datos se entrenó un modelo de *ML* basado en *SVM* y otro basado en Sistemas Neuro-Difusos (ANFIS). Los resultados indicaron que el modelo basado en *SVM* logró una precisión del 96,12 % para detectar sedación profunda, 90,06 % para sedación moderada y 90,24 % para sedación ligera, demostrando la eficacia del enfoque propuesto y la idoneidad del canal F4 para la monitorización anestésica (González Rubio et al., 2019).

Uno de los principales inconvenientes que observamos en los estudios es que se llevaron a cabo exclusivamente en voluntarios sanos y en condiciones altamente controladas, lo que limita su aplicabilidad a escenarios como la UCI. Además, aunque el EEG es actualmente una de las herramientas más precisas para monitorizar en tiempo real el nivel de sedación, no todos los centros hospitalarios disponen de los recursos tecnológicos y humanos necesarios para implementar esta técnica de forma sistemática, lo que reduce su aplicabilidad a gran escala en pacientes sedados de forma prolongada.

En este sentido, el presente trabajo fin de grado (TFG) se plantea como una evolución necesaria, al centrarse específicamente en pacientes críticos ingresados en la UCI y al integrar una gran variedad de variables clínicas, fisiológicas y de monitorización que son recogidas de forma habitual en la práctica asistencial.

1.6 Inteligencia Artificial Explicable

Los sistemas de *ML* se han convertido en una herramienta ideal para ayudarnos en diversas tareas del día a día. Sin embargo, muchos de estos modelos no son capaces de explicar exactamente como llegaron a las predicciones finales. Es lo que se conoce como modelos de “caja negra”. En el ámbito médico, esta falta de información puede resultar en modelos poco confiables (Samek & Müller, 2019).

Ante esta situación, comienza a desarrollarse lo que se conoce como inteligencia artificial explicable o *explicable artificial intelligence* (*XAI*). Se trata de técnicas que producen modelos más explicables, al tiempo que mantienen un alto nivel de rendimiento, estudiando la transparencia de los distintos sistemas (Minh et al., 2021).

Un modelo de *XAI* cumple las siguientes características (Solutions, 2023):

1. Transparencia: si son capaces de describir de manera clara los procesos necesarios para realizar las predicciones.

2. Interpretabilidad: la explicación sobre cómo ha realizado esas predicciones debe ser comprensible para los humanos

3. La explicabilidad: capacidad de descifrar el por qué una determinada característica ofrece más o menos información.

A la hora de explicar un modelo, podemos encontrar diferentes técnicas en función del origen de los datos, es decir, en función de si trabajamos con datos tabulares, imágenes o

texto. En el caso del presente trabajo, nos interesan los métodos explicativos de los sistemas de decisión que actúan sobre datos tabulares. A grandes rasgos, encontramos los siguientes métodos (Bodria et al., 2023):

- Métodos basados en la importancia de las características: se trata de uno de los tipos de explicación más populares de *XAI*. En este caso se trata de asignar a cada variable un valor de importancia que representa que tan importante fue esa característica para realizar la predicción. Dentro de este tipo encontramos el método *LIME* (Local Interpretable Model-agnóstic Explanations), *SHAP* (SHapley Additive exPlanations) …
- Métodos basados en reglas: explican las razones que llevan al modelo a dar las predicciones finales a través de reglas lógicas simples del tipo booleanas. Dentro de este grupo encontramos técnicas como *ANCHOR o LORE* (LOcal Rule-based Explainer).
- Explicadores basados en prototipos: un prototipo o artefacto es un registro que destaca la variables o características que identifican a un grupo de objetivos de la misma clase. Los artefactos sirven como ejemplos comparativos, donde el usuario puede entender el modelo de caja negra mirando registros similares al prototipo.
- Explicaciones basadas en contrafácticos: son técnicas que ofrecen posibles sugerencias en cuanto a las entradas del modelo de caja negra para cambiar el resultado de este. Describen si existe alguna dependencia o relación entre dichas entradas y los atributos que llevaron a tomar la decisión final. Pueden considerarse como el complemento a los explicadores basados en prototipos. En lugar de mostrar casos representativos, explican lo que podría suceder su se cambian ciertas condiciones de entrada.

Para que se cumplan las características mencionadas anteriormente y ver si las técnicas utilizadas han sido útiles, deberíamos poder evaluar cuanto de explicable es de modelo. Existen diversos métodos utilizados para juzgar el resultado de los métodos de *XAI*. Es importante remarcar que, hasta e momento, no existen medidas objetivas de evaluación que permitan seleccionar el mejor método explicativo de *XAI*.

- La fidelidad: evalúa la eficacia del modelo sustituto en imitar el comportamiento del modelo original.
- La estabilidad: evalúa si instancias similares obtienen explicaciones similares. Un parámetro utilizado para ello suele ser la constante de *Lipschitz.*
- La eliminación y la inserción: métricas que eliminan algunas de las variables que el modelos de explicación considero importante y observa como varía el rendimiento del modelo de caja negra. Con estos métodos lo que se busca es que, si eliminamos la característica más importante, el modelo de caja negra se verá obligado a cambiar su decisión.
- La monotonía: es un método de inserción el cual estudia como varía el modelo de caja negra al ir añadiendo cada característica en orden de relevancia. Se espera que el rendimiento aumente al ir añadiendo cada característica, consiguiendo un aumento monótono del rendimiento del modelo
- Tiempo de ejecución: tiempo necesario para producir la explicación.

1.7 Hipótesis y objetivos

En el presente trabajo se parte de la hipótesis de que *es posible automatizar la clasificación del paciente crítico en función del nivel de sedación*.

En base a esta hipótesis, el objetivo general del estudio es *desarrollar un modelo predictivo explicable que permita estimar de forma automática la clasificación de los pacientes en tres grupos: infrasedados, sobresedados y con sedación adecuada*. Para ello, se analizarán distintas variables clínicas, fisiológicas y demográficas a través diversas técnicas de *ML* supervisado. Además, de manera paralela, se identificarán las características más relevantes para la predicción de dichas clases a través de técnicas de *XAI*.

Para poder alcanzar este objetivo, es necesario alcanzar los siguientes objetivos específicos: (i) obtener una base de datos clínicos de pacientes críticos bajo efectos de fármacos analgo-sedantes, (II) desarrollar y aplicar técnicas de depurado de datos y selección de características para mejorar la calidad de los datos y poder obtener mejores resultados, (iii) implementar y evaluar distintos modelos de aprendizaje supervisado, como *LDA*, RL y *AdaBoostM2* y (iv) aplicar técnicas de *XAI* basadas en la importancia de características para proporcionar información sobre la variable más relevante a la hora de predecir el estado de sedación de los pacientes.

1.8 Descripción del documento

A lo largo de este documento se explicará en detalle el estudio planteado para demostrar la hipótesis anterior. En este capítulo, se encuentra una breve introducción donde se explican conceptos claves de la analgesia, sedación y *ML*. En la capítulo 2, se lleva a cabo la explicación de los materiales y métodos del estudio. En él se hablará de los distintos procedimientos llevados a cabo de manera cronológica, así como las herramientas utilizadas para ello. En el capítulo 3, se muestran todos los resultados obtenidos a lo largo del estudio, con una explicación breve de cada uno de ellos. En el cuarto capítulo se discutirán los resultados obtenidos anteriormente, relacionándolos con conceptos clínicos. Por último, se finaliza el documento con un capítulo de conclusiones, donde se resume las aportaciones claves del proyecto.

Capítulo 2. Materiales y métodos

2.1 Introducción

A lo largo de este capítulo comentaremos los distintos procedimientos realizados a lo largo del trabajo. En el inicio del apartado se describe nuestro estudio y la población seleccionada para ello, explicando los correspondientes criterios de inclusión y exclusión. Seguidamente, hablaremos de la creación de la base de datos que recoge información clínica de pacientes críticos ingresados en UCI. Posteriormente, comentaremos una serie de técnicas empleadas en el preprocesado, enfocadas en la detección e imputación de datos perdidos. A continuación, se realizará un desglose de las técnicas de análisis estadístico empleadas en las variables bajo estudio. Además, se detallará la prueba de selección de características empleada. En los últimos apartados, hablaremos de la creación de los conjuntos de entrenamiento y test, así como de los distintos modelos de *ML* seleccionados para realizar la clasificación de sujetos en cada clase. Para finalizar, se comentará la herramienta de *XAI* utilizada para medir la relevancia de cada característica en nuestro mejor modelo.

2.2 Diseño del estudio

Para demostrar la hipótesis planteada, se realizó un estudio con los datos clínicos de pacientes críticos de la UCI del Hospital Clínico Universitario de Valladolid. Con esta información, se creó la base de datos empleada para el desarrollo del algoritmo de *ML* de clasificación automática para predecir el nivel de sedación.

Se tomaron datos de pacientes ingresados en 2024 y 2025, considerándose un estudio retrospectivo en el tiempo. Pudimos recopilar toda esta información gracias a la aplicación ICCA (*IntelliSpace* Cuidados Críticos y Anestesia) y del programa informático Jimena. Además, pudimos añadir información de los pacientes que estaban actualmente ingresados que cumplían los criterios de inclusión, gracias a la colaboración de los médicos de la unidad.

Puesto que solo realizamos la recopilación y análisis de los datos ya existentes, podemos decir que el presente estudio es de carácter observacional. Además, también podemos clasificarlo como analítico, pues los algoritmos de *ML* tratan de buscar relaciones existentes entre las variables para realizar predicciones.

2.2.1 Análisis de la potencia muestral

A la hora del cálculo óptimo del tamaño muestral, realizamos un análisis estadístico independiente. Se consideraron tres puntos importantes: (i) la probabilidad de error Tipo I (α) debe ser 0.05; (ii) la potencia de la prueba ($1-\beta$) debe ser 0.95; (iii) el tamaño del efecto, basado en el conocimiento de las características clínicas y sociodemográficas, puede estimarse como $f = 0.35$ (es decir, efecto moderado, pues las diferencias entre los grupos son bastante marcadas); y (iv) se asumió una prueba ANOVA de efectos fijos con 3 grupos de análisis (paciente infrasedado, sedación adecuada y sobresedado).

Con estos datos, se ha estimado que el tamaño total de la muestra debería ser de 79 o más. Teniendo en cuenta en torno a un 10% de posibles datos perdidos, se considera que la muestra debe ser de, al menos, 87 pacientes. Para asegurarnos de que la población muestral sería suficiente y tener un buen conjunto de entrenamiento, decidimos seleccionar un total de 100 pacientes.

2.2.2 Criterios de inclusión y exclusión

En el contexto de esta investigación, se establecieron unos criterios de inclusión fundamentales para delimitar la población de estudio y garantizar la validez de los resultados.

Estos criterios incluyen:

- Pacientes mayores de 18 años
- Cualquier motivo de ingreso que lleve al paciente a precisar sedación y /o analgesia en la UCI.
- Uso requerido de BIS. Al comienzo de la investigación, recogimos variables de pacientes sin BIS, pero debido a la importancia clínica de esta variable (hoy en día es la única prueba utilizada en los hospitales para medir el nivel de sedación de los pacientes de forma objetiva) creímos muy relevante la inclusión de este criterio.

Respecto los criterios de exclusión, los profesionales médicos marcaron:

- Pacientes menores de 18 años.
- Embarazo confirmado o lactancia materna.

2.2.3 Base de datos

Dentro de la base de datos utilizada en este trabajo, se encuentran tanto variables clínicas como sociodemográficas. En total se emplearon 29 variables, todas ellas seleccionadas previamente con la ayuda de los médicos implicados en el proyecto.

Las variables seleccionadas fueron las siguientes:

- Edad: años cumplidos del paciente al momento del ingreso en UCI. Se trata de un factor clave a la hora de determinar la dosis adecuada de los distintos fármacos sedantes. Los cambios metabólicos y fisiológicos asociados al envejecimiento, producen una disminución progresiva de la capacidad funcional de los órganos y sistemas. Esto provoca una eliminación más lenta de los sedantes, incrementando a su vez la vida media y los niveles plasmáticos de dichos fármacos (López Jiménez & Giménez Prats, 2004).
- Peso: la dosis administrada de fármacos se realiza en base al peso de los pacientes. Un índice de masa corporal (IMC) alto está a asociado a mayor número de complicaciones relacionadas con la sedación, además de ser un factor de riesgo para diversas enfermedades (Olvera-Martínez et al., 2021).
- Sexo: hombre o mujer.
- Motivo de ingreso (MI): causa principal por la que el paciente requiere ingreso en UCI. Dentro de este grupo encontramos una subclasificación.
    - Shock séptico: manifestación más grave de una infección.
    - Status epiléptico: actividad anormal en la superficie del cerebro que provoca descargas eléctricas neuronales corticales excesivas.

    - Patología respiratoria: en este grupo encontramos a pacientes ingresados por diversas enfermedades que afectan al sistema respiratorio, como pueden ser la insuficiencia respiratoria aguada por EPOC (enfermedad pulmonar obstructiva crónica) o el tromboembolismo pulmonar.
    - Accidente cerebrovascular (ACV): trastorno del SNC, provocado por una alteración de la circulación cerebral. Dentro de este grupo encontramos ACV isquémicos (el caso más común, donde se produce la pérdida del flujo sanguíneo cerebral) y el ACV hemorrágico (donde se produce la rotura de algún vaso y como consecuencia la extravasación de la sangre).
    - Patología cardíaca: trastornos que afectan al corazón; como pueden ser arritmias, insuficiencia cardiaca, enfermedad coronaria, valvulopatías o cardiopatías congénitas.
    - TCE o traumatismo craneoencefálico.
    - Neurocrítico: dentro de este grupo hemos incluido otras causas neurológicas graves como pacientes en coma debido a sobreingesta de medicamentos o drogas o síndrome de Guillain Barre, entre otros.
- Antecedentes clínicos: para describir los posibles factores de riesgo, hemos utilizado variables independientes para cada uno de ellos. A continuación, describimos los principales antecedentes más comunes en los pacientes ingresados en UCI:
    - Dislipemia: alteración de los niveles de lípidos, pueden ser tanto elevados como disminuidos.
    - EPOC: enfermedad obstructiva crónica progresiva pulmonar. Generalmente, se suele presentar como dos procesos, bronquitis crónica o bien enfisema pulmonar.
    - Cáncer: en esta variable recogimos los datos de aquellos pacientes que padecían cáncer (sin especificar qué tipo) en el momento de registro de los datos.
    - Riesgo cardiovascular: dentro de este grupo encontramos distintas categorías, desde fumadores, usuarios de drogas, diversos problemas de corazón o bien combinación de las anteriores.
    - Patología renal: cualquier enfermedad que afecte a los riñones.
    - Diabetes Miellitus (DM): trastorno metabólico que se caracteriza por una hiperglucemia crónica causada por un déficit en la secreción de insulina, una acción alterada de la insulina o ambas. Dentro de este grupo encontramos dos tipos de DM: la diabetes tipo 1 (donde se produce la destrucción autoinmune de las células beta pancreáticas que producen la insulina, provocando la ausencia total de insulina en sangre) y la diabetes tipo 2(la más común, donde en este caso sí que hay insulina en sangre, pero no es utilizada de forma eficaz por las células. Se dice que existe resistencia insulínica).
    - Hipertensión arterial (HTA): elevación excesiva de la presión arterial.
- Días de ingreso: número total de días que el paciente permanece ingresado en UCI hasta el momento de la toma de datos.

- Ventilación mecánica (VM): soporte respiratorio artificial.
    - Ventilación mecánica no invasiva: no requiere intubación, en este caso el soporte se coloca bien sobre la nariz, la boca o la faringe a través de cánulas máscaras o tubos. Dentro de este grupo encontramos el modo CPAP (presión positiva continua en la vía aérea) y BIPAP (presión positiva binivel en la vía aérea).
    - Ventilación mecánica invasiva: se conecta un respirador al paciente a través de un tubo endotraqueal o de una traqueostomía. Según el modo ventilatorio encontramos 2 tipos de configuraciones: la ventilación mandatoria, que puede ser controlada (la máquina realiza completamente el trabajo respiratorio) o asistida/controlada (el ventilador asiste parcialmente la respiración del paciente); la ventilación espontánea o en T, donde el paciente respira por sí mismo con mínima o nula asistencia.
- TAM (Tensión Arterial Media): valor promedio de la presión arterial durante un ciclo cardíaco.
- FC (Frecuencia Cardíaca): número de latidos por minuto (lpm).
- FR (Frecuencia Respiratoria): número de respiraciones por minuto.
- Aminas: fármacos vasoactivos empleados para mantener la presión arterial en rangos adecuados mediante la vasoconstricción o aumento del gasto cardíaco. Entre las más comunes están la noradrenalina y la dopamina. En este caso tomaremos como dato la dosis administrada.
- Fármacos sedantes: medicamentos para reducir el nivel de conciencia y facilitar la tolerancia a procedimientos invasivos. Se tomará como referencia la dosis utilizada en cada paciente. Los fármacos recogidos fueron:
    - Propofol
    - Midazolam
    - Dexmedetomidina
    - Sevoflurano
- Fentanilo: fármaco analgésico muy utilizado en la UCI para el tratamiento del dolor. En este caso, también registraremos la dosis administrada a cada paciente.
- Cisatracurio: se trata de un bloqueante neuromuscular, un fármaco capaz de producir la relajación completa de los músculos. Cabe destacar, que esta variable no recogerá la dosis administrada del mismo, sino el hecho de haberla administrado o no. de esta forma, a variable Cisatracurio tomara valores de 1 en el caso de que sea administrada y 0 en caso contrario.
- Escala de dolor ESCID: utilizada para valorar el dolor en pacientes inconscientes mediante la observación de parámetros conductuales y fisiológicos.
- APACHE II (Acute Physiology and Chronic Health Evaluation II): sistema de puntuación que evalúa la gravedad del paciente crítico basado en variables fisiológicas, edad y antecedentes, y permite estimar el riesgo de mortalidad.
- *SOFA* (Sequential Organ Failure Assessment): puntuación que evalúa el grado de disfunción orgánica mediante parámetros respiratorios, cardiovasculares, hepáticos, renales, neurológicos y hematológicos.

- *Glasgow*: valora el nivel de conciencia del paciente mediante la respuesta ocular, verbal y motora. Su puntuación total varía de 3 (coma profundo) a 15 (alerta).
- *RASS*: mide el grado de agitación o sedación del paciente, desde +4 (muy agitado) hasta -5 (sedación profunda).
- BIS: monitorización avanzada que analiza la actividad eléctrica cerebral mediante un EEG para determinar el nivel de sedación, con valores de 100 (despierto) a 0 (silencio eléctrico cerebral).

2.2.4 Variable objetivo

A la hora de demostrar la hipótesis planteada, fue necesario identificar y definir la variable objetivo del modelo. Para ello, decidimos representar el nivel de sedación de los pacientes mediante una variable categórica con tres clases:

- Paciente infrasedado: recibe dosis bajas de fármacos sedantes y presenta signos de agitación, inquietud y dolor.

- Paciente con sedación adecuada: presenta una respuesta controlada, se encuentra tranquilo favoreciendo la aplicación de diferentes tratamientos terapéuticos. No presenta dolor.

- Paciente sobresedado: recibe dosis muy elevadas de sedante y analgésicos, encontrándose en un estado de sueño profundo, sin respuesta frente a estímulos.

Dado el planteamiento del modelo de *ML* supervisado, fue necesario contar con la clasificación previa de los pacientes en las tres categorías. Para ello, el personal médico clasificó las 100 instancias, proporcionando las etiquetas verdaderas con las que trabajará el algoritmo. Con el fin de garantizar una mayor fiabilidad en el etiquetado, este proceso se realizó de forma doble y ciega: cada paciente fue evaluado de manera independiente por dos médicos distintos. Por lo que, finalmente, se obtuvieron dos etiquetas que podrían ser consideradas como '*gold standard*'.

2.3 Preprocesado de los datos

El preprocesado consiste en un conjunto de operaciones que se aplican a los datos antes de desarrollar el modelo. Este paso es fundamental para garantizar la calidad y confiabilidad de los datos utilizados en un estudio o proyecto. El objetivo principal de esta etapa es mejorar la calidad y fiabilidad de los datos, permitiendo un análisis más preciso y robusto.

El curado de datos se refiere al proceso de limpieza y transformación de los datos antes de su análisis. Esta técnica implica varias etapas como la eliminación de datos atípicos, la codificación adecuada de variables categóricas y la detección y manejo de valores faltantes.

- Identificación y corrección de errores

  En esta etapa se revisó de manera intensiva al conjunto de datos en su totalidad. El objetivo fue detectar cualquier tipo de error tipográfico y, en general, identificar cualquier inconsistencia en las variables.

- Codificación de variables categóricas

La codificación es el proceso de asignación de un determinado valor numérico a las variables cualitativas. Se trata de una etapa fundamental, pues las técnicas de *ML* solo trabajan con datos numéricos. En general, no existen reglas específicas para establecer una codificación, pero en la medida de lo posible se intenta tener una codificación simétrica.

Para las variable Sexo, se asignó valor 1 a la característica mujer, y valor 2 al hombre.

Para las variables relacionadas con factores de riesgo de cada paciente, se utilizó codificación binaria, asignado el valor de 1 a la presencia de dicho factor de riesgo y valor nulo en caso contrario. Se hizo una excepción con la variable de riesgo cardiovascular, donde decidimos codificar mediante enteros varias categorías. Se asigno valor 0 si el paciente no padece ningún tipo de riesgo cardiovascular, 1 si era fumador, 2 si consumía drogas, 3 si se cumplían los dos anteriores (fumador y consumidor de drogas), 4 si padecía algún problema de corazón y 5 si padecía problemas de corazón y era fumador.

Para las variables MI y VM mecánica, también optamos por una categorización con números enteros. En la Tabla 5 y Tabla 6 podemos ver los resultados de esta.

Para la variable objetivo, decidimos numerar con un 0 a pacientes sobresedados (habría que reducir la dosis de fármacos), 1 a pacientes sedados adecuadamente y 2 pacientes infrasedados (se debería aumentar la dosis de fármacos).

Tabla 5. *Codificación de la variable Motivo de ingreso*

MOTIVO DE INGRESO	CODIFICACIÓN
Shock séptico	1
Patología respiratoria	2
ACV	3
Status epiléptico	4
Patología cardiovascular	5
TCE	6
Neurocrítico	7

Tabla 6*. Codificación de la variable Ventilación mecánica*

VENTILACIÓN MECÁNICA	CODIFICACIÓN
Tubo endotraqueal en T	1
Tubo endotraqueal. Ventilación controlada	2
Tubo endotraqueal. Ventilación asistida/controlada	3
Traqueostomía. Ventilación asistida/controlada	4
CPAP	5
BIPAP	6

- Identificación y tratamiento de valores faltantes

La calidad de los datos es un aspecto fundamental en el *ML*, pues es de ellos de donde se extrae el conocimiento. La presencia de datos faltantes o *missing data* implica que ciertos registros presenten valores ausentes en variables específicas, lo que puede dificultar el análisis del conjunto de datos.

Para el tratamiento de los datos, se han realizado diversas técnicas de imputación. Se trata de métodos que sustituyen los valores perdidos con otros estimados, basándose en la información disponible. Cabe destacar, que no hemos requerido de la eliminación de instancias o de variables por la existencia de números datos faltantes, ya que consideramos suficiente que cada variable dispusiera de al menos el 80% de los datos.

Para las variables *RASS* y FR, puesto que el porcentaje de datos faltantes era del 2% y del 5% respectivamente, decidimos imputar los *missing values* utilizando la mediana. Se trata de un método explícito de imputación, en el cual se sustituyen los valores faltantes por la mediana de la variable correspondiente. La mediana nos ofrece un valor más próximo al valor real al no verse tan afectada por los valores atípicos o *outliers*, como puede pasar con la media.

Para las variables Glasgow, ESCID Y *SOFA*, con un 13% de datos faltantes, decidimos aplicar *Multivariate Imputation by Chained-Equations* (MICE). Se trata de una técnica más compleja de imputación multivariante. Mientras que la imputación univariante imputa datos solo de una columna específica, la imputación multivariante funciona simultáneamente con todas las variables en todas las columnas, ya sean faltantes u observadas.

MICE utiliza de manera iterativa modelos de regresión múltiple para estimar *missing values*. En cada iteración, cada variable del conjunto de datos se imputa utilizando el resto de las características.

La implementación del algoritmo no fue necesaria, ya que se disponía del software necesario gracias a trabajos previos del Grupo de Ingeniería Biomédica. Dichos algoritmos fueron aplicados a los datos de este estudio, obteniendo una base de datos sin valores faltantes.

2.4 Análisis estadístico de las variables

En este apartado se describirán diversas técnicas cuantitativas empleadas para explorar los datos recopilados. Primero se calcularon diversos estadísticos descriptivos, con el objetivo de obtener una visión general de las distribución de los datos y de cada variable. Seguidamente, se realizaron pruebas no paramétricas para estudiar las relaciones entre las variables independientes y la variable objetivo. Para ello, primero tuvimos que asegurarnos de que lo datos fueran paramétricos o no paramétricos, probando la normalidad y homocedasticidad de nuestros datos. A continuación, se evaluó la existencia de diferencias estadísticamente significativas en las variables analizadas entre los grupos de sedación. Finamente, calculamos la correlación entre todas las variables bajo estudio y la variable de clasificación del nivel de sedación. Cabe recalcar, que estos cálculos se hicieron solo con una de las variables de niveles de sedación proporcionada por los médicos.

2.4.1 Estadística descriptiva

Para extraer información sobre la tendencia de cada variable y su distribución decidimos calcular la mediana y el rango intercuartílico de las variables continuas u ordinales. Estos resultados se ven reflejados en la Tabla 7. La mediana nos permite calcular la tendencia central de la variable. Representa el valor que divide a la muestra en dos mitades. Por otra parte, el rango intercuartílico es una medida de la dispersión de la variable. Se calcula como la diferencia entre el tercer cuartil y el primero (Rendón-Macías et al., 2016).

En el caso de las variables categóricas se realizó el cálculo de la frecuencia relativa. Como vemos en la Tabla 8, se ha calculado para de cada una de las categorías de cada variable en el total de los datos y en cada uno de los grupos en función del nivel de sedación.

Tabla 7*. Mediana y rango intercuartílico de las variables continuas y ordinales. Los datos se expresan como mediana ± rango intercuartílico (RIQ).*

VARIABLES	TODOS	INFRASEDADOS	SOBRESEDADOS	SEDADOS CORRECTAMENTE
Edad	68 ± 21	63 ± 27	67 ± 20	65 ± 21
Peso	74 ± 26.5	69.7 ± 34.1	74 ± 29.25	75 ± 26.75
Días de ingreso	7 ± 8	4.5 ± 6	8.5 ± 11.5	4 ± 6.5
RASS	-3 ± 3	-4 ± 2	-3 ± 2	-4 ± 2
BIS	53 ± 25.5	53 ± 22	57 ± 33	45.5 ± 12
Glasgow	3 ± 3	3 ± 3	3 ± 3	3 ± 3
ESCID	0 ± 1	0 ± 1	0 ± 1	0 ± 0
APACHE	20 ± 12	19.5 ± 10	19 ± 12	20 ± 10
SOFA	6 ± 4	5 ± 4.5	7 ± 4	5 ± 3
TAM	83.5 ± 21	92 ± 19	81 ± 17.5	89.5 ± 36.5
FC	83 ± 35	84 ± 31	84 ± 38.5	83.5 ± 26
FR	19 ± 8	19 ± 8.5	19 ± 8	19 ± 7.5
Aminas	0 ± 0.05	0 ± 0.15	0 ± 0	0.015 ± 0.225
Propofol	1.08 ± 2	1.15 ± 2.36	1.065 ± 1.88	1.235 ± 2.21
Dexmedetomidina	0 ± 0	0 ± 0	0 ± 0	0 ± 0
Sevoflurano	0 ± 0	0 ± 0	0 ± 0	0 ± 0
Midazolam	0 ± 3.5	0 ± 10	0 ± 1	0 ± 9
Fentanilo	0.4 ± 0.2	0.4 ± 0.1	0.31 ± 0.25	0.4 ± 0.3

***Tabla 8**. Frecuencia relativa de cada categoría de las variable en función del nivel de sedación y respecto el total.*

VARIABLES	CATEGORIAS	TOTAL	INFRASEDACIÓN	SOBRESEDACIÓN	SEDACIÓN CORRECTA
Sexo	Mujer = 1	0.39	0.45833	0.39062	0. 33333
	Hombre = 2	0.61	0.54167	0.60938	0. 66667
Dislipemia	No	0.8	0.70833	0.82812	0. 79167
	Sí	0.2	0.29167	0.17188	0. 20833
EPOC	No	0.78	0.75	0.79688	0. 70833
	Sí	0.22	0.25	0.20312	0. 29167
Cáncer	No	0.78	0.83333	0.76562	0. 79167
	Sí	0.22	0.16667	0.23438	0. 20833
Factores de riesgo cardiovasculares	No	0.44	0.70833	0.3125	0.625
	Fumador	0.14	0.041667	0.1875	0.083333
	Bebedor	0.05	0.041667	0.046875	0.083333
	Fumador y alcohólico	0.04	0	0.0625	0
	Patologías cardiacas	0.17	0.125	0.1875	0.083333
	Patologías cardiacas y fumador	0.16	0.083333	0.20312	0.125
Patología renal	No	0.89	0.875	0.90625	0.083333
	Sí	0.11	0.125	0.09375	0.16667
Diabetes Mellitus	No	0.72	0.70833	0.71875	0.79167
	Sí	0.28	0.29167	0.28125	0.20833
HTA	No	0.57	0.625	0.5625	0.58333
	Sí	0.43	0.375	0.4375	0.41667
Motivo de ingreso	shock séptico	0.11	0	0.17188	0
	patología respiratoria	0.23	0.375	0.15625	0.33333
	ACV	0.24	0.375	0.17188	0.29167
	status epiléptico	0.04	0	0.0625	0
	patología cardiovascular	0.07	0.083333	0.0625	0.125
	TCE	0.26	0.125	0.32812	0.16667
	neurocrítico	0.05	0.041667	0.046875	0.083333
VM	tubo endotraqueal en T	0.06	0	0.09375	0
	tubo endotraqueal. Ventilación controlada	0.45	0.66667	0.29688	0.70833
	tubo endotraqueal. Ventilación asistida/controlada	0.29	0.29167	0.3125	0.25
	traqueostomía. Ventilación asistida/controlada	0.02	0	0.03125	0
	CPAP	0.12	0.041667	0.17188	0.041667
	BIPAP	0.06	0	0.09375	0
Cisatracurio	No	0.87	0.79167	0.92188	0.29167
	Sí	0.13	0.20833	0.078125	0.70833

2.4.2 Test de hipótesis

Para poder aplicar pruebas paramétricas, es imprescindible comprobar el cumplimiento de los siguientes supuestos: la normalidad de las variables (es decir, que sigan una distribución normal) y la homocedasticidad (homogeneidad de las varianzas entre grupos).

Para comprobar la primera condición, se utilizó el test de *Lilliefors*. La hipótesis nula establece que los errores se distribuyen normalmente. La hipótesis alternativa plantea que no lo hacen. Para comprobar la homocedasticidad se empleó el test de Levene. En este caso, la hipótesis nula establece que las varianzas de los grupos comparados son iguales, mientras que la hipótesis alternativa plantea que al menos una de ellas es diferente.

Debido a que no todas las variables eran normales ni homocedásticas, se procedió a utilizar pruebas no paramétricas. Considerando que las variables independientes y la variable dependiente (nivel de sedación) son mutuamente independientes, se realizaron los análisis correspondientes.

Para evaluar si existen diferencias estadísticamente significativa entre los tres niveles de sedación de la variable objetivo en cada una de las variables dependientes, se realizaron dos pruebas distintas en base al tipo de variable dependiente. En el caso de las variables ordinales o continuas, se realizó la prueba de *Krustal-Wallis*. Para las variables categóricas calculamos la prueba Chi-cuadrado.

Una de las relaciones más interesantes a analizar consiste en evaluar si existen diferencias significativas entre los distintos niveles de sedación respecto a cada una de la variables independientes. Para ello, se aplicó la prueba de *Mann–Whitney U,* para comparar cada variable continua o cualitativa ordinal entre pares de grupos; y el test de Chi-cuadrado, para las variables nominales.

Todos los resultados anteriores se pueden observar en la Tabla 9.

Tabla 9. *P*-valores globales y entre pares de grupos

	p-valor			
VARIABLES	**I vs S vs SA**	**I vs SA**	**I vs S**	**SA vs S**
Edad	0.6237	0.6758	0.3954	0.4245
Peso	0.1340	0.0829	0.1843	1
Días de ingreso	**0.0125**	**0.0117**	**0.0434**	0.9058
RASS	**0.0046**	**0.0302**	**0.0036**	0.5380
BIS	**0.0178**	0.1801	0.1308	**0**
Glasgow	0.5687	0.3398	0.8606	0.3569
ESCID	0.3147	0.2215	0.5776	0.1408
APACHE	0.4683	0.4810	0.2214	0.9865
SOFA	0.2992	0.1530	0.4675	1
TAM	0.4501	0.6835	0.1646	0.7371
FC	0.5661	0.4676	0.3731	0.7119
FR	0.9276	0.9028	0.7479	0.6860
Aminas	**0.0081**	**0.0027**	0.1199	0.4033
Propofol	0.9456	0.8947	0.8296	0.6926
Dexmedetomidina	0.2312	0.1647	0.3281	1
Sevoflurano	0.5501	0.3239	0.4347	1
Midazolam	0.5620	0.3181	0.5495	0.8886
Fentanilo	0.192	0.4254	0.0731	0.3823
Sexo	0.6268	0.6210	0.4793	0.3336
Dislipemia	0.4360	0.6929	0.1972	0.4142
EPOC	0.5995	0.3774	0.7709	0.4142
Cáncer	0.8629	0.7952	0.6055	0.7659
Cardiovascular	0.0552	0.1028	**0.0463**	0.2590
Patología renal	0.5924	0.3371	0.9088	0.4955
DM	0.4224	0.4881	0.3487	0.1883
HTA	0.9798	0.8605	0.8937	1
MI	**0.0189**	0.0538	**0.0240**	0.4039
VM	**0.0174**	**0.0109**	0.0635	0.7713
Cisatracurio	**0.0260**	**0.0093**	0.9510	0.1564

2.4.3 Análisis de asociación

El análisis de asociación nos permite obtener información sobre posibles correlaciones entre las variables bajo estudio y los niveles de sedación. Para ello hemos calculado el coeficiente de correlación de todas las variables respecto nuestra variable objetivo.

Los coeficientes de correlación permiten evaluar la fuerza y dirección de las relaciones lineales entre pares de variables. Estos no aportan información sobre si una variable se mueve en respuesta a otra, sino indican asociaciones entre variables (Mukaka, 2012).

Cuando ambas variables se distribuyen normalmente y se desean analizar interacciones lineales, se utiliza el coeficiente de correlación de Pearson. Puesto que nuestras variables no siguen una distribución normal y no tenemos una hipótesis previa de qué tipo de relación mantendrán las variables, utilizamos el coeficiente de correlación de Spearman. Además, este último es más robusto frente *outliers (Mukaka, 2012).*

Para una correlación entre las variables x e y, la fórmula para calcular el coeficiente de correlación de Spearman muestral viene dada por:

$$r_s = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)} \quad (1)$$

Donde d es la diferencia entre los rangos para las dos variables.

2.5 Selección de características

La extracción de características se refiere al proceso de identificación y selección de las variables más relevantes de nuestro conjunto de datos. Este proceso es crucial, ya que una selección adecuada de características puede mejorar la precisión y eficiencia de los modelos de aprendizaje automático.

Puesto que hemos optado por un aprendizaje supervisado, antes de realizar la selección de características, se dividió la muestra en un conjunto de entrenamiento (*trainning*) y otro de prueba (*test*). Con el fin de garantizar que ambos subconjuntos fueran representativos de la población total, se aplicó un procedimiento de emparejamiento iterativo que permitió obtener particiones balanceadas.

En cada iteración, se generó una partición aleatoria de nuestros datos en dos subconjuntos de 50 sujetos cada uno. Después, se evaluó la similitud entre los grupos generados en base a la distribución de las variables sociodemográficas más representativas del estudio. Estas variables eran: el sexo del paciente, la edad y la valoración de la sedación (variable objetivo). Para las variables categóricas, como el sexo y la valoración de la sedación, se aplicó la prueba de independencia chi-cuadrado, mientras que para la variable edad se utilizó la prueba no paramétrica de Kruskal-Wallis.

Finalmente, seleccionamos los conjunto de entrenamiento y test donde las diferencias entre ambos grupos son menos significativas, es decir, donde los p-valores sean más próximos a la unidad.

Una vez creados los grupos, se continuó con la selección de características sobre. Para ello, aplicamos *Fast Correlation-Based Filter* (*FCBF*) sobre el conjunto de entrenamiento. Se trata de un método donde, dado un conjunto de características y una clase, el algoritmo encuentra un conjunto de características predominantes relevantes y de baja redundancia para dicha clase.

Para ello, primero calcula cuanto se relaciona cada variable con la clase a través de la incertidumbre simétrica o *Symmetrical Uncertainty* (*SU*). Una vez se dispone de estos valores, se seleccionan aquellas características cuyo *SU* supere cierto umbral predefinido. Luego, se ordenan de mayor a menor relevancia. La segunda parte del algoritmo procesa la lista de características ordenadas anterior para eliminar aquellas que considera redundantes (Yu & Liu, 2003).

2.6 Machine Learning

Una vez seleccionadas las características, obtenemos dos subconjuntos reducidos del espacio de variables original. Utilizando el conjunto de datos de entrenamiento previamente etiquetado, se construye un modelo que sea capaz de predecir la clasificación de nuevos datos en los grupos de nivel de sedación.

Los algoritmos de clasificación pueden ser: supervisados, donde requieren etiquetas de clase conocidas para entrenar el modelo; o no supervisados, donde el modelo agrupa a los sujetos sin etiquetas. Como se indicó, para este estudio optaremos por utilizar un aprendizaje supervisado.

En este trabajo, se plantea el uso de tres tipos de algoritmos de *ML* supervisados: RL, *LDA* y *AdaBoost*. Los dos primeros (RL y LDA) fueron considerados por ser métodos simples y por tanto, de alta explicabilidad. Además, dado que estos datos no habían sido antes explorado y dada la novedad del estudio no exiten estudios previos de referencia, se consideró partir, en primer lugar, de métodos simples. Una vez evaluados estos modelos, se incluyó un método que se suele considerar más potente (*AdaBoost*) para analizar si los resultados de la clasificación podían ser mejorados respecto a los métodos anteriores.

2.6.1 Regresión Logística

La RL es un modelo de *ML* supervisado utilizado cuando la variable dependiente es categórica. Este algoritmo puede servir tanto para clasificar una instancia en dos clases (RL binaria), como en muchas clases (RL multinomial). Dado que el enfoque del TFG es predecir el nivel de sedación entre 3 posibles categorías, a continuación, describiremos la técnica de RL multinomial.

La RL multinomial, también conocida como *softmax regression*, pretende etiquetar cada observación con una clase k dentro de un conjunto de *K* clases. Para ello, el clasificador utiliza una generación de la función sigmoide, la función *softmax*, que devuelve un vector de *K* probabilidades como vemos en la Ecuación (2). Cada valor de este vector representa la probabilidad de que una instancia pertenezca a cada una de las clases posibles, en función de las variables independientes (Daniel Jurafsky & James H. Martin., 2025).

$$P(y = k \mid x) = \frac{e^{\beta_k x}}{\sum_{j=1}^{K} e^{\beta_j x}} \quad (2)$$

Una vez tenemos la probabilidad de ocurrencia del evento, el algoritmo clasifica en base a un umbral.

2.6.2 Análisis del discriminante lineal

El algoritmo de *LDA* busca proyectar la matriz de datos originales en un espacio de menor dimensionalidad, con el objetivo de encontrar combinaciones lineales de características que maximicen la separabilidad de clases mientras se minimiza la varianza intraclase (Tharwat et al., 2017).

Dada la matriz de datos original $X = \{ x_1, x_2, \ldots, x_{N,} \}$, donde x_i representa la i-ésima muestra, y N es el número total de muestras. Cada muestra está representada por M características, por lo que podemos decir que cada muestra está representada como un punto en el espacio de M dimensiones (Tharwat et al., 2017).

Para disminuir M, primero se calcula la varianza entre clases (S_B), que es la distancia entre las medias de las clases, como vemos en la ecuación (3) (Tharwat et al., 2017):

$$(m_i - m)^2 = (W^T \mu_i - W^T \mu)^2 = W^T (\mu_i - \mu)(\mu_i - \mu)^T W \quad (3)$$

Donde m_i representa la proyección de la media de la i-ésima clase, m es la proyección de la media total de todas las clases, W representa la matriz de transformación de *LDA*, μ_i es la media de la i-ésima clase y μ la media total de todas las clases (Tharwat et al., 2017).

Por tanto, la varianza total entre las clases se calcula como (Tharwat et al., 2017):

$$S_B = \sum_{i=1}^{c} n_i S_{Bi} \quad (4)$$

Después, el algoritmo calcula la varianza intraclases (S_W), es decir, la distancia entre la media de una clase y las muestras dentro de dicha clase. Mide como de dispersas están las muestras respecto su punto de masa (la media μ_i), como vemos en la Ecuación (5). En este caso, x_k es una muestra de una clase determinada, y μ_i es la media de dicha clase (Tharwat et al., 2017).

$$S_{W_i} = \sum_{x_k \in \omega_i} (x_k - \mu_i)(\mu_k - \mu_i)^T \quad (5)$$

Por ello, la matriz total de la varianza intraclases (S_w) corresponde al sumatorio de todas las varianzas intraclases (Tharwat et al., 2017).

Por último, se proyectan los datos en un espacio reducido usando la matriz de transformación W. El objetivo es encontrar una matriz donde se maximice la distancia entre clases y se minimice la distancia intraclases (Tharwat et al., 2017).

$$\max J(W) = \frac{W^T S_B W}{W^T S_W W} \quad (6)$$

2.6.3 AdaBoost

El *boosting* es una herramienta eficaz para mejorar la capacidad de predicción de los sistemas de aprendizaje. Esta técnica plantea dos problemas principales: cómo ajustar el conjunto de entrenamiento para que el clasificador débil pueda realizar el entrenamiento y cómo combinar los clasificadores débiles en uno fuerte. Para resolver los problemas anteriores, se planteó el algoritmo *AdaBoost* (*adaptive boosting*), que ajusta el peso de cada característica sin necesidad de ningún conocimiento previo sobre el aprendizaje (Chengsheng et al., 2017).

Existe diversidad de algoritmos *AdaBoost* (*AdaBoost.M1, AdaBoost.M2, AdaBoost.R,* etc.), pero la aplicación de todos ellos se centra en problemas de clasificación (Chengsheng et al., 2017). Para el estudio planteado, se seleccionó el *AdaBoostM2* aplicado a la clasificación multiclase y etiqueta única.

AdaBoost.M2 no solo entrena clasificadores débiles, sino que diseña un esquema para penalizar los errores de clasificación multiclase usando instancias. El algoritmo *AdaBoost.M2* se ejecuta a lo largo de *T* iteraciones. En cada ronda, se llama a una distribución de probabilidad D_t. Esta se va actualizando, dando mayor peso a las clasificaciones erróneas. A partir de esta distribución ya ponderada, el algoritmo entrena un clasificador débil h_t, el cual tratará de minimizar el error respecto los ejemplos más relevantes (de mayor peso) según D_t.

Una vez entrenado el clasificador, se calcula un peso β_t, un parámetro que nos ayuda a cuantificar el rendimiento del clasificador. Para obtener la hipótesis final *H*, se combinan todas las hipótesis débiles h_t a través de los pesos β_t mediante votación ponderada (Lazarevic & Obradovic, 2002).

2.7 Evaluación del rendimiento de los modelos

Una vez definidos los modelos, se procedió a su entrenamiento y evaluación. En total, se entrenaron 6 modelos distintos: dos modelos de *LDA*, dos modelos de RL y dos modelos *AdaBoost.M2*. En cada caso, se entrenó un modelo por cada una de las dos etiquetas de clasificación proporcionada por los médicos.

Para evaluar la eficacia de cada modelo, se utilizaros las siguientes métricas:

- Sensibilidad (en inglés *Sensitivity*, *Se*): también conocida como *Recall* o tasa de verdaderos positivos (TPR), representa la probabilidad de clasificación correcta de los verdaderos positivos (VP) entre todos los casos reales de una clase. Llamamos VP a aquellos sujetos que, según la etiqueta verdadera pertenecen a una clase, y el algoritmo les clasifica de manera correcta en la misma clase.

  Aplicado a nuestro estudio, la Se podría explicarse como la capacidad del modelo para detectar correctamente a los pacientes que pertenecen a un determinado nivel de sedación (Hernández et al., 2022):

$$Se = \frac{VP}{VP + falsos\ negativos\ (FN)} \tag{7}$$

- Especificidad (en inglés *Specificity*, Sp): representa la probabilidad de clasificar correctamente a las instancias que realmente no pertenecen a una clase. Es decir, es la probabilidad de clasificar correctamente a los verdaderos negativos (VN) (Vizcaíno-Salazar, 2017).

$$Sp = \frac{VN}{VN + falsos\ positivos\ (FP)} \tag{8}$$

- Exactitud (en inglés *Accuracy*, *Acc*): es la tasa de acierto general, es decir, la proporción total de predicciones correctas sobre el total de casos evaluados. Nos da una idea sobre lo cercano que es el resultado al valor real. Es la probabilidad de que un paciente se clasifique correctamente según la etiqueta proporcionada por los médicos (Hernández et al., 2022).

  La *Acc* se calcula como la suma de las predicciones correctas, es decir, los VP y VN, dividida entre el número total de predicciones realizadas, como se muestra en la ecuación (10).

$$Acc = \frac{VP + VN}{VP + VN + FP + FN} \tag{9}$$

  Para asegurarnos de un correcto uso de la selección de características, realizamos una comparación entre la *accuracy* del modelo con las características seleccionadas con *FCBF* y la *accuracy* con todas las características.

- Valor Predictivo Positivo (PPV por sus siglas en inglés): representa la proporción de instancias clasificadas en una clase y que realmente pertenecen a ella. Este valor también es conocido con el nombre de precisión. Se trata de una prueba que representa la seguridad de la clasificación automática (Hernández et al., 2022).

$$PPV = \frac{VP}{VP + FP} \tag{10}$$

- Valor Predictivo Negativo (NPV por sus siglas en inglés): representa la proporción de instancias no clasificadas en una clase y que realmente no pertenecen a ella. Al igual que el PPV, representa la seguridad con la que se realizan de las predicciones (Vizcaíno-Salazar, 2017).

$$NPV = \frac{VN}{VN + FN} \tag{11}$$

- Razón de verisimilitud o *Likelihood Ratio (*LR): relación entre la probabilidad de que una observación pertenezca a una clase dada según el modelo, y la probabilidad de que la misma observación pertenezca a esa clase por azar o bajo un modelo nulo. Existen dos tipos de parámetros utilizados para medir dicha relación (Vizcaíno-Salazar, 2017):

- Razón de verosimilitud positiva (LR+): mide cuanto más probable es que una instancia de una determinada clase se clasifique como ello, en comparación con que no lo sea. Se calcula como el cociente entre la Se y el complemento de la Sp. Según los valores de LR+ podemos evaluar la calidad del clasificador: para resultados mayores de 10, el algoritmo sería un gran clasificador; valores entre 5 y 10 corresponden a una buena prueba; valores entre 5 y 2 la calidad de predicción es regular y para valores inferiores a 2 el algoritmo resulta inútil (Aznar-Oroval et al., 2013).

$$LR+ = \frac{VP}{FP} = \frac{Se}{1 - Sp} \tag{12}$$

- Razón de verosimilitud negativa (LR-): mide cuanto de probable es que una instancia que se ha clasificado como no perteneciente a una determinada clase, realmente no pertenezca a dicha clase. En este caso, el rango de valores es diferente para calcula la calidad del modelo. Un LR- entre 0.5 y 1 representa un modelo inútil, valores entre 0.2 y 0.5 representan un modelo regular, valores entre 0.1 y 0.2 representan un algoritmo de buena calidad y menor de 0.1 modelo de clasificación excelente.

$$LR- = \frac{FN}{VN} = \frac{1 - Se}{Sp} \tag{13}$$

- *F1-score*: métrica utilizada para evaluar la capacidad de clasificación de una prueba. Es una medida que muestra la relación entre la Se y el PPV. Los valores del *F1-score* varían entre 0 y 1, de tal forma que un clasificador perfecto tendría un *F1-score* próximo a la unidad; mientras que un clasificador inútil tendría valores nulos (Molina, 2024).

$$F1 = 2 \cdot \frac{Se \cdot PPV}{Se + PPV} \tag{14}$$

- *AUC*: corresponde al parámetro que mide el área bajo la curva ROC. *El AUC e*s un parámetro utiliza para medir la capacidad discriminativa de un test, es decir, que tan bueno es nuestro modelo para clasificar de manera correcta las instancias. Un valor próximo a 1, implica una mayor capacidad discriminativa (Cerda & Cifuentes, 2012).

- Coeficiente *Kappa* de Cohen (κ): medida de concordancia. Evalúa la relación entre las predicciones del modelo y las etiquetas reales. Es un tipo de coeficiente de correlación, por lo que varía de -1 a +1. El valor máximo representa una concordancia perfecta, es decir, la predicciones concuerdan con las etiquetas reales. Un valor nulo, representa el azar y un valor negativo indica que no hay ningún acuerdo entre las predicciones y las etiquetas. Podemos calcular este coeficiente como (Manterola et al., 2018):

$$\kappa = \frac{Pr(a) - Pr(e)}{1 - Pr(e)} \tag{15}$$

donde *Pr(a)* representa la proporción de acuerdo real entre los observadores, mientras que *Pr(e)* corresponde al nivel de concordancia que se esperaría obtener de forma aleatoria

Cabe señalar, que estas métricas no solo sirvieron para la evaluación de los modelos, sino que también fueron utilizadas para comparar de forma directa las dos series de etiquetas de referencia proporcionadas por los médicos. De este modo, pudimos cuantificar el grado de concordancia de las decisiones clínicas.

2.8 Inteligencia Artificial Explicable

La detección de patrones y relaciones entre distintas variables es una de las ventajas que ofrece el uso de la IA. Para aprovechar todo su potencial es necesario comprender qué hay detrás de cada predicción. El continuo incremento de volúmenes de datos hace necesario el uso de algoritmos más sofisticados, dificultándose aún más la tarea de la interpretabilidad. Es aquí donde entran en juego las técnicas de *XAI*.

En este TFG, se propone emplear el uso de la técnica de *SHAP* para identificar qué características son decisivas a la hora de realizar la clasificación para el algoritmo de mayor rendimiento. Esta técnica asigna a cada variable un valor *Shapley* que cuantifica su aportación a la predicción (Lundberg & Lee, 2017).

Para calcular todos los valores *Shapley* recurrimos a kernel *SHAP*, un método que transforma la estimación de importancia en el ajuste de una regresión lineal ponderada. Para construir dicho modelo, primero se generan múltiples "coaliciones", es decir subconjuntos de características (Lundberg & Lee, 2017).

Para u sujeto x de una muestra con M características, generamos las coaliciones $\mathrm{z}' \in \{0,1\}^{\mathrm{M}}$. Este vector z_j' toma valor 1 si la variable j esta activa y 0 si está ausente. Dado que los modelos de *ML* no aceptan directamente entradas con variables ausentes, las sustituimos por valores extraídos del conjunto de entrenamiento (Lundberg & Lee, 2017).

Una vez realizadas las coaliciones, calculamos la predicción $f(h(z'))$ para cada una de ellas y asignamos un peso según la función kernel de *SHAP*, representada en la ecuación 16 (Lundberg & Lee, 2017).

$$\pi(z') = \frac{M-1}{(M\ chooce\ |z'|)|z'|(M-|z'|)} \tag{16}$$

Finalmente, ajustamos el modelo de regresión lineal $f\big(h(z')\big)$ usando los pesos calculados anteriormente. Los coeficientes ϕ_j son los valores de *Shapley* de cada característica (Lundberg & Lee, 2017).

Capítulo 3. Resultados

3.1 Introducción

A lo largo de este apartado comenzaremos a exponer los resultados obtenidos a lo largo del estudio. Primero, mostraremos las gráficas obtenidas a partir de algunos de los análisis de asociación. Seguidamente, se visualizarán las gráficas obtenidas tras la creación de los grupos de entrenamiento y test para asegurar una correcta distribución de las muestras. Continuaremos con los resultados obtenidos tras la selección de características y mostraremos los resultados del rendimiento de todos los modelos. Finalmente, se mostrarán los resultados tras aplicar técnicas de *XAI* al mejor de los modelos.

3.2 Resultados del análisis descriptivo

Una vez calculados los *p*-valores correspondientes para cada una de las variables analizadas, se representaron gráficamente con el objetivo de identificar aquellas que presentan significación estadística. En la Figura 5, vemos la gráfica de barras donde cada barra representa el *p*-valor asociado a una variable en relación con la variable dependiente del estudio. Las barras están coloreadas en azul, mientras que aquellas variables con *p*-valores inferiores al umbral convencional de significación estadística ($p < 0,05$) se muestran en rojo. Este umbral está señalado también mediante una línea discontinua horizontal roja para facilitar su visualización. De esta forma, es posible observar de manera clara cuáles de las variables presentan una asociación estadísticamente significativa.

Las variables estadísticamente significativas, coloreadas en rojo, son: días de estancia, BIS, *RASS*, dosis de aminas, MI, VM y administración de cisatracurio.

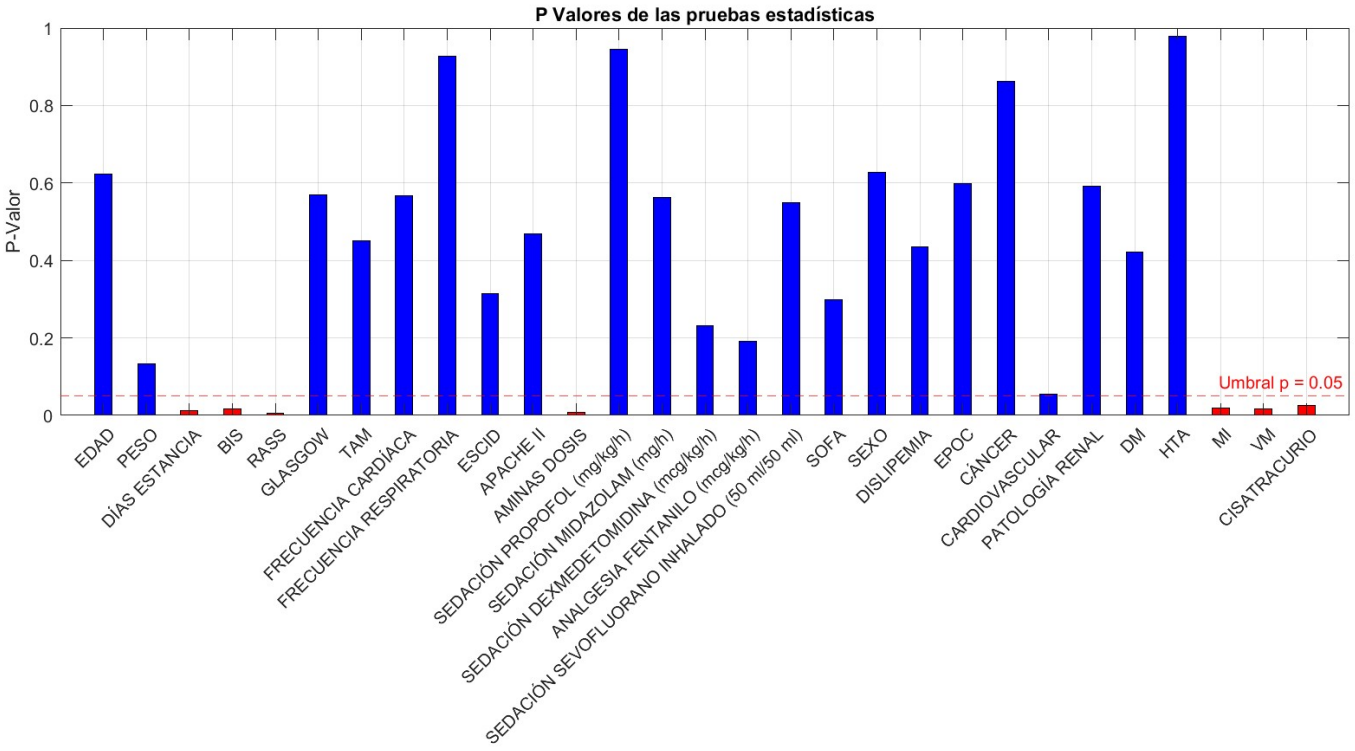

Figura 5. *Gráfica de barra del p-valor asociado a cada variable del estudio respecto la variable dependiente nivel de sedación.*

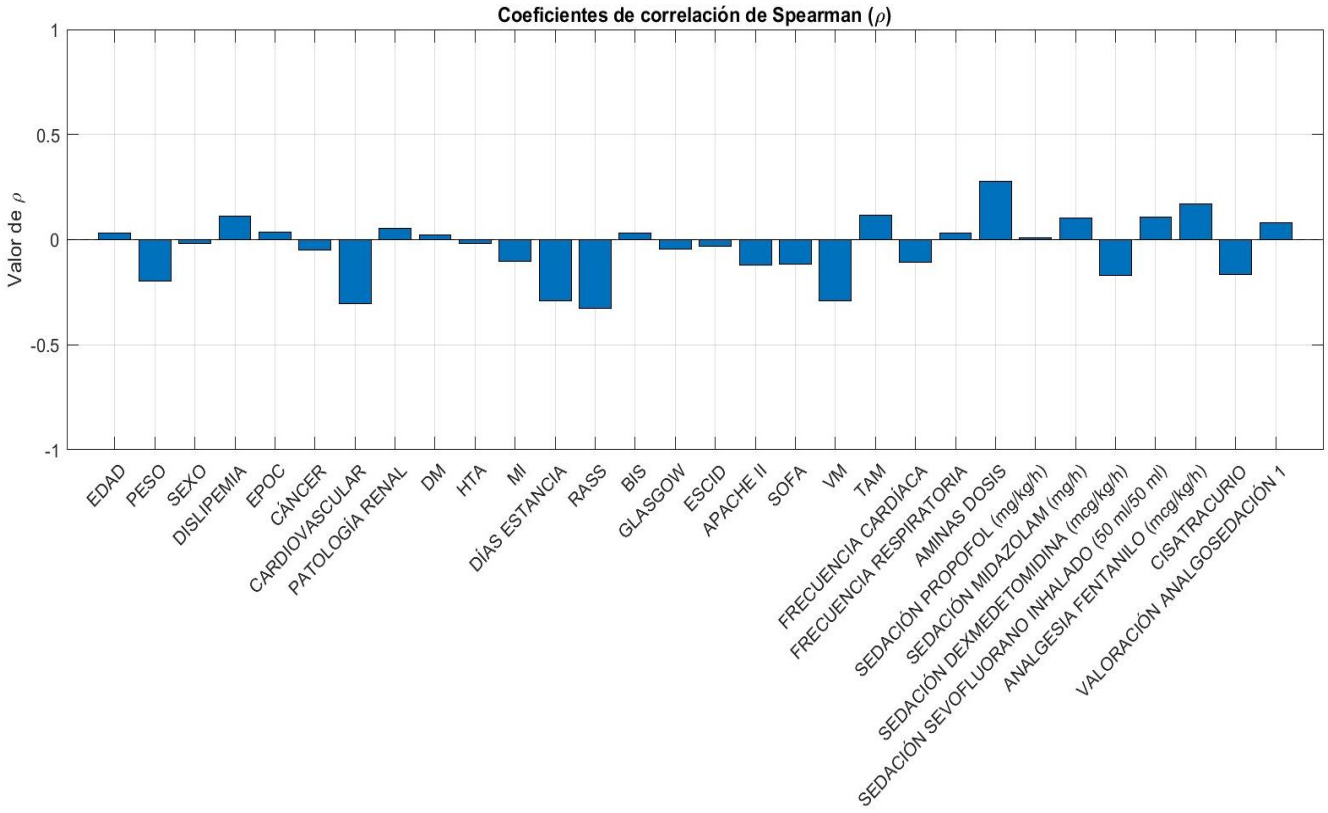

Figura 6. *Gráfica de barras de los coeficientes de correlación de Spearman de cada variable bajo estudio respecto la variable nivel de sedación.*

3.3 Resultados del análisis de asociación

Tras calcular los coeficientes de correlación de Spearman (ρ) entre la variable objetivo y el resto de las variables recogidas en el estudio, se graficaron os resultados. En la Figura 6 se presenta una gráfica de barras en la que se muestran dichos valores. De esta forma, podemos analizar la dirección y fuerza de las posibles asociaciones facilitando el entendimiento.

Los valores positivos indican una correlación directa, mientras que los negativos reflejan una correlación inversa. La magnitud del coeficiente permite apreciar la intensidad de dicha relación, sin implicar causalidad ni significación estadística por sí misma.

En la gráfica observamos que variables como *SOFA*, *RASS*, la presencia de cardiopatías y la frecuencia respiratoria presentan una correlación negativa. Por otro lado, la dosis de aminas, la TAM, la sedación con midazolam o la analgesia con fentanilo indican correlaciones positivas destacadas.

3.4 Resultados selección de características

Como bien explicamos en el capítulo 2, antes de realizar la selección de características, fue necesario crear los grupos balanceados de entrenamiento y test. Cada grupo fue pareado en edad, sexo y nivel de sedación.

En las Figura 7 podemos ver las gráficas utilizadas para visualizar dichos resultados. Para representar la variable de niveles de sedación y la distribución del sexo, al tratarse de variables categóricas, optamos por utilizar diagramas de barras (Figura 7.a y Figura 7.b). Las barras rojas representan el conjunto de prueba y las azules el conjunto de entrenamiento. En la parte inferior, Figura 7.c, encontramos un diagrama de caja donde se muestra la distribución de la edad en ambos subconjuntos. La línea roja representa la mediana, la caja el rango intercuartílico y las cruces los *outliers*.

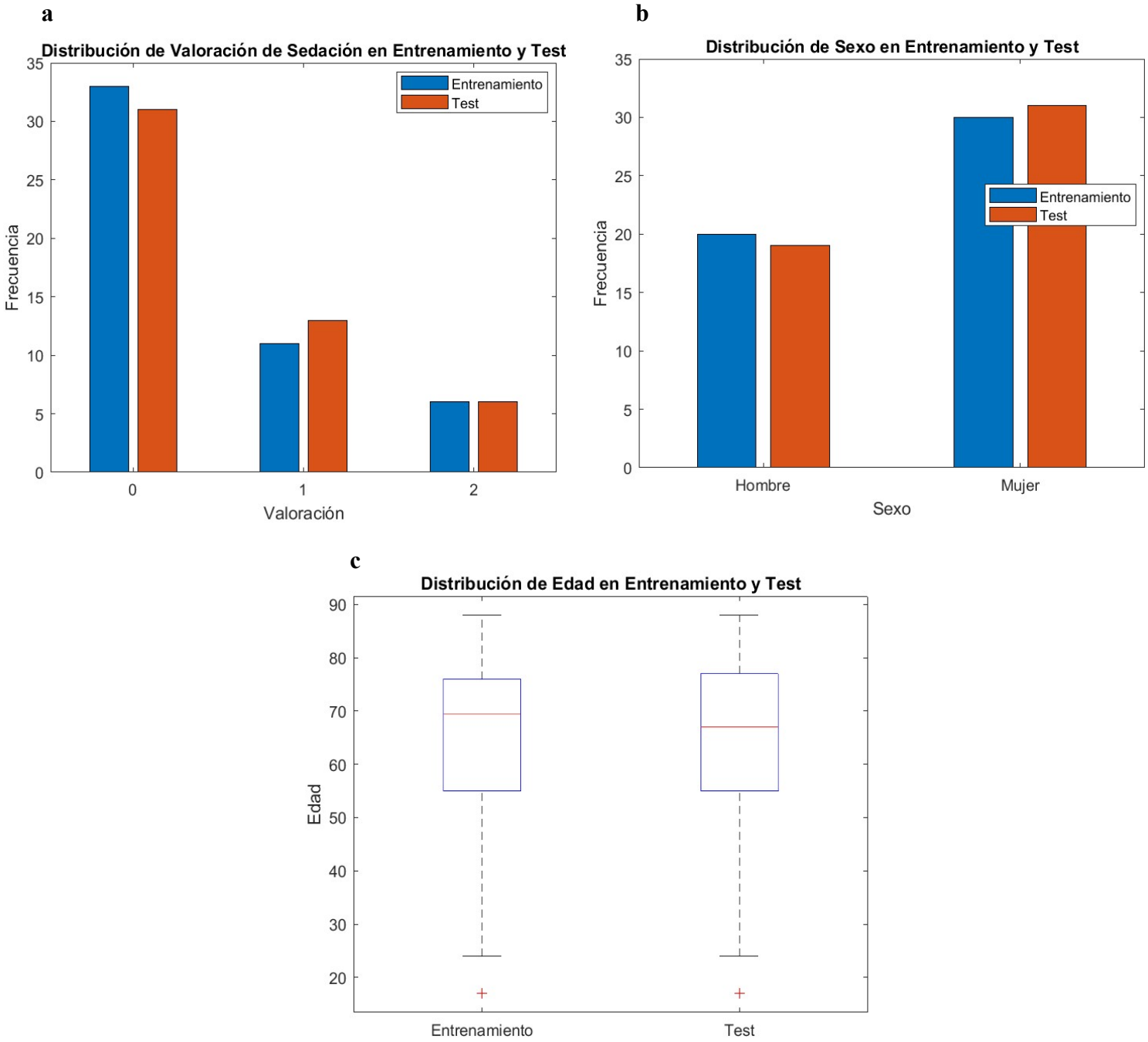

Figura 7 | Graficas para asegurar el balanceo entre el grupo de entrenamiento y test de diferentes variables.
a. *Gráfico de barras de la distribución de cada grupo de la variable nivel de sedación en los conjuntos de entrenamiento y test.* ***b.*** *Gráfico de barras de la distribución de cada grupo de la variable sexo en los conjuntos de entrenamiento y test.* ***c.*** *Diagrama de cajas de la distribución de la variable edad en los subconjuntos deentrenamiento y test.*

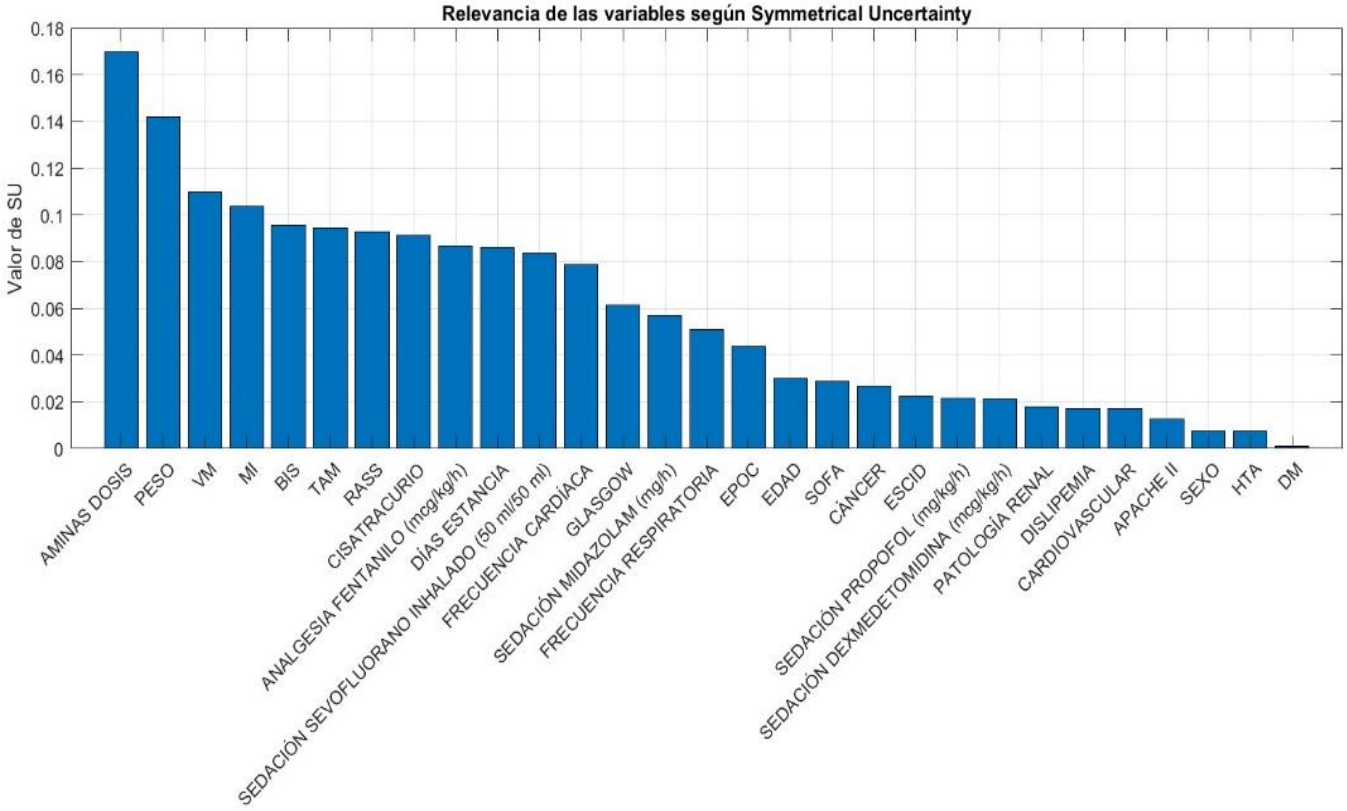

***Figura 8**. Gráfico de barras de las variables del estudio ordenadas en función del parámetro Symmetrical Uncertainty.*

Al aplicar el método *FCBF* al conjunto de entrenamiento, se obtuvo una lista ordenada de características en función de su relevancia, evaluada mediante el parámetro SU. En la Figura 8 se muestra el resultado en forma de gráfico de barras, donde las variables del estudio se encuentran ordenadas de mayor a menor relevancia según su valor de *SU*, representado por la altura de cada barra.

Como resultado del proceso de selección, el algoritmo identificó un total de seis características relevantes: dosis de aminas, peso del paciente, tipo de VM, MI, TAM y sedación con sevoflurano inhalado.

3.5 Resultados de los modelos de Machine Learning

Una vez seleccionadas las variables más significativas, se procedió al entrenamiento de los modelos de *ML*. La idea principal consistió en la creación de dos modelos de RL, cada uno entrenado con una de las dos variables objetivo; así como dos modelos de *LDA*, también entrenados individualmente con cada objetivo. Finalmente, con el fin de evaluar si era posible mejorar el rendimiento predictivo, se aplicó el algoritmo *Adaboost.M2*.

Antes del entrenamiento y validación de dichos algoritmos, se calcularon diversos estadísticos entre las dos series de etiquetas de referencia proporcionadas por los médicos. De este modo, pudimos cuantificar el grado de concordancia de las decisiones clínicas, sirviendo como referencia para la posterior interpretación de los resultados.

El primer resultado obtenido se trata de la matriz de confusión, representada en la Figura 9. En ella podemos observar la clasificación incorrecta de todos los sujetos del grupo 2 (infrasedados). En cambio, del grupo 0 (sobresedados), hay concordancia entre los médicos de que 20 sujetos si pertenecen a dicha clase

En la tabla 10 podemos ver los resultados obtenidos de esta comparación de variables, observando un *Acc* del 52% y un *Kappa* de 0.09.

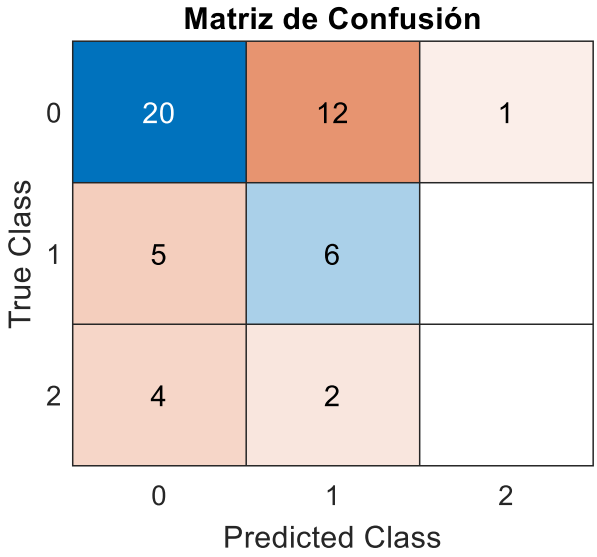

***Figura 9**. Matriz de confusión de las dos etiquetas de clasificación proporcionadas por el personal médico.*

***Tabla 10**. Métricas de rendimiento entre las dos variables objetivo del nivel de sedación.*

	Global	Sobresedados	Sedación adecuada	Infrasedados
Acc (%)	52	56	62	86
Se	0.52	0.61	0.54	0
Sp	0.76	0.47	0.64	0.97
PPV	0.52	0.69	0.3	0
NPV	0.75	0.39	0.83	0.88
F1-score	0.52	0.65	0.39	0
LR+	1.08			
LR-	0.32			
Kappa	0.09			

3.5.1 Modelo regresión logística con la etiqueta 1

En este apartado veremos los resultados obtenidos tras el entrenamiento y la validación con el conjunto de test del modelo de RL multinomial con la etiqueta real número 1.

El primer resultado obtenido se trata de la matriz de confusión, representada en la Figura 10. En ella podemos observar que este modelo es incapaz de clasificar correctamente a los sujetos del grupo 2 (infrasedados). En cambio, se obtienen mejores resultados respecto el número de VP clasificados como sobresedados.

La Figura 11 representa la curva *ROC* de clasificación de cada clase. La curva azul (representante de la clase 0) es la que alcanza mejores resultados, pues se aproxima más

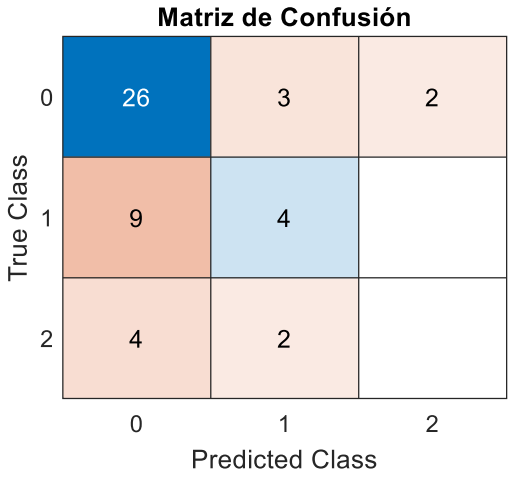

Figura 11. *Matriz de confusión modelo regresión logística con etiqueta 1.*

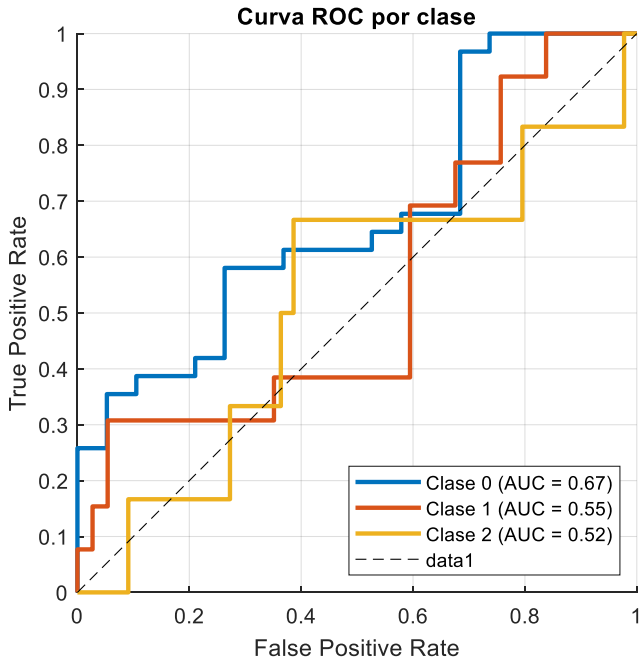

Figura 10. *Curvas ROC de cada clase del modelo regresión logística con la etiqueta 1*

a la esquina superior izquierda. Sin embargo, la curva amarilla (infrasedados) y la curva naranja (sedados adecuadamente) no presentan ningún patrón claro.

En la Tabla 11 se representan las métricas de rendimiento, tanto a nivel global como a nivel específico, en función del nivel de sedación. Este modelo presenta un *Acc* de 60% y un índice *Kappa* de 0.14, mostrando bajo nivel de concordancia entre las predicciones y la etiqueta real, aunque superior a la concordancia entre etiquetas reales (*Kappa* de 0.09).

El análisis por clase nos confirma que el modelo no logra identificar correctamente los casos de infrasedación, con valores de sensibilidad, *PPV* y *F1-score* nulos. Por otro lado, los resultados son más favorables para la clase sobresedados, donde se alcanza un *AUC* de 0.67 y el *F1-score* moderado, representando un equilibrio adecuado de los resultados.

Para comprobar el rendimiento del modelo en función de las características seleccionadas, decidimos realizar una gráfica de comparación de la exactitud del mismo respecto el número de características seleccionadas. En la Figura 12, se realizó respecto las características ya seleccionadas por *FCBF*. En cambio, en la Figura 13, la comparación se realizó con todo el conjunto de características ordenadas en función del *SU*.

En ambos casos, se observa que la máxima precisión del modelo se alcanza cuando se seleccionan las dos primeras características más relevantes. A partir de ese punto, la inclusión de más variable no favorece el rendimiento del modelo.

En la segunda gráfica vemos como a medida que incrementa el número de características seleccionadas (es decir, a lo largo que se escogen más características con menor valor de *SU*), la exactitud del modelo disminuye progresivamente. Esto respalda la utilidad de aplicar técnicas de selección de características, ya que permite mejorar el rendimiento del modelo evitando la inclusión de atributos irrelevantes o redundantes.

Tabla 11. *Métricas de rendimiento modelo regresión logística con etiqueta 1.*

	Global	**Sobresedados**	**Sedación adecuada**	**Infrasedados**
Acc (%)	60	64	72	84
Se	0.60	0.84	0.31	0
Sp	0.80	0.32	0.86	0.95
PPV	0.60	0.67	0.44	0
NPV	0.80	0.55	0.78	0.88
F1-score	0.6	0.74	0.36	0
LR+	1.5	1.22	2.27	0
LR-	0.25	0.51	0.8	1.05
Kappa	0.14			

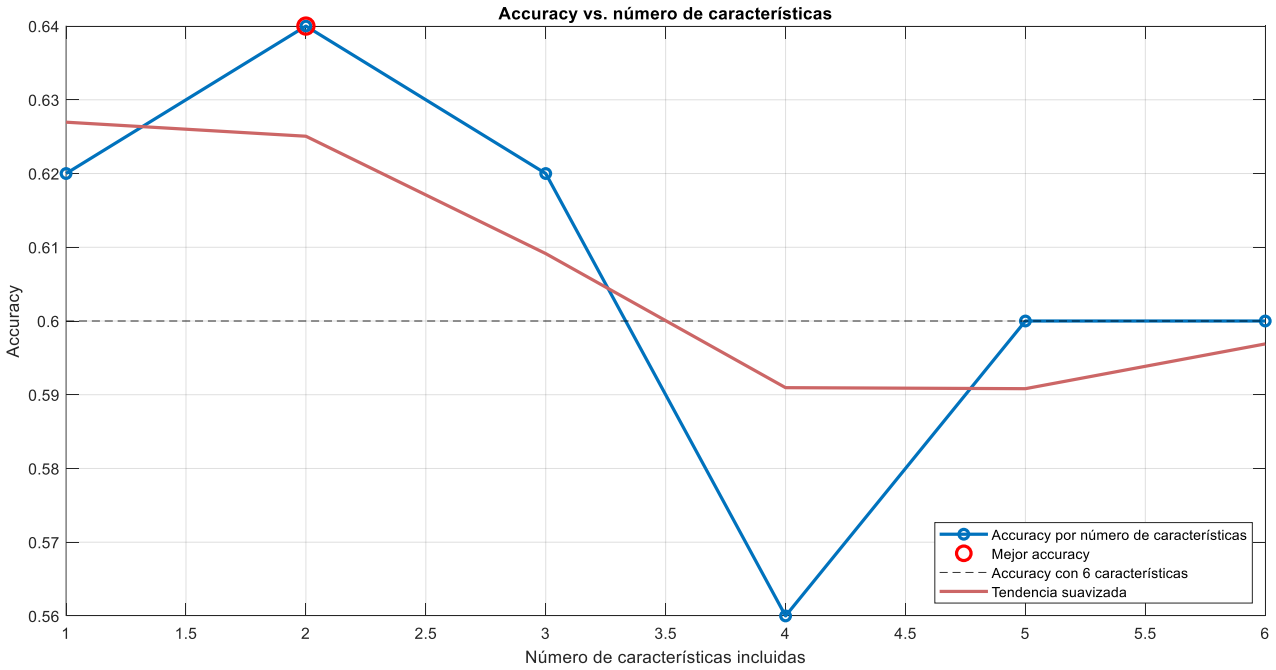

Figura 12. *Accuracy del modelo regresión logística con la etiqueta 1 en función del número de características seleccionadas dentro de las ya seleccionadas previamente con FCBF.*

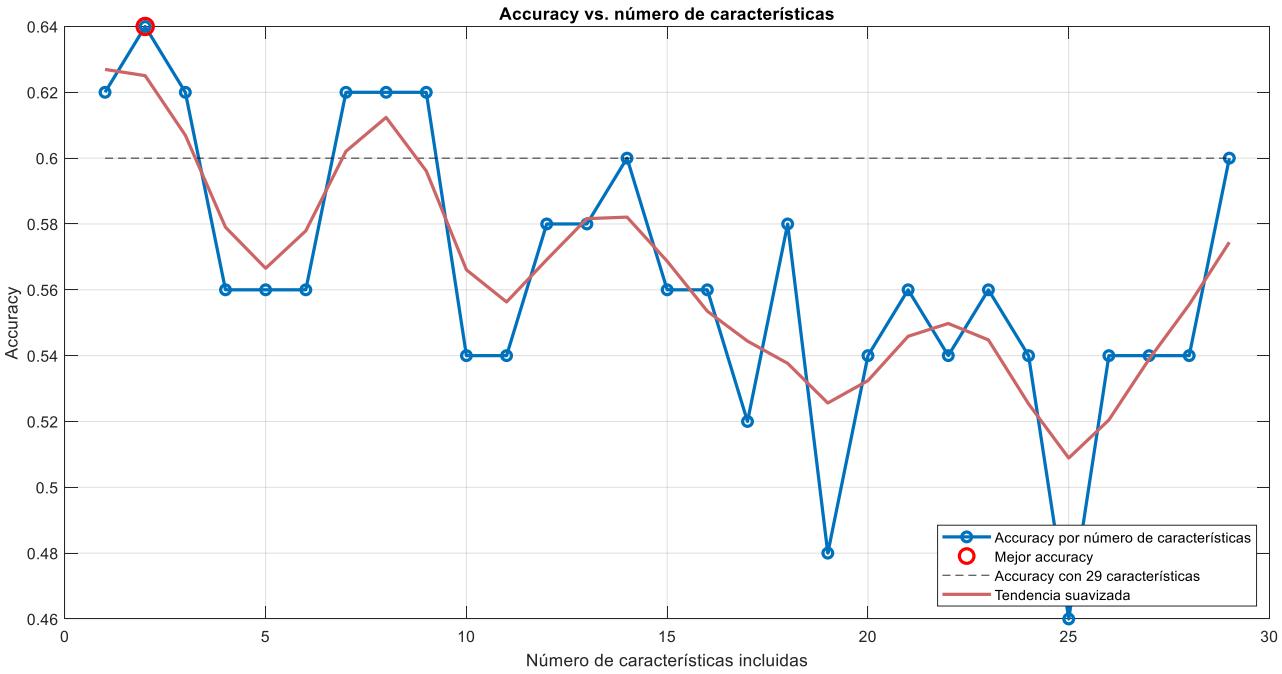

Figura 13*. Accuracy del modelo regresión logística con la etiqueta 1 en función del número de características seleccionadas de todas las variables del modelo ordenadas en base a Symmetrical Uncertainty.*

3.5.2 Modelo regresión logística con la etiqueta 2

En este apartado se comentan los resultados obtenidos con el modelo de RL utilizando la etiqueta 2. En este caso, no se observa una mejora en el comportamiento del modelo. El número de verdaderos positivos (diagonal de la matriz de confusión, representada en la Figura 14) es menor que en el modelo RL con la etiqueta 1.

En la Figura 15 se representa la curva *ROC* para cada clase o nivel de sedación. Esta gráfica permite evaluar el rendimiento del modelo de clasificación en términos de la capacidad para discriminar correctamente entre clases. Los resultados del *AUC* aparecen en la esquina inferior. Estos resultados indican que el modelo tiene un rendimiento ligeramente superior al azar (*AUC* = 0.5) para la clase 0, pero inferior al azar para las clases 1 y 2, cuyas curvas se sitúan por debajo de la diagonal de referencia.

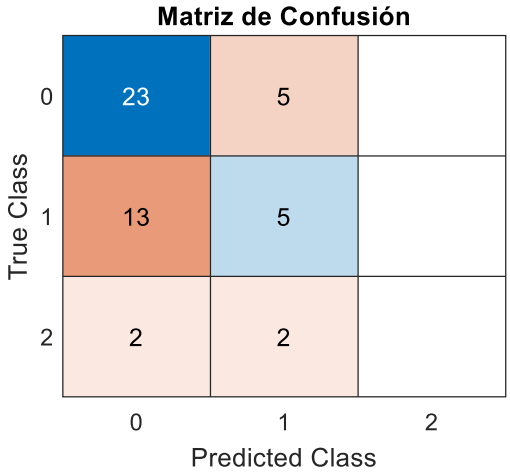

Figura 14*. Matriz de confusión modelo regresión logística con etiqueta 1.*

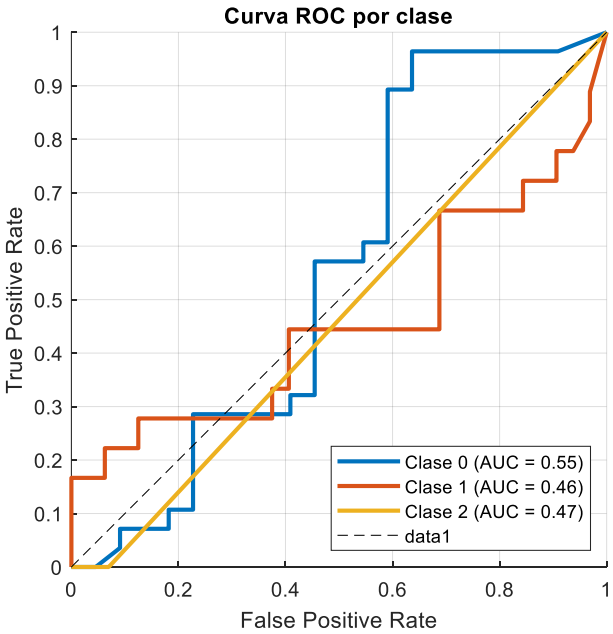

Figura 15. *Curvas ROC de cada clase del modelo regresión logística con la etiqueta 2*

En la Tabla 12 se recogen los resultados del cálculo de distintas métricas de rendimiento. Como vemos, a nivel general se obtuvo una *accuracy* del 56% y un índice *Kappa* de 0.10. Esto sugiere un bajo nivel de acuerdo, siendo peores los resultados que en el modelo anterior. Dentro del análisis por clase, cabe destacar que el grupo de infrasedados tampoco es clasificado correctamente, obtenido nuevamente valores nulos para sensibilidad, precisión y *F1-score*.

En cuanto a las Figuras 16 y 17, representan una comparación de la exactitud del modelo respecto el número de características seleccionadas. En el caso de las características seleccionadas con *FCBF*, podemos decir que no existe una diferencia significativa entre utilizar todas o solo algunas de las variables, pues la precisión puede variar de 0.6 a 0.56 como máximo. En cambio, si nos fijamos en la Figura 19 con el total de las características, la *accuracy* disminuye bruscamente hasta valores próximos a 0.4. Esto puede ser indicativo de que, al igual que en el apartado anterior, la selección de característica es una buena herramienta en el tratamiento de nuestro datos.

***Tabla 12**. Métricas de rendimiento modelo regresión logística con etiqueta 2.*

	Global	**Sobresedados**	**Sedación adecuada**	**Infrasedados**
Acc (%)	56	60	60	92
Se	0.56	0.82	0.28	0
Sp	0.78	0.32	0.78	1
PPV	0.56	0.61	0.42	0
NPV	0.78	0.58	0.66	0.92
F1-score	0.56	0.70	0.33	0
LR+	1.27	1.20	1.27	0
LR-	0.28	0.56	0.92	1
Kappa	0.10			

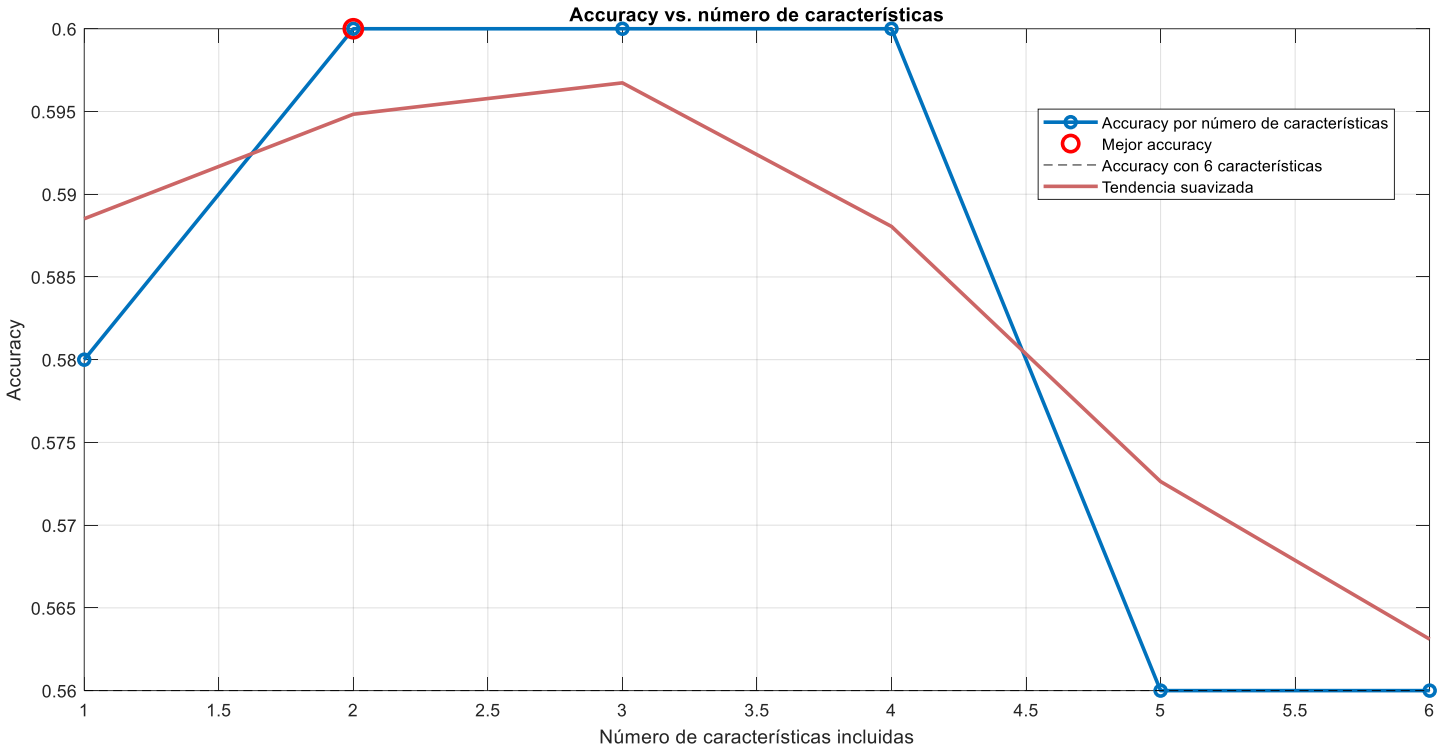

Figura 16. *Accuracy del modelo regresión logística con la etiqueta 2 en función del número de características seleccionadas dentro de las ya seleccionadas previamente con FCBF.*

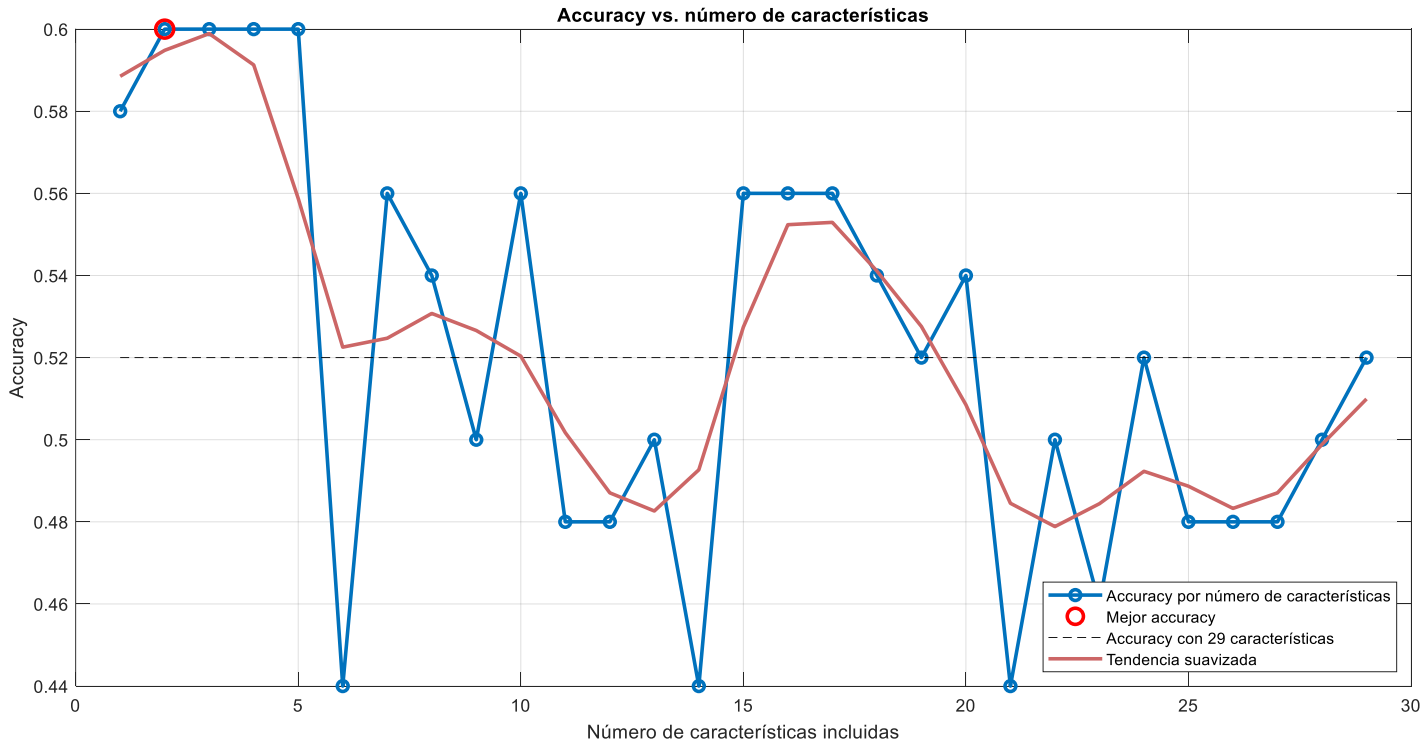

Figura 17. *Accuracy del modelo regresión logística con la etiqueta 2 en función del número de características seleccionadas de todas las variables del modelo ordenadas en base a Symmetrical Uncertainty.*

3.5.3 Modelo análisis discriminante lineal con la etiqueta 1

A continuación, se muestran los resultados obtenidos tras evaluar el rendimiento del modelo *LDA* entrenado con la etiqueta número 1 de la evaluación del nivel de sedación. El primer resultado obtenido de la validación es la matriz de confusión (Figura 18), donde vemos por primera vez la correcta clasificación de un sujeto de la clase 2. Además, los VP del resto de clases se mantiene en números similares a los modelos anteriores.

A partir de los datos de esta matriz de confusión, se obtuvieron las métricas representadas en la Tabla 13. En términos generales, el modelo obtuvo una exactitud del 56%, algo inferior a los modelos inferiores, con una sensibilidad y especificidad de 0.56 y 0.78, respectivamente. El índice *Kappa* tampoco fue significativamente superior a los otros modelos.

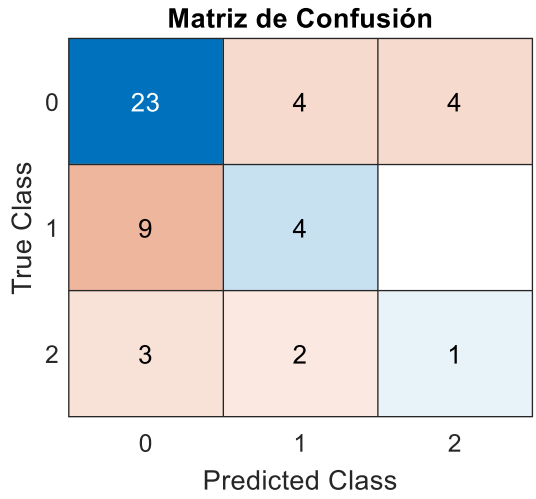

Figura 18. *Matriz de confusión modelo análisis discriminante lineal con etiqueta 1.*

Tabla 13. *Métricas de rendimiento modelo análisis discriminante lineal con etiqueta 1.*

	Global	Sobresedados	Sedación adecuada	Infrasedados
Acc (%)	56	60	70	82
Se	0.56	0.74	0.31	0.17
Sp	0.78	0.37	0.84	0.91
PPV	0.56	0.66	0.4	0.2
NPV	0.78	0.47	0.78	0.89
F1-score	0.56	0.70	0.35	0.18
LR+	1.27	1.17	1.9	1.83
LR-	0.28	0.70	0.83	0.92
Kappa	0.12			

Una de las gráficas más importantes para entender el grado de discriminación de nuestro modelo es la curva *ROC*. En la Figura 19, podemos ver la curva *ROC* para cada clase, es decir, podemos ver la capacidad discriminativa del modelo de *LDA* para clasificar a los pacientes en cada nivel de sedación. En este caso, la clase 0 (sobresedados) alcanza un *AUC* de 0.69, constituyendo el mejor rendimiento obtenido para esta clase entre todos los modelos evaluados hasta el momento. Sin embargo, los resultados siguen siendo próximos al azar para las clases restantes.

Las gráficas representadas en las Figuras 20 y 21, muestran como el modelo alcanza el máximo rendimiento con dos características, obteniendo una exactitud de 0.64. Tras dicho punto, en la gráfica que describe la *accuracy* frente todo el conjunto de características, se puede comprobar como el rendimiento fluctúa con diversos máximos locales.

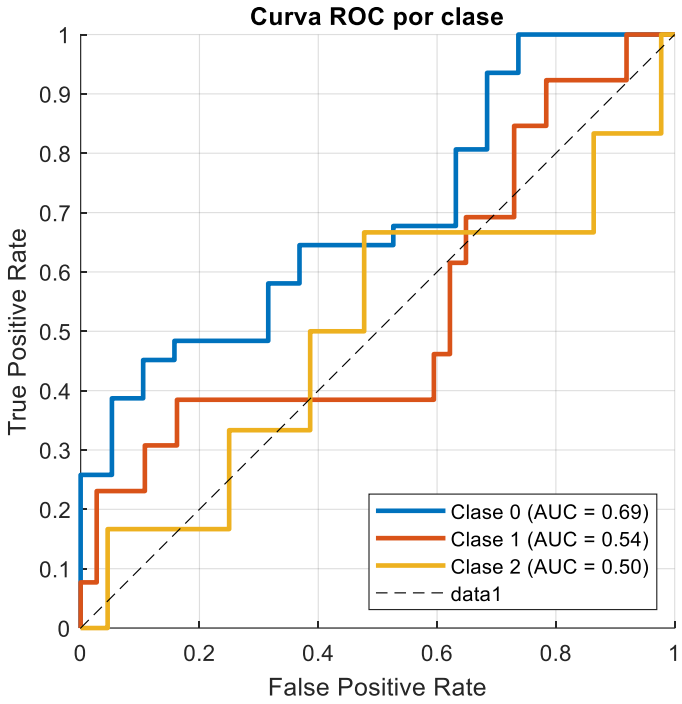

Figura 19. *Curvas ROC de cada clase del modelo análisis del discriminante lineal con la etiqueta 1.*

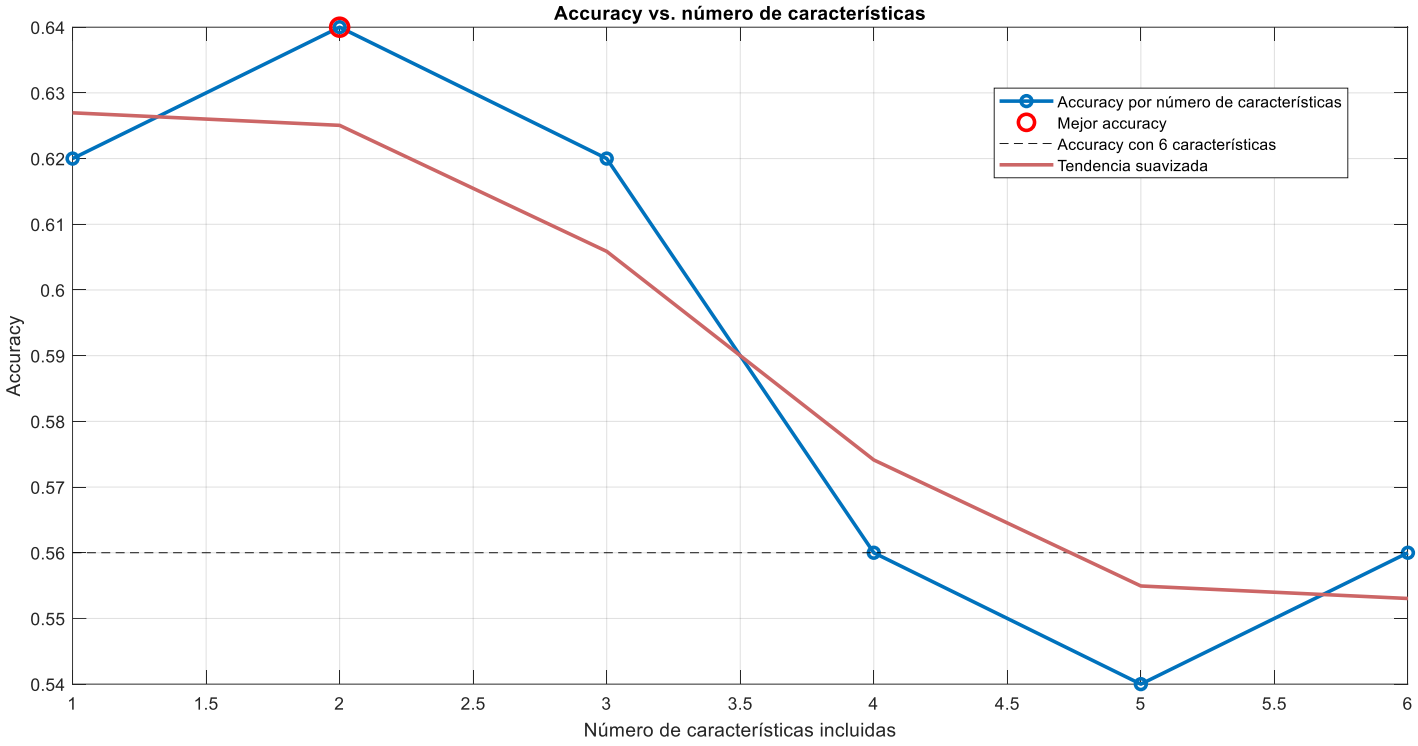

Figura 20. *Accuracy del modelo análisis del discriminante lineal con la etiqueta 1 en función del número de características seleccionadas dentro de las ya seleccionadas previamente con FCBF*

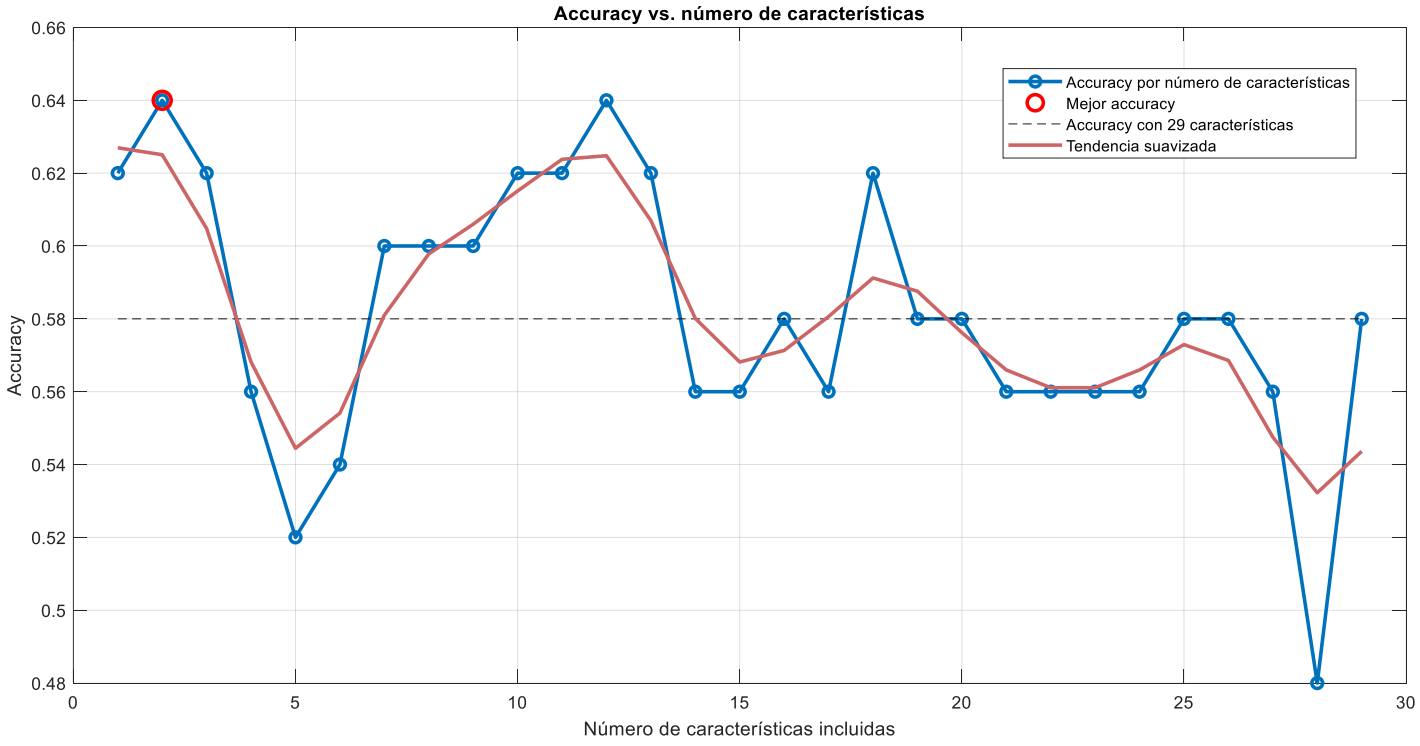

Figura 21*. Accuracy del modelo análisis del discriminante lineal con la etiqueta 1 en función del número de características seleccionadas de todas las variables del modelo ordenadas en base a Symmetrical Uncertainty*

3.5.4 Modelo análisis discriminante lineal con la etiqueta 2

En el presente punto se muestran los resultados obtenidos tras el entrenamiento y evaluación del modelo de *LDA* con la etiqueta o *target* número 2 sobre los niveles de sedación.

En la Tabla 14, vemos los resultados de las métricas de rendimiento del algoritmo. Estas métricas datos fueron calculadas a partir del número de VP, VN, FP y FN, valores reflejados en la matriz de confusión (Figura 22). En líneas generales, este modelo no nos ofrece mejores métricas que cualquiera de los anteriores.

Tabla 14. *Métricas de rendimiento modelo análisis del discriminante lineal con etiqueta 2.*

	Global	Sobresedados	Sedación adecuada	Infrasedados
Acc (%)	58	62	62	92
Se	0.58	0.86	0.28	0
Sp	0.79	0.32	0.81	1
PPV	0.58	0.62	0.45	0
NPV	0.79	0.64	0.67	0.92
F1-score	0.58	0.72	0.34	0
LR+	1.38	1.26	1.48	0
LR-	0.27	0.45	0.89	1
Kappa	0.13			

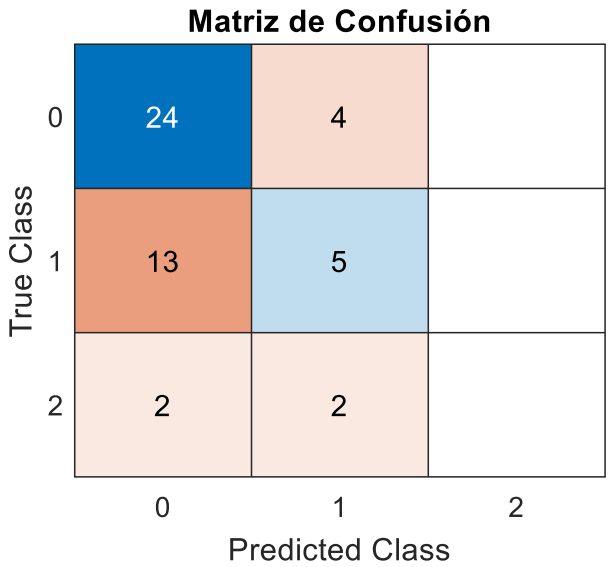

Figura 22. *Matriz de confusión modelo análisis del discriminante lineal con etiqueta 2.*

Al observar las curvas *ROC* de cada clase (Figura 23), se aprecia que el modelo *LDA* presenta un rendimiento especialmente bajo para la clase 2, con un valor de *AUC* de apenas 0.23. Esto indica que el algoritmo tiene una capacidad de discriminación muy limitada para esta clase.

Si nos fijamos en las curvas *ROC* de cada clase (Figura 23), se aprecia que el modelo *LDA* entrenado con la etiqueta 2 presenta un rendimiento especialmente bajo para la clase 2, con un valor de *AUC* de 0.23. Esto indica que el algoritmo tiene una capacidad de discriminación muy limitada para dicha clase. Por otra parte, los valores de *AUC* para la clase 0 y 1 son próximos a 0.5, lo que tiende a una naturaleza azarosa.

En cuanto las curvas de exactitud frente el número de características totales (Figura 25), vemos que este modelo presenta una *accuracy* bastante constante a lo largo de todas las características, pues varía en un rango entre 0.5 y 0.62. Este resultado también se puede comprobar en la Figura 24, donde se representa las *accuracy* respecto el número de características seleccionadas del total de características elegidas con *FCBF*.

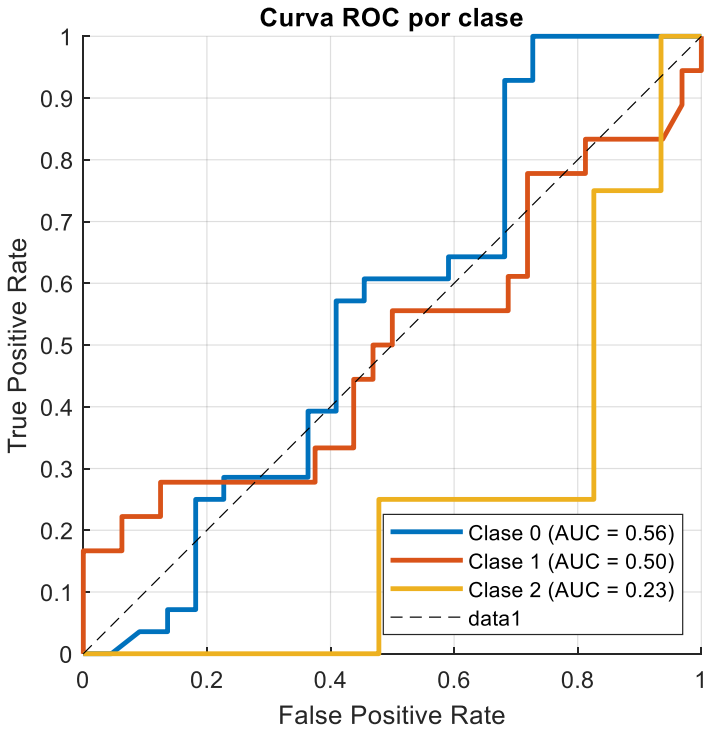

Figura 23. *Curvas ROC de cada clase del modelo de análisis del discriminante lineal con la etiqueta 2.*

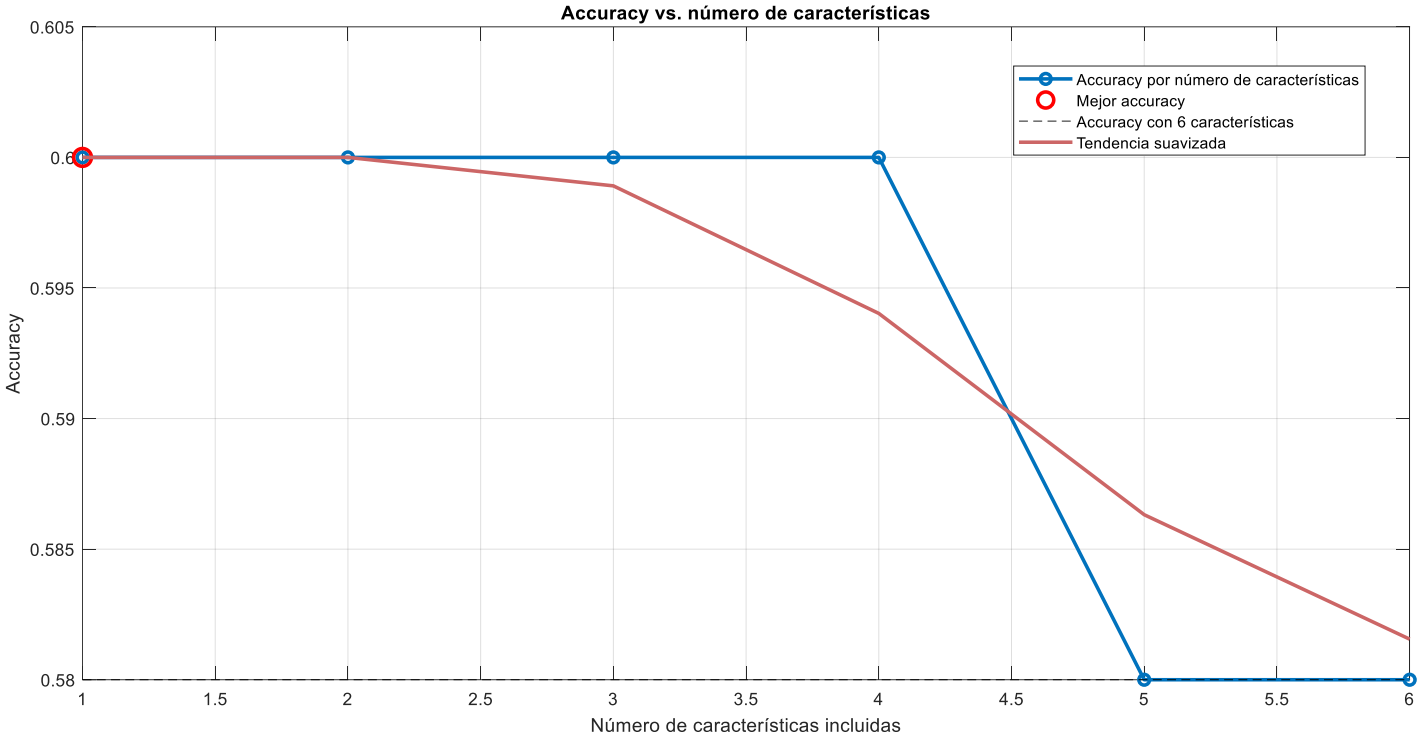

Figura 24. *Accuracy del modelo de análisis del discriminante lineal con la etiqueta 1 en función del número de características seleccionadas dentro de las ya seleccionadas previamente con FCBF.*

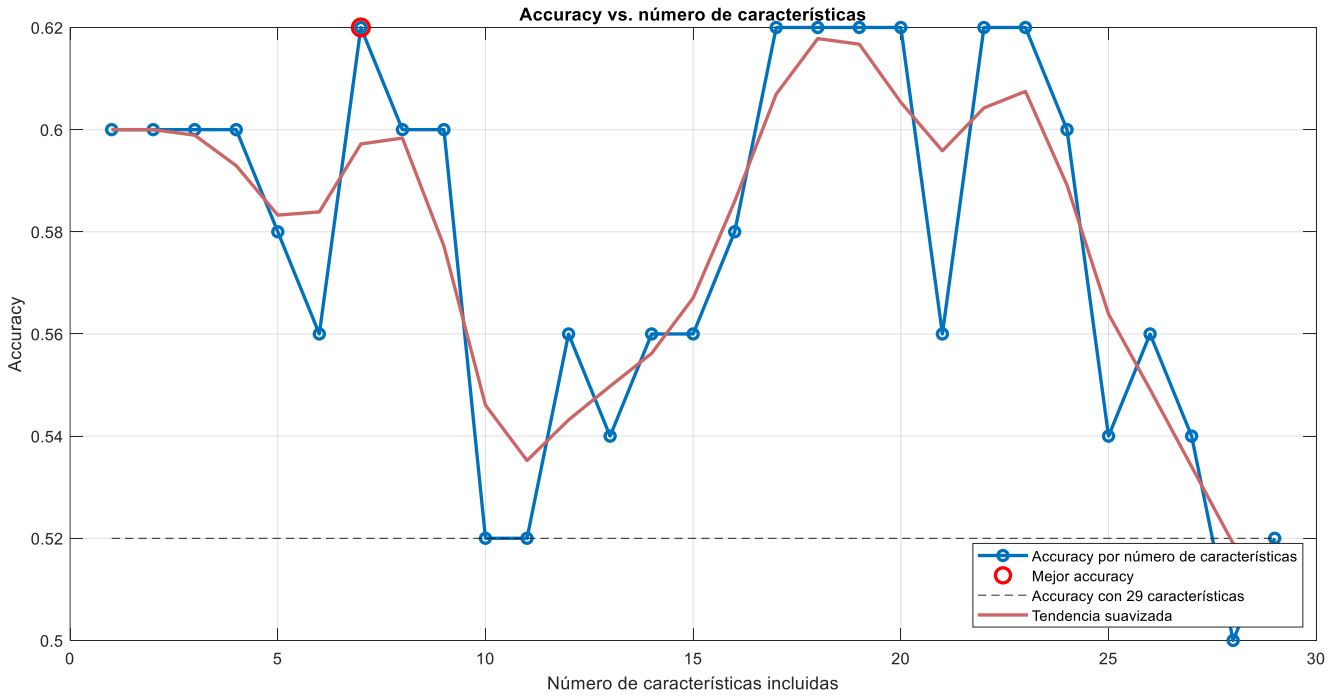

Figura 25. *Accuracy del modelo de análisis del discriminante lineal con la etiqueta 2 en función del número de características seleccionadas de todas las variables del modelo ordenadas en base a Symmetrical Uncertainty*

3.5.5 Modelo AdaBoostM2 con la etiqueta 1

A continuación, se presentan los resultados obtenidos con el algoritmo *AdaBoostM2* entrenado con la etiqueta 1. Este modelo, basado en la técnica de *boosting*, se caracteriza por su mayor complejidad en comparación con otros métodos evaluados.

En la Tabla 15 se muestran las métricas de rendimiento obtenidas, tanto globales como por clase. El modelo alcanzó una exactitud del 66 % y un coeficiente *Kappa* de 0.33, los valores más altos registrados hasta el momento. En la matriz de confusión (Figura 26) se observa la correcta clasificación de tres sujetos pertenecientes a la clase 2 (infrasedados). Aunque en esta clase los valores de sensibilidad (0.38) y especificidad (0.36) son bajos, este fue el único modelo que logró identificar correctamente algún caso. Para la clase de sobresedación, el modelo alcanzó una sensibilidad de 0.81 y un *F1-score* de 0.78. En contraste, los resultados para la clase de sedación adecuada fueron más bajos, con una sensibilidad de 0.38, especificidad de 0.36 y un *F1-score* de 0.37.

Tabla 15. *Métricas de rendimiento modelo AdaBoostM2 con la etiqueta 1.*

	Global	**Sobresedados**	**Sedación adecuada**	**Infrasedados**
Acc (%)	66	72	66	94
Se	0.66	0.81	0.38	0.5
Sp	0.83	0.76	0.36	1
PPV	0.66	0.62	0.45	1
NPV	0.83	0.65	0.78	0.93
F1-score	0.66	0.78	0.37	0.67
LR+	1.94	1.92	1.58	2.25
LR-	0.20	0.33	0.82	0.5
Kappa	0.33			

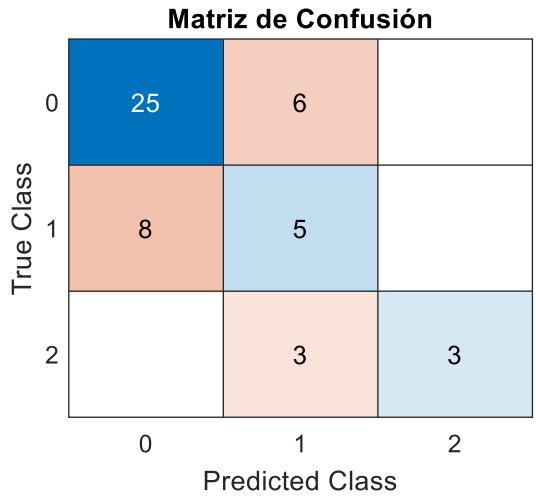

Figura 26*. Matriz de confusión modelo AdaBoostM2 con etiqueta 1.*

En la Figura 27 se presentan las curvas *ROC* correspondientes a cada clase. La clase 0 (sobresedados) muestra el mejor rendimiento, con una curva más próxima a la esquina superior izquierda del gráfico y un valor de *AUC* de 0.81. La clase 2 (infrasedados) alcanza un *AUC* de 0.70, lo que representa una mejora considerable en comparación con los resultados obtenidos por otros modelos. Por otro lado, la clase 1 (sedación adecuada) registra un *AUC* de 0.52, un valor próximo al esperado por el azar.

Respecto a las gráficas representadas en las Figuras 28 y 29, podemos observar que la mayor presión se alcanza con las dos primeras características en el caso del grupo de características seleccionadas con *FCBF*, mientras que, en el conjunto total de características, la mayor precisión se da al incluir 5 características, con un valor de 0.80. a partir de ese momento el rendimiento del modelo empeora notablemente, hasta valores de 0.66.

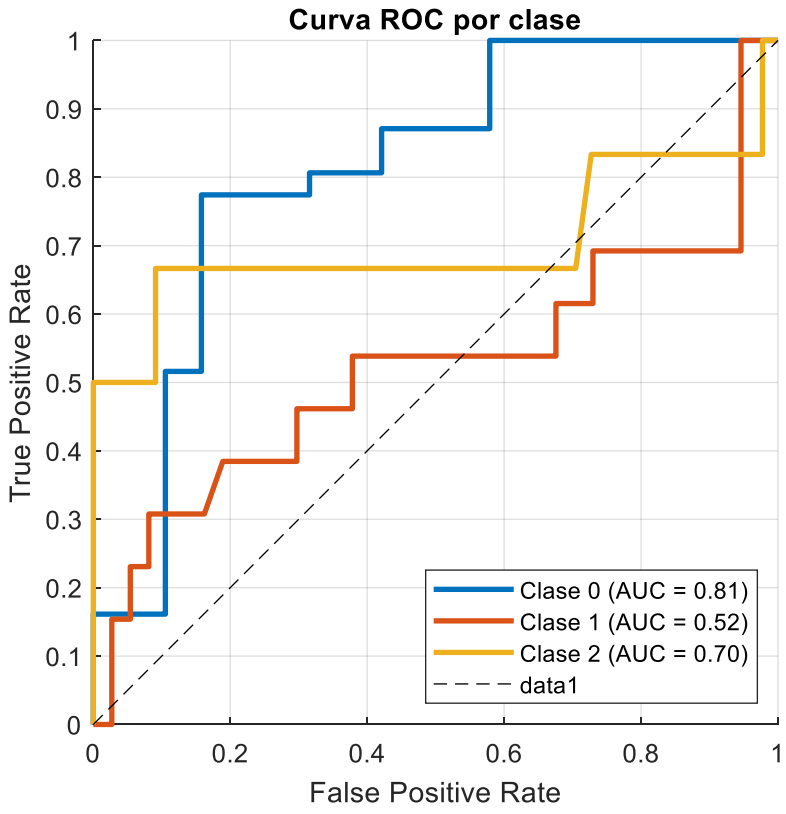

Figura 27. *Curvas ROC de cada clase del modelo AdaBoostM2 con la etiqueta 1.*

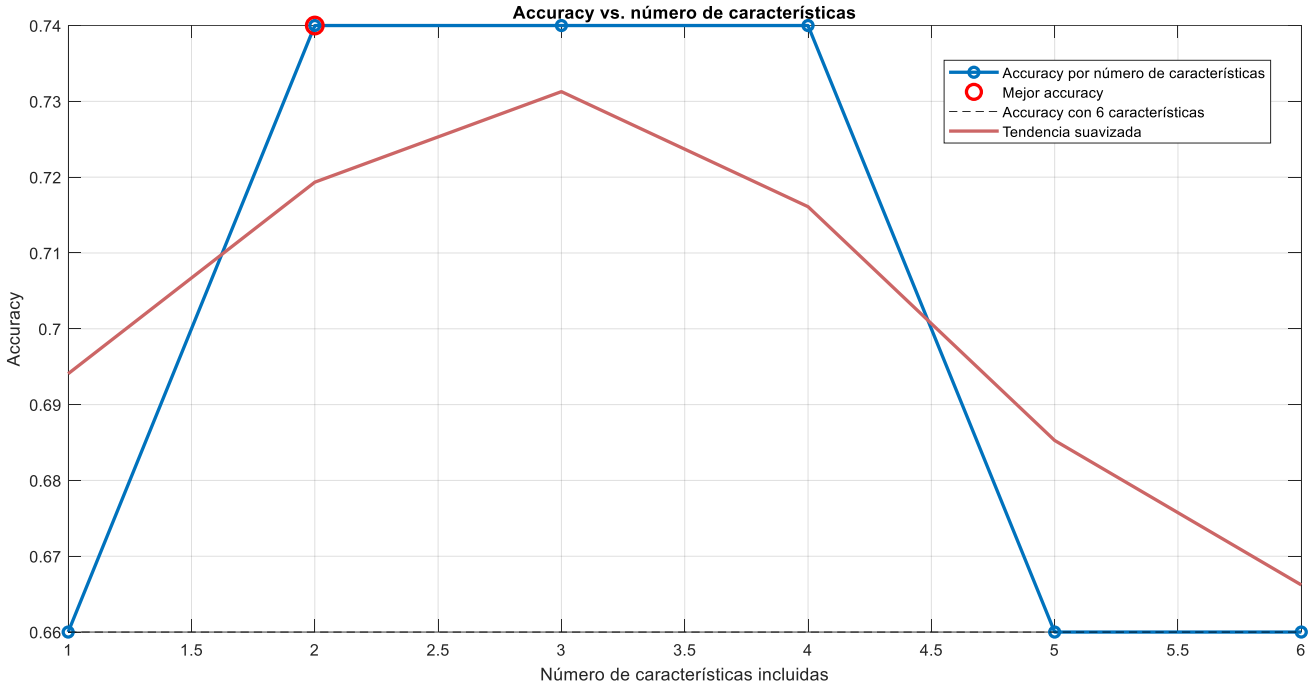

***Figura 28**. Accuracy del modelo AdaBoostM2 con la etiqueta 1 en función del número de características seleccionadas dentro de las ya seleccionadas previamente con FCBF.*

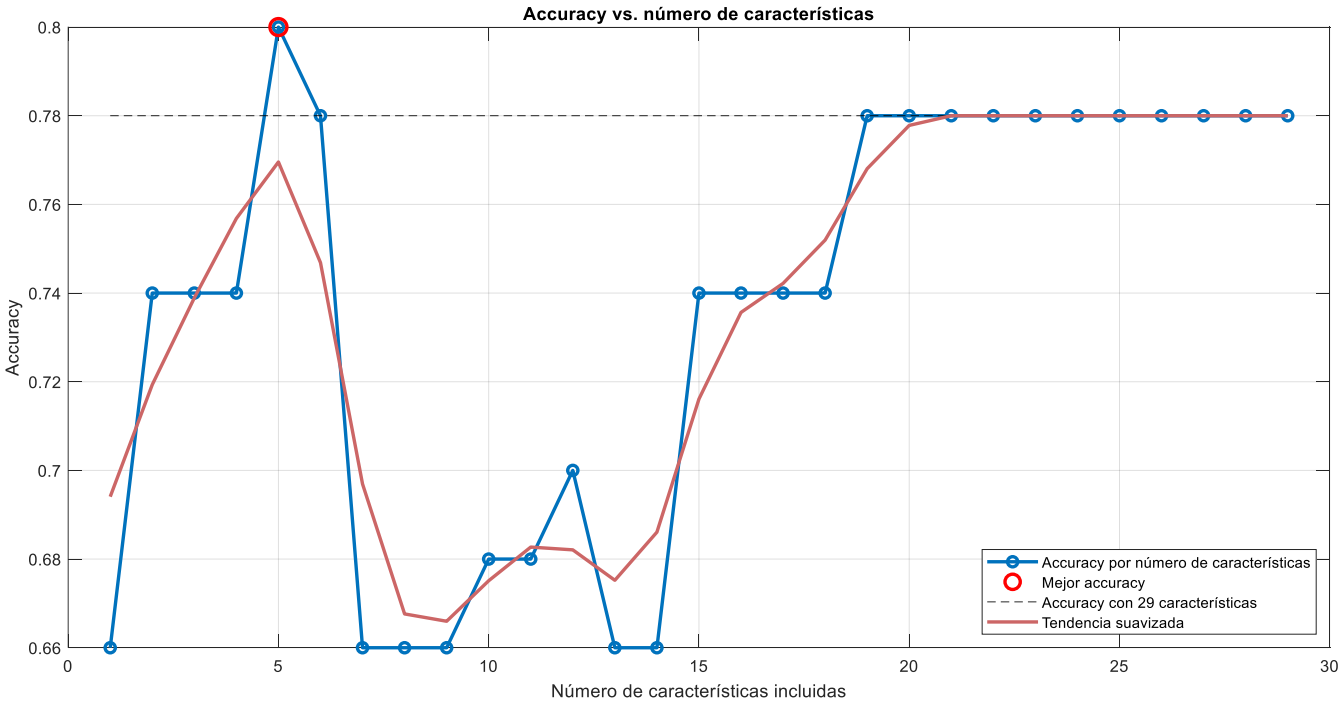

***Figura 29**. Accuracy del modelo AdaBoostM2 con la etiqueta 1 en función del número de características seleccionadas de todas las variables del modelo ordenadas en base a Symmetrical Uncertainty.*

3.5.6 Modelo AdaBoostM2 con la etiqueta 2

El modelo *AdaBoostM2* también fue evaluado utilizando la etiqueta 2 para la clasificación del nivel de sedación. En este caso, los resultados globales fueron similares a los obtenidos con la etiqueta 1, alcanzando una *accuracy* del 68 % y un índice *Kappa* de 0.37, según recoge la Tabla 16. No obstante, el rendimiento por clase muestra diferencias relevantes, especialmente en las categorías de sedación adecuada e infrasedación. La clase de sobresedados mantiene valores similares al modelo anterior, mientras que la clasificación de la sedación adecuada experimenta una mejora notable, con una precisión del 72 %, especificidad del 82 % y un *F1-score* de 0.59. Sin embargo, la clasificación de la clase infrasedados presenta un deterioro, ya que, según se observa en la matriz de confusión (Figura 30) y en los valores recogidos en la tabla, no se obtuvo ningún verdadero positivo.

Tabla 16. *Métricas de rendimiento modelo AdaBoostM2 con etiqueta 2.*

	Global	**Sobresedados**	**Sedación adecuada**	**Infrasedados**
Acc (%)	68	72	72	92
Se	0.68	0.86	0.56	0
Sp	0.84	0.55	0.82	1
PPV	0.68	0.71	0.62	0
NPV	0.84	0.75	0.76	0.92
F1-score	0.68	0.77	0.59	0
LR+	2.12	1.89	2.96	0
LR-	0.19	0.26	0.55	1
Kappa	0.37			

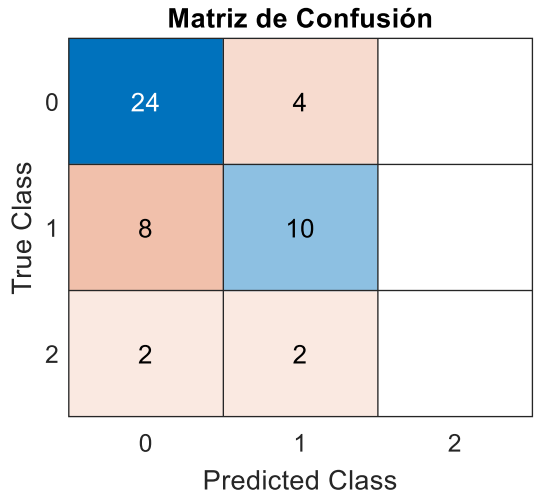

Figura 30. *Matriz de confusión modelo AdaBoostM2 con etiqueta 2.*

Este descenso en el rendimiento del modelo para la clasificación de pacientes infrasedados queda reflejado en la curva *ROC* para dicha clase. Tal como se muestra en la Figura 31, la clase 2 (infrasedados, línea amarilla) presenta un *AUC* de 0.31, lo que se traduce en una curva *ROC* situada próxima a la esquina inferior derecha del gráfico.

En cuanto la Figura 32, que muestra la precisión en función del número de variables seleccionadas para entrenar el modelo dentro de las características seleccionadas con *FCBF*, vemos como la *accuracy* varía en un rango de 56% a 68%, y los mejores valores se dan al coger 4, 5 o todo el conjunto entero de características.

La Figura 33 muestra una gráfica parecida, donde se compara la exactitud del modelo en base al número de características seleccionadas respecto el total y ordenadas en base al *SU*. Vemos como se alcanza la máxima *accuracy* seleccionando las 7 primeras características, y a partir de dicho momento, el rendimiento empeora significativamente hasta alcanzar valores de 0.5.

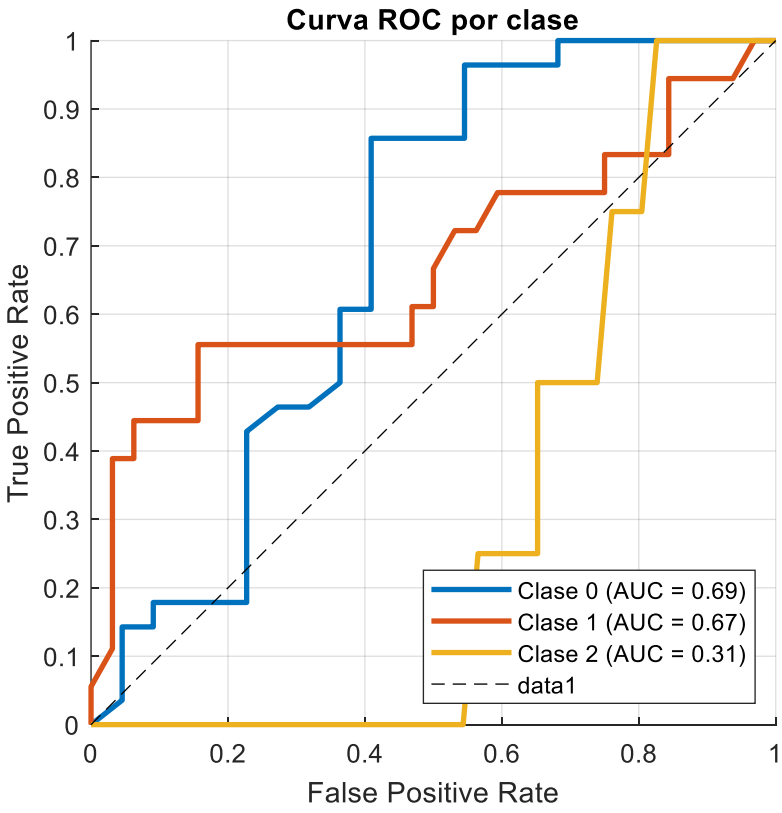

Figura 31. *Curvas ROC de cada clase del modelo AdaBoostM2 con la etiqueta 2.*

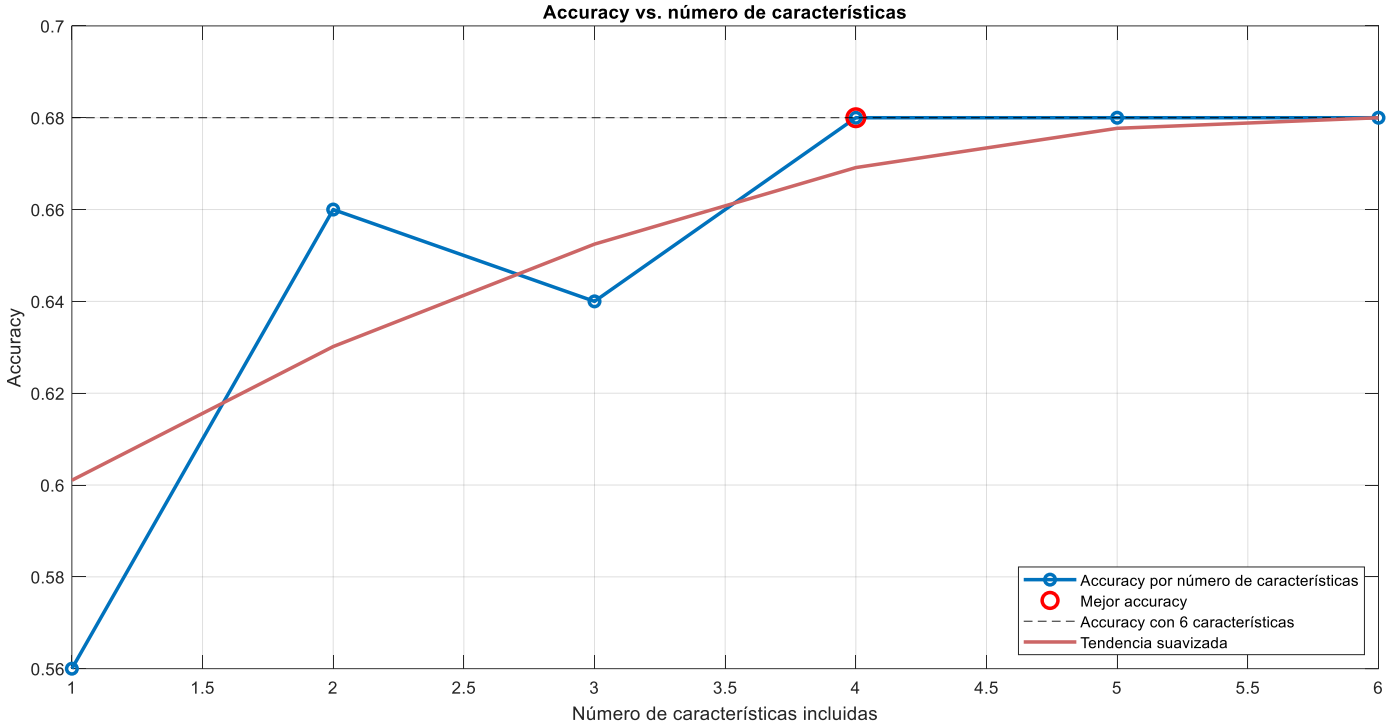

Figura 32. *Accuracy del modelo AdaBoostM2 con la etiqueta 2 en función del número de características seleccionadas dentro de las ya seleccionadas previamente con FCBF.*

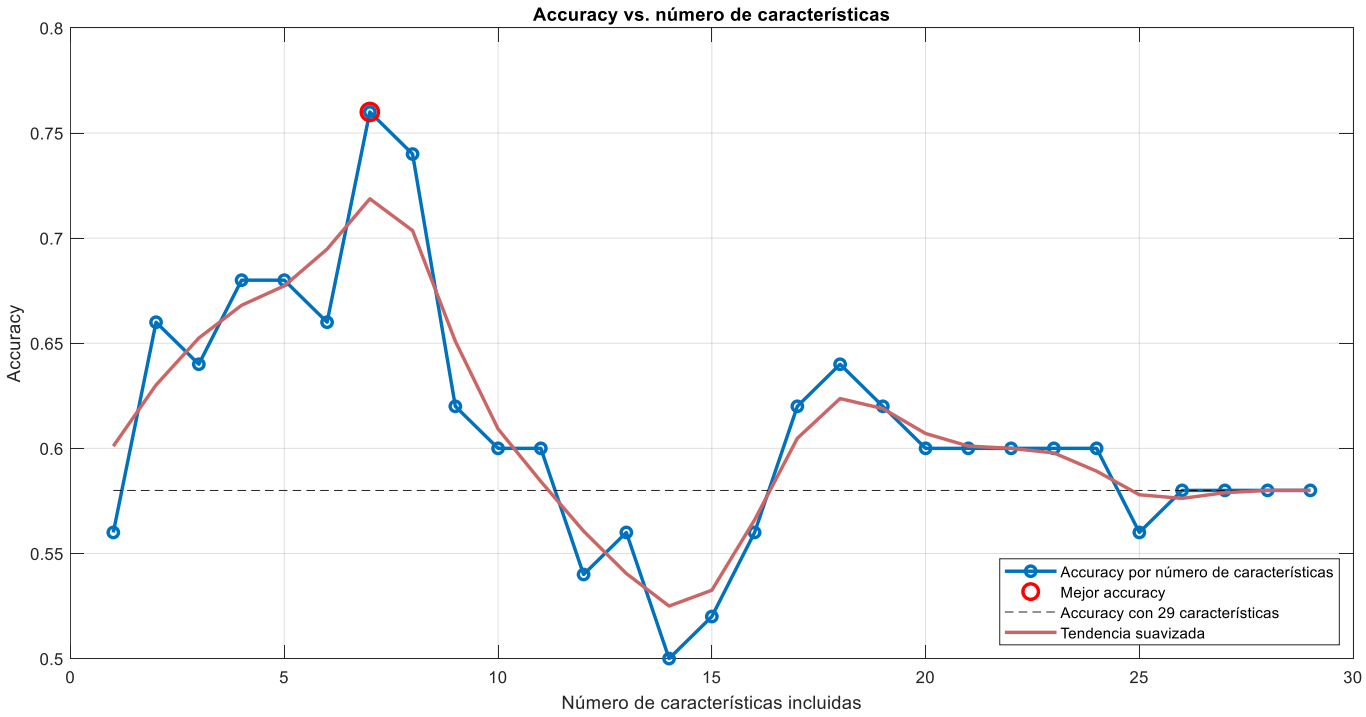

Figura 33. *Accuracy del modelo AdaBoostM2 con la etiqueta 2 en función del número de características seleccionadas de todas las variables del modelo ordenadas en base a Symmetrical Uncertainty.*

En la Tabla 17 se recogen todos los resultados del rendimiento global de cada modelo y en negrita se encuentran marcados los mejores resultados de cada métrica. Por otro lado, en la Tabla 18 se recogen los resultados de los modelos por clase. La clase 0 (sobresedados) está marcada en azul, la clase 1 (sedación adecuada) de naranja y la clase 2 (infrasedados) de verde. En negrita, y con el color correspondiente a cada clase, se encuentran subrayados los mejores resultados de cada métrica de rendimiento para cada clase.

***Tabla 17**. Resultados de las métricas de rendimiento globales de los modelos*

	RL 1	**RL 2**	**LDA 1**	**LDA 2**	**AdaBoost 1**	**AdaBoost 2**
Acc (%)	60	56	56	58	66	**68**
Se	0.60	0.56	0.56	0.58	0.66	**0.68**
Sp	0.80	0.78	0.78	0.79	0.83	**0.84**
PPV	0.60	0.56	0.56	0.58	0.66	**0.68**
NPV	0.80	0.78	0.78	0.79	0.83	**0.84**
F1-score	0.6	0.56	0.56	0.58	0.66	**0.68**
LR+	1.5	1.27	1.27	1.38	1.94	**2.12**
LR-	0.25	0.28	0.28	0.27	0.20	**0.19**
Kappa	0.14	0.10	0.12	0.13	0.33	**0.37**

***Tabla 18.** Resultados de las metricas de rendimiento por clase de todos los modelos*

modelo	**clase**	***Acc (%)***	***Se***	***Sp***	**PPV**	**NPV**	**F1-score**	**LR+**	**LR-**
	0	64	0.84	0.32	0.67	0.55	0.74	1.22	0.51
RL 1	1	**72**	0.31	**0.86**	0.44	**0.78**	0.36	2.27	0.8
	2	84	0	0.95	0	0.88	0	0	1.05
	0	60	0.82	0.32	0.61	0.58	0.70	1.20	0.56
RL2	1	60	0.28	0.78	0.42	0.66	0.33	1.27	0.92
	2	92	0	1	0	0.92	0	0	1
	0	60	0.74	0.37	0.66	0.47	0.70	1.17	0.7
LDA1	1	70	0.31	0.84	0.4	**0.78**	0.35	1.90	0.83
	2	82	0.17	0.91	0.2	0.89	0.18	1.83	0.92
	0	62	**0.86**	0.32	0.62	0.64	0.72	1.26	0.45
LDA 2	1	62	0.28	0.81	0.45	0.67	0.34	1.48	0.89
	2	92	0	1	0	0.92	0	0	1
	0	**72**	0.81	**0.76**	0.62	0.65	**0.78**	**1.92**	**0.33**
AB 1	1	66	0.38	0.36	0.45	**0.78**	0.37	1.58	0.82
	2	**94**	**0.5**	**1**	**1**	**0.93**	**0.67**	**2.25**	**0.5**
	0	**72**	**0.86**	0.55	**0.71**	**0.75**	0.77	1.89	0.6
AB 2	1	**72**	**0.56**	0.82	**0.62**	0.76	**0.59**	**2.96**	**0.55**
	2	92	0	1	0	0.92	0	0	1

3.6 Resultados de la aplicación técnicas inteligencia artificial explicable

Tras realizar el estudio del rendimiento de los diferentes modelos con las diferentes etiquetas, se aplicaron las técnicas de *SHAP* en el modelo de mayor rendimiento y solo para aquellas muestras clasificadas correctamente. En nuestro caso, optamos por el modelo *AdaBoostM2* entrenado con la etiqueta 1, ya que es el único que muestra un buen desempeño general, incluyendo una clasificación efectiva de los infrasedados.

En la Figura 34 se muestra la distribución de los valores *SHAP* para cada una de las variables más influyentes del estudio por clase. En el eje Y se representan las características del modelo, mientras que en el eje X se presentan los valores *SHAP*, es decir, cuanto contribuye esa característica a aumenta o disminuir la predicción.

Cada una de las gráficas representa la distribución de una de las clases, y cada punto constituye una muestra o paciente, de tal forma que se ordenan por colores. Las instancias en rosa presentan valores *SHAP* más altos para dicha variable, y en azul, valores más bajos.

Otra forma de analizar gráficamente la importancia global de las variables es mediante el valor medio absoluto de *SHAP* para cada una de ellas. Este valor corresponde a la media de los valores *SHAP* de todos los pacientes para cada variable. De esta forma, obtenemos

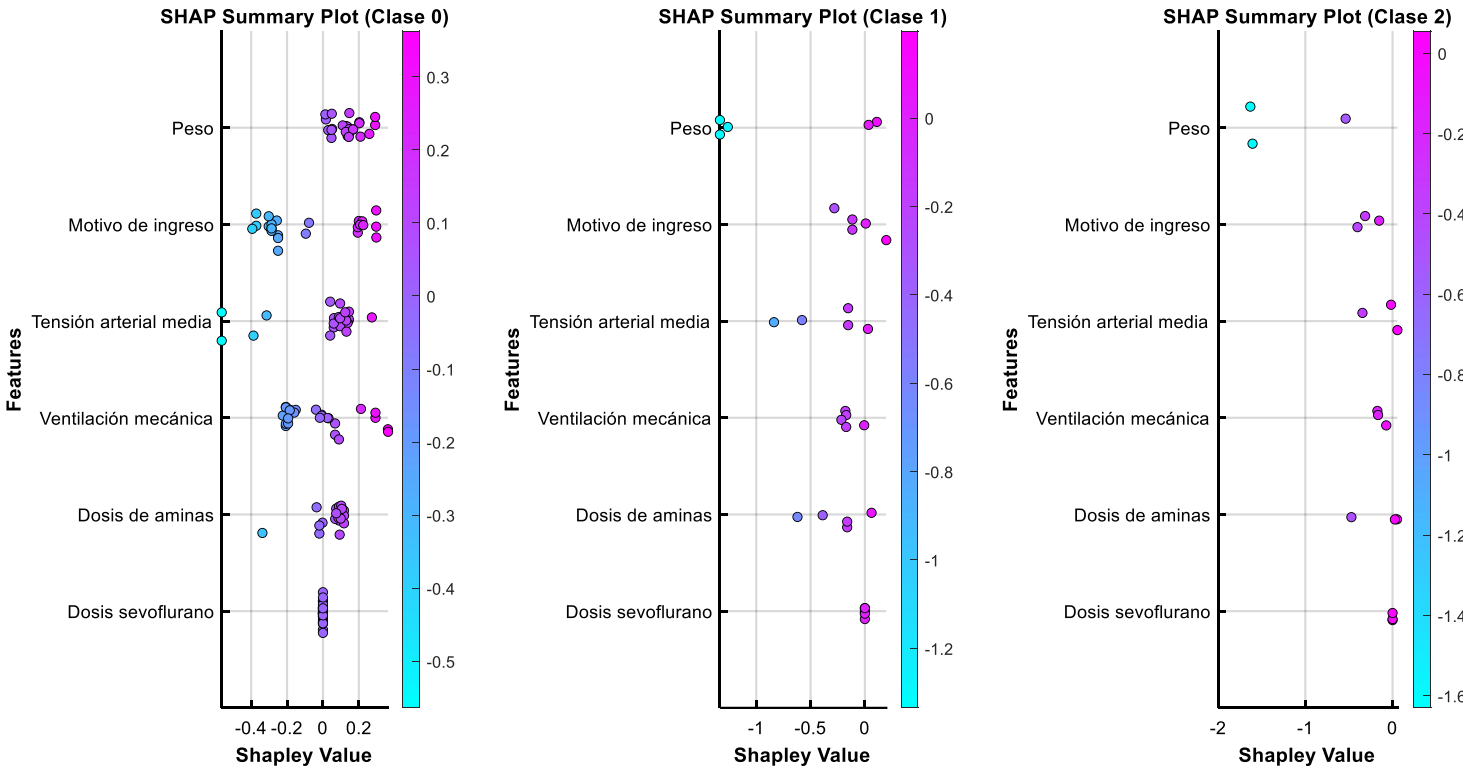

Figura 34. *Gráfica de valores SHAP de cada paciente para cada variable.*

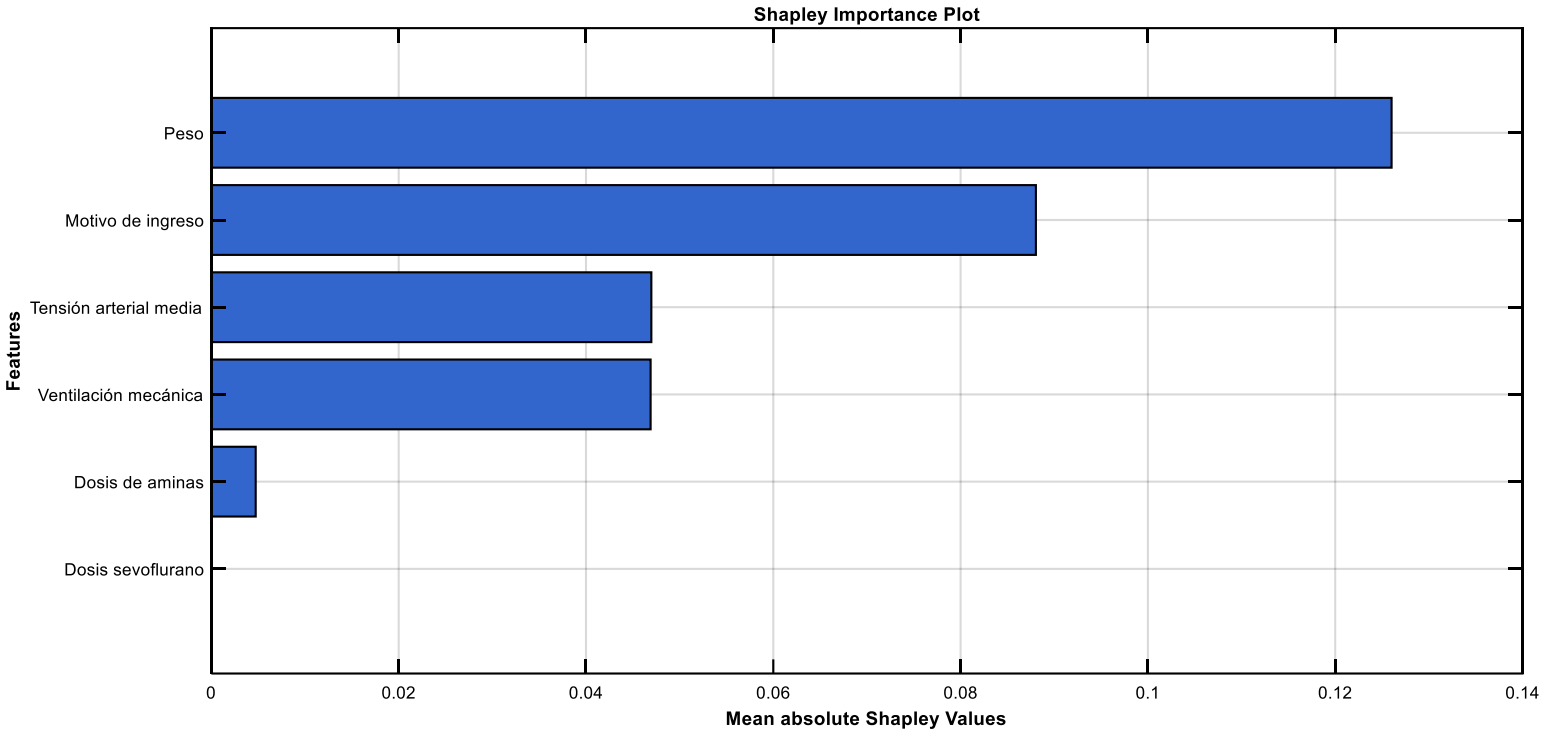

Figura 35. *Gráfico de barras de importancia de cada característica en base al valor medio absoluto de SHAP.*

una visión general de cual serían las características más importantes para el modelo a la hora de realizar predicciones. Esta información se ve reflejada en la Figura 35, donde vemos que la variable peso presenta un valor más alto, mientras que la variable sedación sevoflurano inhalado presenta una contribución nula.

Capítulo 4. Discusión

4.1 Introducción

El correcto ajuste de la dosis de fármacos analgésicos y sedantes es esencial para asegurar la comodidad y el confort de los pacientes en UCI. Actualmente, el principal método para evaluar el nivel de sedación consiste en el uso del BIS. Sin embargo, su aplicación supone un elevado gasto y un aumento de la carga de trabajo del profesional sanitario. Por ello, en este TFG se han aplicado algoritmos de *ML* para desarrollar un modelo capaz de predecir la clasificación del nivel de sedación del paciente crítico mediante el análisis de diferentes variables clínicas, sociodemográficas, escalas y dosis de fármacos.

En la actualidad, se ha incrementado el interés por el uso de la IA en el ámbito sanitario, dado que el campo de la medicina se caracteriza por el manejo de enormes cantidades de datos con información muy relevante. Las enormes dimensiones de estos datos requieren la utilización de algoritmos de gran complejidad, donde a menudo resulta compleja su interpretabilidad. Por ello, en este trabajo se plantea también el uso de técnicas de *XAI* para extraer información del modelo sobre la relevancia de las variables a la hora de realizar la predicción.

A lo largo de este capítulo se discutirán los distintos resultados obtenidos. Primero se analizarán las diferencias estadísticas y las asociaciones existentes entre las variables respecto cada nivel de sedación. Después, se discutirán los resultados obtenidos tras el proceso de selección de características. Seguidamente, se realizará un análisis de los diversos modelos implementados, explicando cual es el óptimo de ellos tan a nivel global como por clases. Tras ello, se analizarán los resultados extraídos al aplicar *SHAP*. Por último, comentaremos las limitaciones del trabajo.

4.2 Interpretación de los hallazgos estadísticos

Uno de los aspectos analizados en el presente trabajo ha sido la relación entre la variable objetivo, es decir, el nivel de sedación, con cada variable bajo estudio. Para ello, realizamos un análisis de hipótesis a través de *p*-valores y calculamos la correlación de Spearman de dicha asociación.

En la primera columna de la Tabla 9 se encuentran los *p*-valores de cada variable respecto la variable a predecir. Estos valores también se pueden ver representado en la gráfica de barras de la Figura 5. La mayoría de las variable presentan *p*-valores superiores a 0.05, indicando que no existe evidencia estadística suficiente como para afirmar que existe una relación entre el nivel de significación y las variables. Esto podría indicar la necesidad de un tamaño muestral mayor para detectar posibles diferencias más precisas.

Por otra parte, sí que existen algunas variables con un *p*-valor menor de 0.05, pudiendo decir que presenta correlaciones estadísticamente significativas con el nivel de sedación, y, por tanto, podrían tener un papel importante en el ámbito clínico. Estas variables aparecen en rojo en la gráfica mencionada anteriormente.

A continuación, vamos a analizar qué tipo de relación tienen estas variables estadísticamente significativas con respecto la variable del nivel de sedación:

- Días de estancia (ρ = -0.29 y un *p*-valor = 0.012): presenta una correlación estadísticamente significativa con el nivel de sedación, de tal forma que los

pacientes ingresados más días tienden a estar sobresedados. Esta información concuerda con los protocolos de sedación, donde se confirma que un uso excesivo de sedante tiende a provocar estancias hospitalarias más largas (Estébanez-Montiel et al., 2008).

- BIS (ρ = 0.03 y un *p*-valor = 0.018): presenta una correlación muy débil, próxima a 0 pero ligeramente positiva. Esto significa que valores altos de BIS (menor profundidad anestésica) se asocian en este caso con infrasedación. Esto concuerda con el significado de dicha variable, como bien se explica en estudios previos (Deogaonkar et al., 2004).

- *RASS* (ρ =-0.32 y *p*-valor = 0.005): se trata de la correlación más fuerte encontrada. Al ser negativa, indica que un *RASS* muy bajo (valores negativos) está asociado a sobre sedación. Esto concuerda con el significa de la propia escala, donde asegura que un nivel adecuado de sedación sería un *RASS* aproximado de -2 (Frade Mera et al., 2009).

- Dosis de aminas (ρ = 0.3 y *p*-valor = 0.008): la correlación positiva indica que los pacientes a los que se es administro aminas tienden a necesitar dosis más elevados de fármacos sedantes.

- MI (ρ = -0.1 y *p*-valor = 0.019): al tratarse de una variable categórica, una correlación negativa indica que los códigos más bajos de MI (shock séptico y patología respiratoria) estarían asociados ligeramente a una sobresedación. Esto puede indicar que ciertas patologías requieran más precisión o atención por parte del personal médico a la hora de administrar las dosis.

- VM (ρ = -0.3 y un *p*-valor = 0.017): al igual que la anterior, también es una variable categórica con correlación de Spearman negativa, por lo que las clases más bajas (tubo endotraqueal en T y tubo endotraqueal con ventilación contrada) se asocian a sobresedación. Esto tiene sentido, pues se trata de métodos de ventilación invasiva, donde se suelen administrar altas dosis de sedantes al tener que intubar al paciente (Estébanez-Montiel et al., 2008).

- Cisatracurio (ρ = -0.16 y *p*-valor = 0.026): la correlación negativa indica que a los pacientes que se le administran dicho fármaco tienen a estar sobresedados. Esta relación tiene sentido pues el cisatracurio es un relajante muscular cuyo uso requiere de una sedación profunda para evitar que el paciente este despierto y paralizado (Accord Healthcare S.L.U, 2023).

Llama la atención que variables como la FC o TAM, no muestran correlaciones significativas con la variable objetivo, cuando si bien se ve más adelante, resultan de interés para el algoritmo. Puede que estos resultados se deban a alteraciones patológicas previas que experimenten los pacientes.

Por otro lado, en el resto de las columnas de la Tabla 9, se incluyen los p-valores que muestra el análisis por pares de grupo de sedación de cada variable. En negrita encontramos subrayados aquellos valores estadísticamente significativos, es decir, inferiores a 0.05.

Estos valores revelaron distintos patrones de diferenciación entre los diferentes grupos. La comparación entre infrasedados y sedación adecuada mostró diferencias significativas para las variables días de ingreso ($p = 0.0117$), RASS ($p = 0.0302$), dosis de aminas ($p = 0.0027$), ventilación mecánica ($p = 0.0109$) y cisatracurio ($p = 0.0093$). estos resultados sugieren que la infrasedación puede estar asociada con una mayor inestabilidad clínica y mayor agitación, reflejándose en el número de días hospitalizados.

La comparación entre infrasedados y sobresedados reveló que las variables días de ingreso, RASS y ventilación mecánicas eran las únicas con diferencias significativas. Esto refuerza la importancia de dichas variables a la hora de realizar la clasificación o distinción de pacientes.

Para la comparación de sobresedados y sedación adecuada, en cambio, la única variable que tiene diferencias significativas entre ambos grupos es el factor de riesgo de la DM. Esto podría abrir alguna hipótesis sobre la relación entre la dosis de sedación y dicha patología. Es posible que los pacientes con DM presenten una farmacocinética alterada o por complicaciones asociadas a la enfermedad se les administre dosis mayores de sedante.

4.3 Evaluación del proceso de selección de características

Como se explicó en apartados anteriores, para asegurarnos de realizar un estudio balanceado y con una correcta distribución de las muestras, realizamos un emparejamiento estratificado de las mismas. Para ello, nos aseguramos de que las variables sexo, edad y niveles de sedación estuvieran repartidas equitativamente en los grupos de entrenamiento y test. En la Figura 7 se muestran los resultados obtenidos.

La correcta división de las muestras según los niveles de sedación es un factor crítico, dado que nuestra base de datos presenta un desbalance natural entre las tres clases: la mayoría de los 100 pacientes pertenecen a la clase de no sedación, mientras que las clases de sedación adecuada y sobresedación están menos representadas, pudiendo influenciar a los modelos al tender a predecir dicha clase mayoritaria.

En cuanto al sexo, la proporción hombre-mujer se mantiene constante en entrenamiento y test. El *boxplot* de la edad muestra una distribución prácticamente idéntica, con una mediana en ambos grupos próxima a 70 años. Además, podemos observar la presencia de *outliers* similares en ambos subconjuntos.

Tras analizar los subconjutnos, se aplicó la técnica de selección de características *FCBF* al conjunto de entrenamiento. Como resultado, se obtuvo la gráfica que se muestra en la Figura 8, donde se ordenan las variables en función del parámetro *SU*, identificando las características que el algoritmo considera más relevantes para el entrenamiento del modelo.

En dicha gráfica, la dosis de aminas obtiene el valor más alto de *SU*. Esto concuerda con los resultados de los *p*-valores y el coeficiente de correlación, convirtiéndose en un factor clave para el control de la sedación. Por otro lado, el peso obtiene un *SU* próximo a 0.14, resultado lógico pues la dosis de fármacos varía en función del peso del paciente, requiriendo dosis más elevadas en casos de sobrepeso.

Las variables VM y MI tienen valores parecido de *SU*, comportamiento consistente con el análisis de correlación previo. Esto indica que constituyen factores clave a tener en cuenta a la hora de administrar los fármacos sedantes.

El BIS, *RASS*, días de estancia y la administración o no de cisatracurio también muestran valores similares de *SU*. Dentro de este marco encontramos también a la TAM, variable que no mostraba relación estadísticamente significativa con los niveles de sedación, pero que puede que sea de interés clínico desde otros aspectos.

A partir de cierto punto, los valores de *SU* comienzan a decaer por debajo de 0.05. En este grupo encontramos a variables como la dosis de diversos fármacos sedante o las escalas de dolor, indicando que su contribución individual e importancia son menores a lo esperado inicialmente.

Esta clasificación de las escalas de dolor resulta inesperada, pues se trata de medidas frecuentemente utilizadas en el ámbito clínico de la UCI. Esta contradicción podría deberse a varios motivos. Por una parte, al desequilibrio de clases del *dataset,* donde la mayoría de los pacientes son sobresedados y en ese caso una evaluación del dolor resulta inútil, pues tendrían alterados por completo la sensación del dolor. Y, por otra parte, debido a la subjetividad de evaluación del dolor, pues podría introducir sesgo en los datos y no representar una escala realista.

Finalmente, las características seleccionadas con *FCBF* fueron: dosis de aminas, peso del paciente, tipo de VM, MI, TAM y sedación con sevoflurano inhalado. Vemos como en este caso se han incorporado las variables como TAM y sevoflurano, variables cuyo *SU* no era tan elevado como otras. Esto demuestra la capacidad de dicho algoritmo para capturar interacciones complejas, pues *FCBF* no solo evalúa la *SU* de cada variable, sino elimina las características relevantes entre sí.

A lo largo de la descripción de los resultados de los modelos bajo estudio, se realizaron pruebas de exactitud respecto el número de características seleccionados. Tras ellas llegamos en todos los casos a la misma conclusión. Un mayor número de características no implica tener mejores resultados en el rendimiento del modelo. De hecho, se puede observar cómo se llegan a mejores resultados tan solo con unas pocas variables, siendo estas variables seleccionadas las de mayor importancia.

Hemos comprobado como al tener un *dataset* pequeño, si entrenamos a los modelos con numerosas características (siendo algunas irrelevantes) estos pueden tender a memorizar patrones, sin llegar a generalizar bien datos buenos y disminuyendo así la *accuracy* de los modelos. Además, reducir el número de características implica un entrenamiento más rápido y, con ello, menor peso computacional.

4.4 Análisis comparativo del rendimiento de los modelos

Para elegir el mejor modelo para nuestro estudio es necesario realizar una comparación, por un lado, del rendimiento global de cada uno; y, por otro lado, del rendimiento por clase de cada uno. Para ello nos fijaremos en los resultados recogidos en las Tablas 17 y 18.

En cuanto a los resultados globales, ambos modelos de *AdaBoostM2* demostraron superioridad total, alcanzando los mejores resultados. En negrita podemos observar los mejores resultados de las métricas de rendimiento corresponden a la etiqueta 2. Sin embargo, si nos centramos en el resultado por clases, los resultados difieren.

La clase 0 (sobresedados) obtiene valores de *accuracy* relativamente altos en todos los modelos, siendo los dos de *AdaBoostM2* los de mayor exactitud. En general, dicho algoritmo, indiferentemente de con que etiqueta fue entrenado, tiene buenos resultados en todas las métricas. Sin embargo, el modelo entrenado con el *target* 1 es ligeramente superior. El *F1-score* de 0.78, su razón de verosimilitud positiva (1.92) y negativa (0.33) y su *AUC* de 0.81 indican una capacidad discriminativa superior, característica crucial en el caso de nuestro estudio.

Para la clase 1, por el contrario, se consiguieron mejores resultados con el modelo *AdaBoostM2* entrenado con la etiqueta 2. En este caso, se obtuvo una exactitud considerablemente elevada (72%). Dicho modelo es capaz de generar un *LR+* de 2.96 y un *AUC* de 0.67, indicando una gran capacidad para confirmar casos de sedación adecuada. Sin embargo, los valores de *Se, Sp* y *F1-score* obtenidos en todos los modelos no son lo suficientemente elevados como para considerar un buen rendimiento.

Para la clase de infrasedados (clase 2), la única opción fiable es el modelo AdaBosstM2 entrenado con la etiqueta 1. Se trata del modelo capaz de detectar el mayor número de VP. A pesar de ello, tan solo obtiene una *Se* de 0.5 y una *AUC* de 0.52, sugiriendo una capacidad discriminativa próxima al azar.

A la hora de elegir el mejor modelo, nos decantamos finalmente por el entrenado con la etiqueta 1. A pesar de tener un rendimiento global ligeramente inferior al entrenado con la etiqueta 2 (*accuracy* del 66% frente al 68% de la etiqueta 2 e índice *Kappa* de 0.33 frente al 0.37 de la etiqueta 2) es el único capaz de obtener buenos resultados en todas las clases. Además, obtuvo valores de *AUC* similares en alguna clase a los resultados proporcionados en otros estudios (Ramaswamy et al., 2022), sin necesidad de BIS e incluyendo a pacientes de diversas patologías.

La exactitud de dicho modelo supera en un 14% a la proporcionada por el personal médico en la clasificación. Esto quiere decir, que este modelo es capaz de predecir el nivel de sedación un 14% mejor que el personal médico.

4.5 Comparación con estudios previos

Actualmente, no se dispone de estudios previos directamente comparables con el presente trabajo, ya que no se ha identificado ninguna investigación que utilice el mismo conjunto de variables predictoras para clasificar el nivel de sedación en tres categorías: infrasedación, sedación adecuada y sobresedación. Además, la mayoría de los estudios existentes se apoyan en herramientas adicionales como el BIS o el registro completo de la señal EEG. Esta falta de comparabilidad directa, lejos de ser una limitación, refuerza el carácter novedoso y el potencial clínico de nuestra propuesta.

Uno de los estudios más similares encontrados es el trabajo de Ramaswamy et al. (Ramaswamy et al., 2022). En él se desarrollan distintos modelos de ML para estimar una clasificación binaria para diferenciar entre el estado de vigilia y el de sedación. Para ello, utilizaron un conjunto de datos agrupados durante la infusión de propofol, sevoflurano,

remifentanilo y dexmedetomidina. Su enfoque se centró en la aplicación de distintos algoritmos de ML no lineales para diseñar un sistema de monitorización del nivel de sedación independiente del fármacos utilizando características cuantitativas derivadas del EEG frontal. Los rendimientos de los diferentes modelos presentaron valores de AUC superiores a 0.8, evidenciando una buena discriminación.

En nuestro estudio, en cambio, logramos una clasificación del nivel de sedación en tres clases sin la necesidad de señales fisiológicas, como el EEG o el BIS. Además, se desarrolló sobre una muestra más heterogénea de pacientes ingresados en UCI, abarcando múltiples patologías y un amplio rango de edades. Esto representa una diferencia importante respecto al estudio de Ramaswamy, el cual se enfoca en la sedación durante operaciones quirúrgicas y en condiciones controladas, sin contemplar los desafíos clínicos derivados de los largos periodos de sedación de los pacientes críticos.

En cuanto a los resultados, el modelo AdaBoostM2 entrenado con la etiqueta 1 alcanzó un AUC de 0.81 para la clasificación de pacientes sobresedados. Se trata de un valor comparable al de los estudios previos, pero obtenido sin necesidad de equipamiento especializado ni monitorización avanzada. Esto sugiere que nuestro enfoque presenta una capacidad discriminativa robusta, con el valor añadido de su aplicabilidad en entornos clínicos con recursos limitados.

4.6 Interpretación de los resultados de Inteligencia Artificial Explicable

Los valores *SHAP* ofrecen una doble ventaja: nos aportan información relevante sobre el comportamiento interno y la interpretabilidad del modelo; y por otra parte aportan nuevos conocimientos que puede enriquecer la práctica clínica.

Saber la importancia que proporciona cada característica en las predicciones del modelo resulta de gran interés a los profesionales sanitario. Estos hallazgos pueden integrarse en su experiencia clínica diaria, ayudando a identificar patrones que quizás no se habían considerado previamente. De esta forma, el estudio de la interpretabilidad del modelo constituye una herramienta clave para mejorar la toma de decisiones médicas.

En la Figura 34 podemos observar de manera detallada como se distribuyen los valores *SHAP* para las tres clases del modelo. De esta forma se consigue comprender como influyen individualmente cada una de las variables en la predicción de cada individuo.

En el primer gráfica, donde se muestran los valores *SHAP* para los pacientes clasificados correctamente como sobresedados, observamos como las variables peso y TAM tiene una influencia clara en la predicción del modelo. La variable peso presenta valores positivos para todos sus pacientes, considerándose de gran relevancia para el modelo a la hora de clasificar a los pacientes dentro de la clase 0. Por otro lado, la variable TAM también sugiere ser de gran relevancia a la hora de clasificar a los pacientes en la mayoría de los casos.

En el segundo gráfico, donde se representan a los pacientes sedados adecuadamente, podemos observar cómo los valores *SHAP* en general son menores para todas las variables en comparación con la gráfica de su izquierda. En este caso, la única variable relevante para la clasificación correcta en la clase 1 es el MI.

En el tercer gráfico llama la atención como la mayoría de las variables presentan valores *SHAP* negativos, lo que podría interpretarse como un patrón clínico caracterizado por un descenso generalizado en los valores de esas variables para predecir esta clase.

La Figura 35, muestra la jerarquía de importancia de las variables del mejor de nuestros modelos, el *AdaBoostM2* con la etiqueta 1. Estos valores nos revelan que el peso del paciente se trata de una característica imprescindible para la clasificación, siendo el predictor más influyente con un valor *SHAP* aproximado de 0.125. Aunque era difícil predecir que el peso sería la característica más relevante en la predicción, los resultados concuerdan con lo esperado, pues la dosis administrada de fármacos sedantes para conseguir concentraciones en sangre adecuadas se realiza en base al peso del paciente.

Seguidamente, la segunda característica más relevante que encontramos es el MI, con un *SHAP* de 0.09. Cada patología o motivo de ingreso va a requerir distintas necesidades de sedación debido a las diversas condiciones fisiopatológicas que experimenten los pacientes. Si esto lo relacionamos con los valores de correlación de Spearman, destacamos como el shock séptico y las patologías respiratorias estarían asociados ligeramente a una sobresedación.

En el tercer puesto encontramos a la variable TAM, seguida de la VM y la dosis de aminas. Será importante controlar y tener en cuenta dichos criterios a la hora del ajuste de los fármacos.

Finalmente, cabe destacar que la variable sedación con sevoflurano tiene un valor *SHAP* próximo a 0. Esto significa que el algoritmo no tiene en cuenta en ningún momento dicha variable para tomar la decisión de la clasificación.

Los resultados obtenidos muestran como el modelo *AdaBoostM2*, entrenado únicamente con cinco variables, ha logrado una precisión superior al criterio de clasificación de los profesionales médicos, quienes toman sus decisiones basándose en todas las variables disponibles, incluyendo el índice BIS. Esto plantea una reflexión importante: el modelo es capaz de identificar patrones clínicos relevantes sin necesidad de recurrir a una monitorización que, aunque tradicionalmente considerada clave, no ha demostrado ser imprescindible para la predicción en este caso. Este hecho representa un nuevo conocimiento con gran potencial para optimizar la toma de decisiones en contextos clínicos donde el acceso a ciertas variables puede estar limitado o donde se busca una mayor eficiencia diagnóstica.

4.7 Limitaciones

Una de las mayores limitaciones a las que se enfrenta el estudio es el reducido número de pacientes de la base de datos. Las *dataset* clínicos se caracterizan por presentar una enorme cantidad de *missing values.* La tarea de encontrar pacientes que cumpliesen los requisitos para pertenecer al estudio (mayores de edad, bajo sedación y con monitorización con BIS) y a la vez que tuvieran información recogida y guardada correctamente en su historial fue compleja y laboriosa.

Por otra parte, de los 100 pacientes recogidos en nuestra base de datos, la gran mayoría están etiquetados como sobresedados, siendo muy reducido el número de pacientes clasificados como infrasedadsos o sedación adecuada. Esto genera un gran desbalance en el *dataset*, dificultando la capacidad del modelo de aprender patrones de las clases

minoritario y tendiendo a generalizar a la clasificación de nuevas muestras como sobresedados, ya que es la clase que más información se tiene.

En cuanto al rendimiento de los modelos, a pesar de haber encontrado un algoritmo con mejores métricas de rendimiento que el acuerdo medico de clasificación, algunos indicadores sugieren margen de mejora en su capacidad predictiva. Un *Kappa* de 0.33 es indicativo de una concordancia aún limitada, sugiriendo que el modelo puede beneficiarse de ajustes adicionales o del uso de datos más representativos. Además, la capacidad discriminativa de dicho algoritmo para clasificar al grupo de infrasedados mostró un *AUC* de 0.52. Si bien este valor es solo ligeramente superior al azar, permite establecer una base sobre la que seguir trabajando para optimizar la clasificación de esta clase en futuras versiones del modelo.

4.8 Líneas futuras

Puesto que el estudio planteado en el presente documento se ha realizado desde la base, existen diversas opciones de mejora en el futuro para poder obtener resultados más fiables. Primeramente, a pesar de haber seleccionado un tamaño muestral correcto en base al análisis de tamaño muestral realizado previamente, se recomienda aumentar el número de pacientes. Por otro lado, en base a superar las limitaciones anteriores, se recomienda balancear el número de instancias por clase, para evitar la generalización a una determinada clase, como se explicó anteriormente.

Una parte importante a la hora de aumentar la base de datos consistiría en la ampliación de la toma de muestra de diversos hospitales (estudio multicéntrico). A pesar de existir protocolos de sedación muy sólidos y fiables, la manera de trabajar de cada hospital puede llevar a tomar diferentes decisiones a la hora de utilizar un fármaco u otro, pudiendo variar los resultados. Además, ampliar el estudio a diversas UCIs, podrían ofrecernos mayor diversidad de características poblacionales.

Otro posible línea de estudio consistiría en la recogida de múltiples evaluaciones del nivel de sedación por parte de diferentes profesionales médicos para cada paciente. Como hemos visto en el Capítulo 3, existen discrepancias en la clasificación clínica de las diferentes instancias del modelo. Por ello, sería conveniente tener el registro de múltiples etiquetas de clasificación de los niveles de sedación, realizadas por distintos profesionales, y poder obtener un consenso final. Para ello, una alternativa sería aplicar métricas como la F1* (Gurdiel et al., 2025), la cual permite evaluar la concordancia entre observadores de manera simétrica, sin necesidad de asumir una etiqueta como "verdadera". Asimismo, otra posibilidad sería la creación de etiquetas compuestas a partir de las anotaciones múltiples. Por ejemplo, una opción sería la creación de una nueva variable objetivo basada en la intersección o unión de las clasificaciones realizadas por los distintos profesionales, considerando únicamente las instancias en las que todos los expertos coinciden en su evaluación.

Adicionalmente, la utilización de otras variables fisiológicas como la saturación de oxígeno, variables de predicción del riesgo de delirium o el registro del EEG, ayudarían a tener un control más preciso del paciente, además que serían características de las que se podría disponer de manera continua en el tiempo (Celis-Rodríguez et al., 2013).

Por último, un punto clave para futuras investigaciones, consistiría en la toma de un registro continuo durante las 24 horas del días de los sujetos bajo estudio, analizando las

mismas variables. De esta forma, podríamos tener un control de la anestesia y sedación de manera continua y efectuar como varían las variables fisiológicas con la subida o la bajada de dosis de sedantes.

Capítulo 5. Conclusiones

En este TFG se analizaron diversas variables sociodemográficas y clínicas de 100 pacientes analgosedados en UCI. En primer lugar, se creó la base de datos, realizándose la correspondiente imputación de datos faltantes. Seguidamente, se llevó a cabo un análisis estadístico, evaluando la asociación entre todas las variables disponibles con la variable objetivo (nivel de sedación). Después, se seleccionaron las características más relevantes mediante el algoritmo *FCBF* y se desarrollaron seis modelos de *ML* para predecir la clasificación. Por último, se aplicaron técnicas de *XAI* para obtener información de las características más relevantes del modelo con mejor rendimiento.

Los resultados revelaron que *AdaboostM2* se consolida como el mejor modelo, con un índice *Kappa* de 0.33 y una *accuracy* del 66% para la etiqueta 1. Asimismo, las características seleccionadas más relevantes fueron, en orden: dosis de aminas, peso del paciente, tipo de VM, MI, TAM y sedación con sevoflurano inhalado. Los valores *SHAP*, mostraron la relevancia del peso a la hora de hacer predicciones, con un valor *SHAP* de 0.35, así como la irrelevancia del factor sevoflurano, con un *SHAP* nulo.

Tras examinar los resultados, a continuación, se describirán las conclusiones obtenidas.

1. El mejor modelo de *ML* es capaz de clasificar según el nivel de sedación en los tres grupos definidos (sobresedación, sedación adecuada e infrasedación) con una exactitud 14% superior a la concordancia entre juicios clínicos, demostrando la viabilidad de la presente propuesta. A pesar de ello, el modelo presenta limitaciones. Si bien es el que mayor *AUC* presenta para la clase infrasedados, su valor es ligeramente superior a 0.5, lo que indica un margen claro de mejora mediante el aumento de la muestra y la incorporación de datos de múltiples centros hospitalarios
2. La selección de características permitió trabajar con 6 variables claves, siendo el BIS un parámetro finalmente no utilizado. De esta manera, se consigue un modelo capaz de predecir la sedación sin la necesidad de la monitorización BIS, reduciendo costes y carga de trabajo. Este hecho hace factible la implantación de este modelo en el entorno clínico de la UCI. Sería ideal que en futuras investigaciones se ampliaran las características bajo estudio y confirmar la capacidad del modelo de clasificación sin la necesidad del BIS.
3. La aplicación de técnicas de interpretabilidad desveló que el peso, el MI y la TAM se consideran los factores más influyentes en la clasificación. Además, se demostró como la variable dosis de sevoflurano resultó insignificante a la hora de hacer predicciones. Esta información resulta de gran valor en la práctica clínica, complementando los conocimientos del personal profesional.

Gracias a la elaboración de este TFG, se ha conseguido demostrar la viabilidad de implementar un modelo de *ML* para automatizar la clasificación de pacientes en base al nivel de sedación. Este modelo no solo supera al juicio clínico, sino que también elimina la necesidad del BIS, reduciendo así costes. Este avance proporcionaría un mayor control de las dosis requeridas para cada paciente, permitiendo reducir los riesgos asociados a una sedación inadecuada y mejorando potencialmente la seguridad del paciente.

Bibliografía

Accord Healthcare S.L.U. (2023). *Ficha técnica: Cisatracurio Accordpharma 2 mg/ml y 5 mg/ml solución inyectable y para perfusión EFG*.

Aguilar García, C. R., Martínez Torres, C., Aguilar García, C. R., & Martínez Torres, C. (2017). La realidad de la Unidad de Cuidados Intensivos. *Medicina crítica (Colegio Mexicano de Medicina Crítica)*, *31*(3).

Ashmore, R., Calinescu, R., & Paterson, C. (2022). Assuring the Machine Learning Lifecycle: Desiderata, Methods, and Challenges. En *ACM Computing Surveys* (Vol. 54, Número 5). https://doi.org/10.1145/3453444

Aznar-Oroval, E., Mancheño-Alvaro, A., García-Lozano, T., & Sánchez-Yepes, M. (2013). Razón de verosimilitud y nomograma de Fagan: 2 instrumentos básicos para un uso racional de las pruebas del laboratorio clínico. *Revista de Calidad Asistencial*, *28*(6), 390-391. https://doi.org/10.1016/j.cali.2013.04.002

Barr, J., Fraser, G. L., Puntillo, K., Ely, E. W., Gélinas, C., Dasta, J. F., Davidson, J. E., Devlin, J. W., Kress, J. P., Joffe, A. M., Coursin, D. B., Herr, D. L., Tung, A., Robinson, B. R. H., Fontaine, D. K., Ramsay, M. A., Riker, R. R., Sessler, C. N., Pun, B., … Jaeschke, R. (2013). Clinical practice guidelines for the management of pain, agitation, and delirium in adult patients in the intensive care unit. *Critical Care Medicine*, *41*(1). https://doi.org/10.1097/CCM.0b013e3182783b72

Bels, D., Bousbiat, I., Perriens, E., Blackman, S., & Honoré, P. (2023). Sedation for adult ICU patients: A narrative review including a retrospective study of our own data. En *Saudi Journal of Anaesthesia* (Vol. 17, Número 2). https://doi.org/10.4103/sja.sja_905_22

Bodria, F., Giannotti, F., Guidotti, R., Naretto, F., Pedreschi, D., & Rinzivillo, S. (2023). Benchmarking and survey of explanation methods for black box models. *Data Mining and Knowledge Discovery*, *37*(5), 1719-1778. https://doi.org/10.1007/S10618-023-00933-9/TABLES/9

Brochard, L. (2008). Sedation in the intensive-care unit: good and bad? En *The Lancet* (Vol. 371, Número 9607). https://doi.org/10.1016/S0140-6736(08)60082-3

Caballero López, J., Manuela García Sánchez, M., & Carola Giménez-Esparza Vich, C. (2020). *Protocolo de sedación inhalatoria en UCI. Recomendaciones del grupo de trabajo de sedación, analgesia y delirium de la Sociedad Española de Medicina Intensiva, Crítica y Unidades Coronarias (SEMICYUC).* www.semicyuc.org

Cardoso-Ortiz, J., López-Luna, M. A., Lor, K. B., Cuevas-Flores, M. R., Flores de la Torre, J. A., Covarrubias, S. A., Cardoso-Ortiz, J., López-Luna, M. A., Lor, K. B., Cuevas-Flores, M. R., Flores de la Torre, J. A., & Covarrubias, S. A. (2020). Opioids: Pharmacology and Epidemiology. *Revista bio ciencias*, *7*.

Celis-Rodríguez, E., Birchenall, C., de la Cal, M. Á., Castorena Arellano, G., Hernández, A., Ceraso, D., Díaz Cortés, J. C., Dueñas Castell, C., Jimenez, E. J., Meza, J. C., Muñoz Martínez, T., Sosa García, J. O., Pacheco Tovar, C., Pálizas, F., Pardo Oviedo,

J. M., Pinilla, D.-I., Raffán-Sanabria, F., Raimondi, N., Righy Shinotsuka, C., … Rubiano, S. (2013). Guía de práctica clínica basada en la evidencia para el manejo de la sedoanalgesia en el paciente adulto críticamente enfermo. *Medicina Intensiva*, *37*(8), 519-574. https://doi.org/10.1016/j.medin.2013.04.001

Cerda, J., & Cifuentes, L. (2012). Uso de curvas ROC en investigación clínica: Aspectos teórico-prácticos. *Revista chilena de infectología*, *29*, 138-141.

Chamorro, C., Martínez-Melgar, J. L., Barrientos, R., & Grupo de trabajo de analgesia y sedación de la SEMICYUC. (2008). Monitorización de la sedación. *Med Intensiva*, *32*, 45-52.

Chen, J. H., & Asch, S. M. (2017). Machine Learning and Prediction in Medicine — Beyond the Peak of Inflated Expectations. *The New England journal of medicine*, *376*(26), 2507. https://doi.org/10.1056/NEJMP1702071

Chengsheng, T., Huacheng, L., & Bing, X. (2017). AdaBoost typical Algorithm and its application research. *MATEC Web of Conferences*, *139*, 00222. https://doi.org/10.1051/matecconf/201713900222

Chulay, M. (2004). Sedation assessment: Easier said than done! En *Critical Care Nursing Clinics of North America* (Vol. 16, Número 3 SPEC. ISS.). https://doi.org/10.1016/j.ccell.2004.04.006

Cooke, S. E., Dasta, J., Fish, D., Hassan, E., Horst, H. M., Kelly, K., Kaiser, K., Jackson, C. E., Rudis, M., Schoenberger, C., Schoonove, L., Takaniski, G., Teres, D., & Thompson, K. (2002). Clinical practice guidelines for the sustained use of sedatives and analgesics in the critically ill adult. En *American Journal of Health-System Pharmacy* (Vol. 59, Número 2). https://doi.org/10.1093/ajhp/59.2.150

Daniel Jurafsky, & James H. Martin. (2025). Speech and language processing. En *Speech and Language Processing*.

de la Hoz Manotas, A. K., Martínez-Palacio, U. J., & Mendoza-Palechor, F. E. (2013). Técnicas de ML en medicina cardiovascular. *Memorias*, *11*(20), 41-46.

Deogaonkar, A., Gupta, R., DeGeorgia, M., Sabharwal, V., Gopakumaran, B., Schubert, A., & Provencio, J. J. (2004). Bispectral Index monitoring correlates with sedation scales in brain-injured patients. *Critical Care Medicine*, *32*(12). https://doi.org/10.1097/01.CCM.0000147442.14921.A5

Eckhardt, C. M., Madjarova, S. J., Williams, R. J., Ollivier, M., Karlsson, J., Pareek, A., & Nwachukwu, B. U. (2023). Unsupervised machine learning methods and emerging applications in healthcare. En *Knee Surgery, Sports Traumatology, Arthroscopy* (Vol. 31, Número 2). https://doi.org/10.1007/s00167-022-07233-7

Ekundayo, T. (2020). *Evaluation of Machine Learning Algorithms for Regression and Classification Problems*. https://doi.org/10.13140/RG.2.2.10368.05120

Estébanez-Montiel, M. B., Alonso-Fernández, M. Á., Sandiumenge, A., & Jiménez-Martín, M. J. (2008). Sedación prolongada en Unidades de Cuidados Intensivos. En *Medicina Intensiva* (Vol. 32, Número SUPPL. 1).

Frade Mera, M. J., Guirao Moya, A., Esteban Sánchez, M. E., Rivera Álvarez, J., Cruz Ramos, A. M., Bretones Chorro, B., Viñas Sánchez, S., Jacue Izquierdo, S., & Montane López, M. (2009). Análisis de 4 escalas de valoración de la sedación en el paciente crítico. *Enfermeria Intensiva*, *20*(3). https://doi.org/10.1016/S1130-2399(09)72588-X

Gil-Castillejos, D., Palomanes-Espadalé, M. L., Rosich-Andreu, S., Vallés-Fructuoso, O., & Plans-Galvan, O. (2025). Uso seguro de la sedación inhalada en pacientes críticos con ventilación mecánica invasiva. *Enfermería Intensiva*, *36*(1). https://doi.org/10.1016/j.enfi.2024.04.003

González Rubio, T., Rodríguez Aldana, Y., Drullet Ferrer, J. L., Marañon Reyes, E. J., & Montoya Pedrón, A. (2019). Monitorización automática de estados de sedación en señales electroencefalográficas. *Revista Cubana de Informática Médica*, *11*, 18-32.

Guerrero Gutiérrez Manuel, Pérez Nieto Orlando, Escarraman Martínez Diego, Ojeda Niño antonio, Zamarrón López Eder, Olivares Reséndiz ricardo, Días Martínez Manuel, Deloya Tomás Ernesto, Sánchez Días Jesús, Silva Llorente Maikel, Chora Pérez Karla, Mosqueda Aguilera Laura, Carbajo Martínez Susana, Torres Prado Dora, & Ferrando Carlos. (2023). Analgesia multimodal en el paciente crítico. *Revista Chilena de Anestesia*, *52*(2), 177-192. https://doi.org/10.25237/revchilanestv5223121124

Gurdiel, E., Vaquerizo-Villar, F., Gomez-Pilar, J., Gutierrez-Tobal, G. C., Campo, F. d., & Hornero, R. (2025). Beyond the Ground Truth, XGBoost Model Applied to Sleep Spindle Event Detection. *IEEE Journal of Biomedical and Health Informatics*, *29*(7), 4873-4883. https://doi.org/10.1109/JBHI.2025.3544966

Handelman, G. S., Kok, H. K., Chandra, R. V, Razavi, A. H., Lee, M. J., & Asadi, H. (2018). eDoctor: machine learning and the future of medicine. *Journal of Internal Medicine*, *284*(6), 603-619. https://doi.org/https://doi.org/10.1111/joim.12822

Hernández, A., Rosales, G., Santiago, H., & Lee, M. (2022). Métricas de rendimiento para evaluar el aprendizaje automático en la clasificación de imágenes petroleras utilizando redes neuronales convolucionales. *Ciencia Latina Revista Científica Multidisciplinar*, *6*, 4624-4637. https://doi.org/10.37811/cl_rcm.v6i5.3420

Htun, H. H., Biehl, M., & Petkov, N. (2023). Survey of feature selection and extraction techniques for stock market prediction. En *Financial Innovation* (Vol. 9, Número 1). https://doi.org/10.1186/s40854-022-00441-7

Hurtado Oliver, B., Giménes-Esparza, C., Alcántara Carmona, S., & Rodríguez Ruiz, S. (2022). *Algoritmos de actuación en analgo-sedación y delirium*. https://semicyuc.org/wp-content/uploads/2022/11/GTSAD-ALGORITMOS-DE-ACTUACION-EN-ANALGOSEDACION-Y-DELIRIUM-SEMICYUC-DELIRIUM.pdf

International association for the study of pain. (2020). *IASP Revises Its Definition of Pain for the First Time Since 1979*. International association for the study of pain.

Jackson, D. L., Proudfoot, C. W., Cann, K. F., & Walsh, T. S. (2009). The incidence of sub-optimal sedation in the ICU: A systematic review. *Critical Care*, *13*(6). https://doi.org/10.1186/cc8212

Jain, S., & Iverson, L. M. (2025). Glasgow Coma Scale. En *StatPearls*. StatPearls Publishing. https://www.ncbi.nlm.nih.gov/books/NBK513298/

Kubota, Y., Nakamoto, H., Egawa, S., & Kawamata, T. (2018). Continuous EEG monitoring in ICU. En *Journal of Intensive Care* (Vol. 6, Número 1). https://doi.org/10.1186/s40560-018-0310-z

Lanzagorta-Ortega, D., Carrillo-Pérez, D. L., & Carrillo-Esper, R. (2023). Inteligencia artificial en medicina: presente y futuro. *Gaceta Médica de México*, *158*(91). https://doi.org/10.24875/gmm.m22000688

Lazarevic, A., & Obradovic, Z. (2002). *Effective pruning of Neural Network classifier ensembles*.

López Jiménez, J., & Giménez Prats, M. J. (2004). Sedación en el paciente geriátrico. *Medicina Oral, Patología Oral y Cirugía Bucal (Ed. impresa)*, *9*(1), 45-55. https://scielo.isciii.es/scielo.php?script=sci_arttext&pid=S1698-44472004000100007&lng=es&nrm=iso&tlng=es

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, *2017-December*.

Machova, K., František, B., & Bednár, P. (2006). A Bagging Method using Decision Trees in the Role of Base Classifiers. *Acta Polytechnica Hungarica*, *3*.

Mahesh, B. (2020). Machine Learning Algorithms - A Review. *International Journal of Science and Research (IJSR)*, *9*(1). https://doi.org/10.21275/art20203995

Manterola, C., Grande, L., Otzen, T., García, N., Salazar, P., & Quiroz, G. (2018). Confiabilidad, precisión o reproducibilidad de las mediciones. Métodos de valoración, utilidad y aplicaciones en la práctica clínica. *Revista chilena de infectología*, *35*, 680-688.

Martin, M. G. I. M. X. U., & Idoate, M. (2015). Aprendizaje de distancias basadas en disimilitudes para el algoritmo de clasificación kNN. *Universidad pública de Navarra, Pamplona*.

Mattia, C., Savoia, G., Paoletti, F., Piazza, O., Albanese, D., Amantea, B., Ambrosio, F., Belfiore, B., Berti, M., & Bertini, L. (2006). SIAARTI Recommendations for analgo-sedation in intensive care unit LINEE GUIDA SIAARTI. *Minerva Anestesiologica*, *72*, 769-805. www.pnlg.it

Mayr, A., Binder, H., Gefeller, O., & Schmid, M. (2014). The evolution of boosting algorithms: From machine learning to statistical modelling. *Methods of Information in Medicine*, *53*(6). https://doi.org/10.3414/ME13-01-0122

Mienye, D., & Jere, N. (2024). A Survey of Decision Trees: Concepts, Algorithms, and Applications. *IEEE Access*, *PP*, 1. https://doi.org/10.1109/ACCESS.2024.3416838

Minh, D., Wang, H. X., Li, Y. F., & Nguyen, T. N. (2021). Explainable artificial intelligence: a comprehensive review. *Artificial Intelligence Review 2021 55:5*, *55*(5), 3503-3568. https://doi.org/10.1007/S10462-021-10088-Y

Molina, M. (2024). An intruder from another world: F1-score. *Revista Electrónica AnestesiaR*, *16*. https://doi.org/10.30445/rear.v16i4.1258

Monostori, L. (2019). Artificial Intelligence. En S. Chatti, L. Laperrière, G. Reinhart, & T. Tolio (Eds.), *CIRP Encyclopedia of Production Engineering* (pp. 73-76). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-53120-4_16703

Mukaka, M. M. (2012). A guide to appropriate use of Correlation coefficient in medical research. *Malawi Medical Journal : The Journal of Medical Association of Malawi*, *24*(3), 69. https://pmc.ncbi.nlm.nih.gov/articles/PMC3576830/

Murel Ph.D., J., & Kavlakoglu, E. (2024, enero 20). *What is a feature engineering?* . IBM. https://www.ibm.com/think/topics/feature-engineering

Naeem, S., Ali, A., Anam, S., & Ahmed, M. M. (2023). An Unsupervised Machine Learning Algorithms: Comprehensive Review. *International Journal of Computing and Digital Systems*, *13*(1). https://doi.org/10.12785/ijcds/130172

Nasteski, V. (2017). An overview of the supervised machine learning methods. *HORIZONS.B*, *4*. https://doi.org/10.20544/horizons.b.04.1.17.p05

Navarro, P. H. (2023). Critical patient pain management. *NPunto*, *6*, 99-125. https://www.npunto.es/revista/66/manejo-del-dolor-en-el-paciente-critico

Navarro Suay, R., Gutiérrez Ortega, C., & Gilsanz Rodríguez, F. (2016). Empleo del índice biespectral para monitorización de la hipnosis en sedación durante anestesia regional, experiencia en tres pacientes militares. *Sanid. mil*, *72*(3).

Olmos, M., Varela, D., & Klein, F. (2019). Enfoque actual de la analgesia, sedación y el delirium en cuidados críticos. *Revista Médica Clínica Las Condes*, *30*(2). https://doi.org/10.1016/j.rmclc.2019.03.002

Olvera-Martínez, R., Loredo-García, N. G., Mutis-Ospino, B., Olvera-Martínez, R., Loredo-García, N. G., & Mutis-Ospino, B. (2021). Sedación en paciente con superobesidad: reporte de caso. *Cirugía y cirujanos*, *89*, 49-53. https://doi.org/10.24875/CIRU.20001100

Oti, E. U., Olusola, M. O., Eze, F. C., & Enogwe, S. U. (2021). Comprehensive Review of K-Means Clustering Algorithms. *International Journal of Advances in Scientific Research and Engineering*, *07*(08). https://doi.org/10.31695/ijasre.2021.34050

Paixão, G. M. de M., Santos, B. C., Araujo, R. M. de, Ribeiro, M. H., Moraes, J. L. de, & Ribeiro, A. L. (2022). Machine Learning na Medicina: Revisão e Aplicabilidade. *Arquivos Brasileiros de Cardiologia*, *118*(1), 95-102. https://doi.org/10.36660/abc.20200596

Pardo, C., Muñoz, T., & Chamorro Jambrina, C. (2008). Monitorización del dolor. Recomendaciones del grupo de trabajo de analgesia y sedación de la SEMICYUC.

En *Medicina Intensiva* (Vol. 32, Número SUPPL. 1). https://doi.org/10.1016/s0210-5691(06)74552-1

Peng, J., Lee, K., & Ingersoll, G. (2002). An Introduction to Logistic Regression analysis and reporting. *Journal of Educational Research - J EDUC RES*, *96*, 3-14. https://doi.org/10.1080/00220670209598786

Pérez-Bárcena, J., Homar, J., Abadal, J. M., Ibáñez, J., Barceló, B., Molina, F. J., de la Peña, A., & Sahuquillo, J. (2005). Comparación de la eficacia del pentobarbital y el tiopental en el control de la hipertensión intracraneal refractaria. Resultados preliminares en una serie de 20 pacientes. *Neurocirugía*, *16*(1). https://doi.org/10.1016/s1130-1473(05)70426-7

Qiu, J., Wu, Q., Ding, G., Xu, Y., & Feng, S. (2016). A survey of machine learning for big data processing. En *Eurasip Journal on Advances in Signal Processing* (Vol. 2016, Número 1). https://doi.org/10.1186/s13634-016-0355-x

Ramaswamy, S. M., Kuizenga, M. H., Weerink, M. A. S., Vereecke, H. E. M., Struys, M. M. R. F., & Belur Nagaraj, S. (2022). Frontal electroencephalogram based drug, sex, and age independent sedation level prediction using non-linear machine learning algorithms. *Journal of Clinical Monitoring and Computing*, *36*(1). https://doi.org/10.1007/s10877-020-00627-3

Ramírez, P. E. G., Molina, L. C. V., Araujo, E. Z., Morales, C. M. L., Molina, R. C., & Magro, P. M. H. (2018). Prevalence of pain in patients hospitalized at the Metabolic Intensive Care Unit with endotracheal intubation and sedation, measured with COMFORT scale. *Revista de la Sociedad Espanola del Dolor*, *25*(1), 7-12. https://doi.org/10.20986/resed.2017.3581/2017

Raschka, Sebastian., & Mirjalili, Vahid. (2019). *Python Machine Learning Machine Learning and Deep Learning with Python, Scikit-Learn, and TensorFlow 2* (3.ª ed.). http://lib.ugent.be/catalog/ebk01:4100000010011030

Ray, S. (2019). A quick review of Machine Learning algorithms. *Proceedings of the International Conference on Machine Learning, Big Data, Cloud and Parallel Computing: Trends, Prespectives and Prospects, COMITCon 2019*. https://doi.org/10.1109/COMITCon.2019.8862451

Reade, M. C., & Finfer, S. (2014). Sedation and Delirium in the Intensive Care Unit. *New England Journal of Medicine*, *370*(5). https://doi.org/10.1056/nejmra1208705

Rendón-Macías, M.-E., Villasís-Keever, M., & Miranda-Novales, M. (2016). Estadística descriptiva. *Revista Alergia México*, *63*, 397. https://doi.org/10.29262/ram.v63i4.230

Rubiños, C., & Godoy, D. A. (2020). Monitorización electroencefalográfica en el paciente crítico: ¿qué información útil puede aportar? *Medicina Intensiva*, *44*(5), 301-309. https://doi.org/10.1016/j.medin.2019.03.012

Sá, J., Almeida, A., Pereira da Rocha, B., Mota, M., De Souza, J. R., & Dentel, L. (2016). *Lightning forecast using data mining techniques on hourly evolution of the convective available potential energy*. https://doi.org/10.21528/CBIC2011-27.1

Samek, W., & Müller, K. R. (2019). Towards Explainable Artificial Intelligence. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11700 LNCS*, 5-22. https://doi.org/10.1007/978-3-030-28954-6_1

Sánchez Mangas, A. (2012). *Análisis de componentes principales : versiones dispersas y robustas al ruido impulsivo*. https://hdl.handle.net/10016/15618

Sapon, M. A., Ismail, K., Zainudin, S., & Ping, C. S. (2011). Diabetes Prediction with Supervised Learning Algorithms of Artificial Neural Network. *IPCSIT vol. 9 (2011)*.

Schweickert, W. D., & Kress, J. P. (2008). Strategies to optimize analgesia and sedation. En *Critical Care* (Vol. 12, Número SUPPL. 3). https://doi.org/10.1186/cc6151

Sessler, C. N., Gosnell, M. S., Grap, M. J., Brophy, G. M., O'Neal, P. V., Keane, K. A., Tesoro, E. P., & Elswick, R. K. (2002). The Richmond Agitation-Sedation Scale: Validity and reliability in adult intensive care unit patients. *American Journal of Respiratory and Critical Care Medicine*, *166*(10). https://doi.org/10.1164/rccm.2107138

Solutions, M. (2023). *Explainable artificial intelligence (XAI) Desafíos en la interpretabilidad de los modelos*. www.managementsolutions.com

Sufentanilo. (2015). *Panorama Actual del Medicamento*, *39*, 295-299.

Tharwat, A., Gaber, T., Ibrahim, A., & Hassanien, A. E. (2017). Linear discriminant analysis: A detailed tutorial. *Ai Communications*, *30*, 169-190. https://doi.org/10.3233/AIC-170729

Vizcaíno-Salazar, G. J. (2017). Importancia del cálculo de la sensibilidad, la especificidad y otros parámetros estadísticos en el uso de las pruebas de diagnóstico clínico y de laboratorio. *Medicina y Laboratorio*, *23*(7-8). https://doi.org/10.36384/01232576.34

Wall, D. P., Kosmicki, J., DeLuca, T. F., Harstad, E., & Fusaro, V. A. (2012). Use of machine learning to shorten observation-based screening and diagnosis of autism. *Translational Psychiatry*, *2*(4), e100-e100. https://doi.org/10.1038/tp.2012.10

Yu, L., & Liu, H. (2003). Feature celection for high-dimensional data: a Fast Correlation-Based Filter solution. En *Proceedings, Twentieth International Conference on Machine Learning* (Vol. 2).

Zhang, S. (2022). Challenges in KNN Classification. *IEEE Transactions on Knowledge and Data Engineering*, *34*(10). https://doi.org/10.1109/TKDE.2021.3049250

Apéndices

A. Código de MATLAB

1. “preprocesado”

```
%% ------------------------------------------------------------------------
%% --------CARGA DE LA BASE DE DATOS ORIGINAL EN FORMATO EXCEL ------------
%% ------------------------------------------------------------------------

% Se lee la tabla de datos con valores ausentes (NaN) desde un archivo .xlsx
MD = readtable('DATOSconNAN.xlsx', 'VariableNamingRule', 'preserve');

%% ------------------------------------------------------------------------
%% -----IMPUTACIÓN Y CODIFICACIÓN DE VARIABLES CATEGÓRICAS Y NUMÉRICAS-----
%% ------------------------------------------------------------------------


% Codificación de la variable categórica SEXO
sexo = MD.SEXO;
sexo_codificado = zeros(height(MD),1);
for i = 1:height(MD)
    if strcmpi(sexo{i}, 'mujer')
        sexo_codificado(i) = 1;
    elseif strcmpi(sexo{i}, 'hombre')
        sexo_codificado(i) = 2;
    else
        sexo_codificado(i) = NaN; % valores inesperados
    end
end
MD.SEXO = sexo_codificado;

% Codificación de la variable MI (Motivo de Ingreso)
MI = MD.MI;
MI_codificado = zeros(height(MD),1);
for i = 1:height(MD)
    switch lower(MI{i})
        case 'shock séptico'
            MI_codificado(i) = 1;
        case 'patología respiratoria'
            MI_codificado(i) = 2;
        case 'acv'
            MI_codificado(i) = 3;
        case 'shock epiléptico'
            MI_codificado(i) = 4;
        case 'patología cardiovascular'
            MI_codificado(i) = 5;
        case 'tce'
            MI_codificado(i) = 6;
        case 'neurocrítico'
            MI_codificado(i) = 7;
        otherwise
            MI_codificado(i) = NaN;
    end
end
MD.MI = MI_codificado;
```

```
% Imputación de valores faltantes en RASS mediante la mediana
RASS = MD.RASS;
RASS(isnan(RASS)) = median(RASS, 'omitnan');
MD.RASS = RASS;

% Imputación de valores faltantes en GLASGOW mediante CART + moda
GLASGOW = MD.GLASGOW;
imputados = cart_imputation(MD, GLASGOW, 20,
length(GLASGOW(~isnan(GLASGOW))));
MD.GLASGOW(isnan(GLASGOW)) = mode(imputados, 2);

% Codificación de la variable VM (Ventilación Mecánica)
VM = MD.VM;
VM_codificado = zeros(height(MD),1);
for i = 1:height(MD)
    switch lower(VM{i})
        case 'tubo endotraqueal  en t'
            VM_codificado(i) = 1;
        case 'tubo endotraqueal. ventilacion controlada'
            VM_codificado(i) = 2;
        case 'tubo endotraqueal. ventilacion asistida/controlada'
            VM_codificado(i) = 3;
        case 'traqueostomia. ventilacion asistida/controlada'
            VM_codificado(i) = 4;
        case 'cpap'
            VM_codificado(i) = 5;
        case 'bipap'
            VM_codificado(i) = 6;
        otherwise
            VM_codificado(i) = NaN;
    end
end
MD.VM = VM_codificado;

% Imputación de FRECUENCIA RESPIRATORIA mediante la mediana
FR = MD.('FRECUENCIA RESPIRATORIA');
FR(isnan(FR)) = median(FR, 'omitnan');
MD.('FRECUENCIA RESPIRATORIA') = FR;

% Imputación de ESCID mediante CART + moda
ESCID = MD.ESCID;
imputados = cart_imputation(MD, ESCID, 20, length(ESCID(~isnan(ESCID))));
MD.ESCID(isnan(ESCID)) = mode(imputados, 2);

% Codificación de CISATRACURIO (uso de bloqueador neuromuscular)
CISATRACURIO = MD.CISATRACURIO;
CISATRACURIO_codificado = zeros(height(MD),1);
for i = 1:height(MD)
    if strcmpi(CISATRACURIO{i}, 'sí')
        CISATRACURIO_codificado(i) = 1;
    elseif strcmpi(CISATRACURIO{i}, 'no')
        CISATRACURIO_codificado(i) = 2;
    else
        CISATRACURIO_codificado(i) = NaN;
    end
end
MD.CISATRACURIO = CISATRACURIO_codificado;

% Imputación de SOFA mediante el método CART
```

```
SOFA = MD.SOFA;
imputados = cart_imputation(MD, SOFA, 20, length(SOFA(~isnan(SOFA))));
MD.SOFA(isnan(SOFA)) = mode(imputados, 2);

%% ------------------------------------------------------------------------
%% ------ GUARDADO DEL DATASET LIMPIO PARA ANÁLISIS POSTERIORES -----------
%% ------------------------------------------------------------------------

% Guardado en formato MATLAB
save('MD.mat', 'MD');

% Exportación a archivo Excel
writetable(MD, 'DATOSsinNAN.xlsx');


%% ------------------------------------------------------------------------
%% --SEGMENTACIÓN DEL DATASET SEGÚN NIVEL DE SEDACIÓN (VARIABLE DE SALIDA)--
%% ------------------------------------------------------------------------

% Se asume que la última columna representa la variable de salida
valoracion_1 = MD{:, end};
grupo_labels = {'SobreSedacion', 'SedacionAdecuada', 'InfraSedacion'}; %
Clases: 0,1,2

% Generación de archivos Excel para cada grupo
for grupo = 0:2
    filas_grupo = MD(valoracion_1 == grupo, :);
    nombre_archivo = sprintf('%s.xlsx', grupo_labels{grupo + 1});
    writetable(filas_grupo, nombre_archivo);
end







%Se sigue la metodología mostrada en
    %L.L. Doove, S. Van Buuren, E. Dusseldorp,
    %Recursive partitioning for missing data imputation in the presence of
interaction effects,
    %Computational Statistics & Data Analysis, Volume 72,2014,Pages 92-
104,ISSN 0167-9473,
    % https://doi.org/10.1016/j.csda.2013.10.025.


function [matriz_imputacion, final_imputacion] = cart_imputation(data,
target, m, n, varargin)
% CART_IMPUTATION - Imputación múltiple mediante árboles de decisión (CART)
%
% Entradas:
%   data      : tabla de datos completa (sin NaNs en predictores)
%   target    : vector columna con valores faltantes (preferiblemente
ordinales)
%   m         : número de imputaciones (iteraciones MICE)
%   n         : tamaño de la muestra bootstrap por imputación
%   varargin  : parámetros opcionales para el ajuste del árbol (key-value
pairs)
%
```

```
% Salidas:
%   matriz_imputacion : matriz de imputaciones (filas: observaciones con NaN,
columnas: iteraciones)
%   final_imputacion  : imputación de la última iteración (se puede usar como
valor final imputado)

    % Identificar los índices de valores faltantes y observados
    NaN_index = find(isnan(target));
    idx_full = find(~isnan(target));

    % Separar datos completos y faltantes
    X_obs = data(idx_full, :);         % Datos completos
    y_obs = target(idx_full);          % Target observado
    X_miss = data(NaN_index, :);       % Datos con valores faltantes en
target

    % Inicializar matriz de imputaciones
    matriz_imputacion = NaN(length(NaN_index), m);

    % Crear input parser para parámetros opcionales
    p = inputParser;
    addParameter(p, 'MinLeafSize', 5);         % Tamaño mínimo de hoja
    addParameter(p, 'MaxNumSplits', 20);       % Máximo número de divisiones
    addParameter(p, 'SplitCriterion', 'gdi');  % Criterio de división ('gdi',
'deviance', etc.)
    addParameter(p, 'Surrogate', 'off');       % Uso de variables sustitutas
    addParameter(p, 'MaxNumCategories', 10);   % Categorías máximas para
predictores
    addParameter(p, 'Prune', 'off');           % Poda del árbol (on/off)
    parse(p, varargin{:});
    params = p.Results;

    % Iteraciones de imputación múltiple
    for i = 1:m
        % Bootstrap de los datos observados
        idx_bootstrap = randsample(length(y_obs), n, true);
        X_obs_boot = X_obs(idx_bootstrap, :);
        y_obs_boot = y_obs(idx_bootstrap);

        % Entrenar árbol de clasificación
        tree = fitctree(X_obs_boot, y_obs_boot, ...
            'MinLeafSize', params.MinLeafSize, ...
            'MaxNumSplits', params.MaxNumSplits, ...
            'SplitCriterion', params.SplitCriterion, ...
            'Surrogate', params.Surrogate, ...
            'MaxNumCategories', params.MaxNumCategories);

        % Aplicar poda si se especifica
        if strcmp(params.Prune, 'on')
            tree = prune(tree, 'Level', 1);
        end

        % Predecir nodos terminales para los casos con valores perdidos
        [~, node] = predict(tree, X_miss);

        % Imputar valores según probabilidades por nodo terminal
        predicted_values = zeros(length(NaN_index), 1);
        class_labels = tree.ClassNames;
```

```
        for j = 1:length(NaN_index)
            % Selección aleatoria ponderada según distribución en el nodo
            predicted_values(j) = randsample(class_labels', 1, true, node(j,
:));
        end

        % Guardar imputación de la iteración actual
        matriz_imputacion(:, i) = predicted_values;
    end

    % Resultado final: última imputación realizada
    final_imputacion = predicted_values;
end
```

2. “analisis_estadistico”

```
%-------------------------------------------------------------------------
% -----CÁLCULO DE ESTADÍSTICOS NECESARIOS PARA EL ESTUDIO DE LOS DATOS-----
%-------------------------------------------------------------------------

%% ---------------------- CARGA DE DATOS ---------------------------------
% Se cargan los datos generales y los subconjuntos según el tipo de sedación

MD = readtable('DATOSsinNAN.xlsx','VariableNamingRule', 'preserve');
% Matriz general de datos (sin valores ausentes)
MD_infra = readtable('InfraSedacion.xlsx','VariableNamingRule', 'preserve');
% Subconjunto: Infrasedación
MD_sobre = readtable('SobreSedacion.xlsx','VariableNamingRule', 'preserve');
% Subconjunto: Sobresedación
MD_adec = readtable('SedacionAdecuada.xlsx','VariableNamingRule',
'preserve');    % Subconjunto: Sedación adecuada

%% --------------------DEFINICIÓN DE VARIABLES ----------------------------
% Se establecen las variables continuas y categóricas a analizar

nombres_continuas = {'EDAD', 'PESO', 'DÍAS ESTANCIA', 'BIS', 'RASS', ...
    'GLASGOW','TAM', 'FRECUENCIA CARDÍACA', 'FRECUENCIA RESPIRATORIA', ...
    'ESCID', 'APACHE II', 'AMINAS DOSIS', ...
    'SEDACIÓN PROPOFOL (mg/kg/h)', 'SEDACIÓN MIDAZOLAM (mg/h)', ...
    'SEDACIÓN DEXMEDETOMIDINA (mcg/kg/h)', 'ANALGESIA FENTANILO (mcg/kg/h)',
...
    'SEDACIÓN SEVOFLUORANO INHALADO (50 ml/50 ml)', 'SOFA'};

nombres_categoricas = {'SEXO', 'DISLIPEMIA', 'EPOC', 'CÁNCER',
'CARDIOVASCULAR', ...
                       'PATOLOGÍA RENAL', 'DM', 'HTA', 'MI', 'VM',
'CISATRACURIO'};


%% ------------------VARIABLE OBJETIVO ----------------------------------
% Se extrae la variable objetivo que clasifica el tipo de sedación

valoracion_1 = MD.('VALORACIÓN ANALGOSEDACIÓN 2');
%% ANÁLISIS DE CORRELACIONES
% Se calculan los coeficientes de correlación de Spearman entre las variables
% y la variable objetivo
```

```
rho_table = calcularSpearman(MD, "VALORACIÓN ANALGOSEDACIÓN 2");

%% -----------------ESTADÍSTICAS DESCRIPTIVAS -----------------------------
% Se calculan las estadísticas descriptivas de las variables continuas
(mediana e IQR)
% y de las categóricas (frecuencia absoluta y relativa) para cada conjunto

tabla_frec_MD = frecuencias(MD, nombres_categoricas);
tabla_frec_infra = frecuencias(MD_infra, nombres_categoricas);
tabla_frec_sobre = frecuencias(MD_sobre, nombres_categoricas);
tabla_frec_adec = frecuencias(MD_adec, nombres_categoricas);

med_iqr_MD = medianas_iqr(MD, nombres_continuas);
med_iqr_MD_infra = medianas_iqr(MD_infra, nombres_continuas);
med_iqr_MD_sobre = medianas_iqr(MD_sobre, nombres_continuas);
med_iqr_MD_adec = medianas_iqr(MD_adec, nombres_continuas);

%% -------------------NORMALIDAD Y HOMOCEDASTICIDAD ------------------------
% Se identifica qué variables presentan distribución normal y
homocedasticidad
% Estas condiciones son necesarias para aplicar ciertas pruebas estadísticas
paramétricas

[normalidad_tabla, variables_NyH] = NormalidadYHomoscedasticidad(MD);

%% ------------------ ANÁLISIS DE HIPÓTESIS --------------------------------
% Se calculan los p-valores globales y por pares para evaluar la relación
entre
% las variables independientes y la variable objetivo

[pval_global, pval_pares] = p_valor(MD, valoracion_1, ...
                                    nombres_continuas, nombres_categoricas);



function rho_table = calcularSpearman(MD, variable_objetivo)
    % CALCULARSPEARMAN calcula los coeficientes de correlación de Spearman
    % entre todas las variables de una tabla y una variable objetivo.
    %
    % INPUTS:
    %   MD - tabla con los datos
    %   variable_objetivo - nombre de la variable objetivo (string o char)
    %
    % OUTPUT:
    %   rho_table - tabla con nombres de variables y coeficientes rho

    n_variables = width(MD) - 2;
    rho_todos = zeros(1, n_variables);
    var_names = MD.Properties.VariableNames(1:end-2); % nombres de variables

    for i = 1:n_variables
        x = MD{:, i};
        y = MD.(variable_objetivo);
        rho = corr(x, y, 'Type', 'Spearman');
        rho_todos(i) = rho;
    end

    % Crear tabla de resultados
    rho_table = table(var_names', rho_todos', ...
```

```
        'VariableNames', {'Variable', 'Rho'});

    % Mostrar resultados
    disp(rho_table);

    % Graficar resultados
    figure;
    bar(rho_todos);
    title('Coeficientes de correlación de Spearman (\rho)', 'FontWeight',
'bold');
    ylabel('Valor de \rho');
    set(gca, 'XTick', 1:length(var_names), ...
             'XTickLabel', var_names, ...
             'TickLabelInterpreter', 'none', ...
             'XTickLabelRotation', 45,...
             'FontSize', 10);
    grid on;
    ylim([-1 1]);
    yline(0, '--k'); % línea base en 0
end



function tabla_frec = frecuencias(MD, nombres_categoricas)
% FRECUENCIAS - Cálculo de frecuencias absolutas y proporcionales para
variables categóricas
%
% Entradas:
%   MD                  : tabla con datos (puede contener variables numéricas
y categóricas)
%   nombres_categoricas: celda con nombres (strings) de las variables
categóricas a analizar
%
% Salida:
%   tabla_frec : celda con tablas que resumen frecuencias y proporciones por
variable

    n_vars = length(nombres_categoricas);      % Número de variables
categóricas
    tabla_frec = cell(1, n_vars);              % Inicializar celda de
resultados

    for i = 1:n_vars
        var_name = nombres_categoricas{i};     % Nombre de la variable
        x = MD.(var_name);                     % Extraer columna
correspondiente

        % Convertir a categórica si no lo es
        if ~iscategorical(x)
            x = categorical(x);
        end

        % Obtener categorías únicas y sus frecuencias
        categorias = categories(x);            % Lista de categorías posibles
        frecuencia_abs = countcats(x);         % Frecuencia absoluta por
categoría
        frecuencia_rel = frecuencia_abs / sum(frecuencia_abs);  % Proporción
relativa
```

```
        % Crear tabla de resultados para esta variable
        tabla_frec{i} = table(categorias, frecuencia_abs, frecuencia_rel, ...
            'VariableNames', {'Categoría', 'Frecuencia', 'Proporción'});
    end

    % Mostrar resultados por pantalla (opcional)
    disp('--- Tabla resumen de frecuencias y proporciones ---');
    for i = 1:n_vars
        disp(['Variable: ', nombres_categoricas{i}]);
        disp(tabla_frec{i});
        disp(' ');
    end
end



function stats_table = medianas_iqr(MD, nombres_continuas)
% MEDIANAS_IQR - Calcula la mediana e IQR para variables continuas
%
% Entradas:
%   MD                 : tabla de datos (con columnas numéricas)
%   nombres_continuas : celda con los nombres de las variables continuas
%
% Salida:
%   stats_table : tabla resumen con mediana e IQR por variable

    n_vars = length(nombres_continuas);              % Número de variables
    medianas = zeros(1, n_vars);                     % Inicializar vector de
medianas
    rango_intercuartil = zeros(1, n_vars);           % Inicializar vector de
IQRs

    for i = 1:n_vars
        var_name = nombres_continuas{i};             % Nombre de la variable
        var_data = MD.(var_name);                    % Extraer columna de
datos
        medianas(i) = median(var_data);              % Calcular mediana
        rango_intercuartil(i) = iqr(var_data);       % Calcular IQR
    end

    % Crear tabla con resultados
    stats_table = table(nombres_continuas', medianas', rango_intercuartil',
...
        'VariableNames', {'Variable', 'Mediana', 'Rango_intercuartil'});
End



function [resultados, variables_validas] = NormalidadYHomoscedasticidad(MD)
% NORMALIDADYHOMOSCEDASTICIDAD - Evalúa normalidad y homocedasticidad de
variables
%
% Entradas:
%   MD : tabla con variables numéricas
%
% Salidas:
%   resultados        : tabla con los p-valores e interpretaciones de cada
prueba
```

```
%   variables_validas : celda con nombres de variables que cumplen ambos
supuestos

    vars = MD.Properties.VariableNames;
    n_vars = width(MD) - 2;  % Excluir últimas columnas si son no numéricas o
metadatos

    % Inicializar tabla de resultados
    resultados = table('Size', [n_vars 5], ...
        'VariableTypes', {'string', 'double', 'string', 'double', 'string'},
...
        'VariableNames', {'Variable', 'p_Normalidad', 'Normalidad',
'p_Levene', 'Homoscedasticidad'});

    variables_validas = {};  % Inicializar lista de variables que cumplen
ambos criterios

    for i = 1:n_vars
        nombreVar = vars{i};
        variable = MD.(nombreVar);

        % Test de normalidad de Lilliefors (basado en Kolmogorov–Smirnov)
        warning('off', 'all');  % Evitar advertencias por valores constantes
o vacíos
        [~, p_norm] = lillietest(variable);
        warning('on', 'all');

        if p_norm > 0.05
            normalidad = "Normal";

            % Crear grupos por encima y debajo de la mediana para test de
Levene
            grupo = variable > median(variable);

            % Test de homocedasticidad de Levene
            p_levene = vartestn(variable, grupo, ...
                'TestType', 'LeveneAbsolute', 'Display', 'off');

            if p_levene > 0.05
                homosced = "Homocedástica";
                variables_validas{end+1} = nombreVar;
            else
                homosced = "No homocedástica";
            end
        else
            normalidad = "No normal";
            p_levene = NaN;
            homosced = "-";
        end

        % Registrar resultados
        resultados.Variable(i) = nombreVar;
        resultados.p_Normalidad(i) = p_norm;
        resultados.Normalidad(i) = normalidad;
        resultados.p_Levene(i) = p_levene;
        resultados.Homoscedasticidad(i) = homosced;
    end
end
```

```
function [pval_globales, pval_pares] = p_valor(MD, variable_objetivo, ...
    variables_continuas, variables_categoricas)
%p_valor: Evalúa diferencias entre grupos mediante pruebas estadísticas.
%
% Entradas:
%   MD                  : Tabla de datos
%   variable_objetivo   : Variable categórica que define los grupos
%   variables_NyH       : Variables continuas normales y homocedásticas
%   variables_continuas : Variables continuas no paramétricas
%   variables_categoricas : Variables categóricas
%
% Salidas:
%   pval_globales : Vector de p-valores de las pruebas globales
%   pval_pares    : Tabla con p-valores de comparaciones por pares

    nombres_var = {};
    pval_globales = [];

    % Inicializar tabla vacía para resultados por pares
    pval_pares = table('Size', [0, 5], ...
        'VariableTypes', {'string', 'categorical', 'categorical', 'double',
'string'}, ...
        'VariableNames', {'VarName', 'Grupo1', 'Grupo2', 'Pvalor', 'Tipo'});


    %% 1. Variables continuas no paramétricas (Kruskal-Wallis + RankSum por
pares)
    for i = 1:length(variables_continuas)
        var_name = variables_continuas{i};
        var_data = MD.(var_name);
        p_kw = kruskalwallis(var_data, variable_objetivo, 'off');
        pval_globales(end+1) = p_kw;
        nombres_var{end+1} = var_name;

        groups = unique(variable_objetivo);
        nGroups = length(groups);
        idx = 1;
        n_pairs = nchoosek(nGroups, 2);

        % Inicializar vectores
        varNames = repmat({var_name}, n_pairs, 1);
        grupos1 = categorical([]);
        grupos2 = categorical([]);
        pvalores = zeros(n_pairs, 1);
        tipos = repmat({'continua'}, n_pairs, 1);

        % Comparaciones por pares con RankSum
        for k = 1:nGroups-1
            for j = k+1:nGroups
                xi = var_data(variable_objetivo == groups(k));
                xj = var_data(variable_objetivo == groups(j));
                p_mw = ranksum(xi, xj);  % Mann-Whitney U test
                grupos1(idx, 1) = categorical(groups(k));
                grupos2(idx, 1) = categorical(groups(j));
                pvalores(idx, 1) = p_mw;
                idx = idx + 1;
```

```
            end
        end

        temp_table = table(varNames, grupos1, grupos2, pvalores, tipos, ...
            'VariableNames', {'VarName', 'Grupo1', 'Grupo2', 'Pvalor',
'Tipo'});
        pval_pares = [pval_pares; temp_table];
    end

    %% 2. Variables categóricas (Chi-cuadrado global y por pares)
    for i = 1:length(variables_categoricas)
        var_name = variables_categoricas{i};
        var_data = MD.(var_name);
        [~, ~, p_chi] = crosstab(var_data, variable_objetivo);
        pval_globales(end+1) = p_chi;
        nombres_var{end+1} = var_name;

        groups = unique(variable_objetivo);
        nGroups = length(groups);
        idx = 1;
        n_pairs = nchoosek(nGroups, 2);

        varNames = repmat({var_name}, n_pairs, 1);
        grupos1 = categorical([]);
        grupos2 = categorical([]);
        pvalores = zeros(n_pairs, 1);
        tipos = repmat({'categorica'}, n_pairs, 1);

        for k = 1:nGroups-1
            for j = k+1:nGroups
                mask = variable_objetivo == groups(k) | variable_objetivo ==
groups(j);
                xi = var_data(mask);
                xj = variable_objetivo(mask);
                [~, ~, p_chi2] = crosstab(xi, xj);
                grupos1(idx, 1) = categorical(groups(k));
                grupos2(idx, 1) = categorical(groups(j));
                pvalores(idx, 1) = p_chi2;
                idx = idx + 1;
            end
        end

        temp_table = table(varNames, grupos1, grupos2, pvalores, tipos, ...
            'VariableNames', {'VarName', 'Grupo1', 'Grupo2', 'Pvalor',
'Tipo'});
        pval_pares = [pval_pares; temp_table];
    end

    %% 3. Visualización de p-valores globales
    % Colores: rojo si p < 0.05, azul en caso contrario
    x = 1:length(pval_globales);  % posiciones personalizadas

    bar_colors = repmat([0 0 1], length(pval_globales), 1);  % azul
    bar_colors(pval_globales < 0.05, :) = repmat([1 0 0], sum(pval_globales <
0.05), 1);  % rojo

    figure;
    b = bar(x, pval_globales, 0.4, 'FaceColor', 'flat');  % ancho reducido a
0.4
```

```
    b.CData = bar_colors;

    title('P Valores de las pruebas estadísticas');
    ylabel('P-Valor');

    set(gca, 'XTick', x, ...
        'XTickLabel', nombres_var, ...
        'TickLabelInterpreter', 'none', ...
        'XTickLabelRotation', 45, ...
        'FontSize', 10);

    grid on;
    ylim([0 1]);
    yline(0.05, '--r', 'Umbral p = 0.05');

    tabla_pvalores = table(nombres_var', pval_globales', ...
    'VariableNames', {'Variable', 'P_Valor'});

    % Mostrar la tabla
    fprintf('\n=== TABLA DE P-VALORES GLOBALES ===\n');
    disp(tabla_pvalores);
end
```

3. “pareado_seleccion”

```
%% ------------------------------------------------------------------------
%% ------------- DIVISIÓN DE LOS DATOS EN ENTRENAMIENTO Y TEST--------------
%% ------------------------------------------------------------------------

% Se crean los conjuntos de entrenamiento y test asegurando el balance de
% las variables sociodemográficas en ambos grupos.

load('MD.mat'); % Carga de la tabla completa de datos

% La función 'crearGruposBalanceados' divide el conjunto de datos en un 50%
% para entrenamiento y otro 50% para test, usando 1000 iteraciones para
% asegurar una distribución adecuada.
[Xtrain, Y1train, Y2train, Xtest, Y1test, Y2test] =
crearGruposBalanceados(MD, 0.5, 1000);

%% ------------------------------------------------------------------------
%%----------- NORMALIZACIÓN Y ESTANDARIZACIÓN DE VARIABLES CONTINUAS-------
%% ------------------------------------------------------------------------

% Se estandarizan las variables continuas en el conjunto de entrenamiento
% y se aplica la misma transformación al conjunto de test para garantizar
% coherencia en la escala de los datos.

nombres_continuas = {'EDAD', 'PESO', 'DÍAS ESTANCIA', 'BIS', 'RASS', ...
    'GLASGOW','TAM', 'FRECUENCIA CARDÍACA', 'FRECUENCIA RESPIRATORIA', ...
    'ESCID', 'APACHE II', 'AMINAS DOSIS', ...
    'SEDACIÓN PROPOFOL (mg/kg/h)', 'SEDACIÓN MIDAZOLAM (mg/h)', ...
    'SEDACIÓN DEXMEDETOMIDINA (mcg/kg/h)', 'ANALGESIA FENTANILO (mcg/kg/h)',
...
    'SEDACIÓN SEVOFLUORANO INHALADO (50 ml/50 ml)', 'SOFA'};

% Estandarización del conjunto de entrenamiento
Xtrain_est = Xtrain;
```

```
for i = 1:length(nombres_continuas)
    var_name = nombres_continuas{i};
    Xtrain_est.(var_name) = zscore(Xtrain_est.(var_name), 0, 1);
end

% Estandarización del conjunto de test con los parámetros del entrenamiento
Xtest_est = Xtest;
for i = 1:length(nombres_continuas)
    var = nombres_continuas{i};
    media = mean(Xtrain.(var), 'omitnan');
    desvest = std(Xtrain.(var), 'omitnan');
    Xtest_est.(var) = (Xtest.(var) - media) ./ desvest;
end

% Guardado de los conjuntos estandarizados
save('Xtrain_est.mat', 'Xtrain_est');
save('Xtest_est.mat', 'Xtest_est');

%% ------------------------------------------------------------------------
%% ----------SELECCIÓN DE CARACTERÍSTICAS (Feature Selection)-----------
%% ------------------------------------------------------------------------

% Se seleccionan las variables más relevantes mediante el algoritmo FCBF
% (Fast Correlation-Based Filter) con un umbral de redundancia definido.

% Obtención de los nombres de las variables
var_names = Xtrain.Properties.VariableNames;

% Conversión a arrays para el algoritmo
Xtrain2 = table2array(Xtrain_est);
Y1train2 = table2array(Y1train); % Variable objetivo

% Aplicación del algoritmo FCBF
redundancyTolerance = 1.25;
[selectedFeatureIndices, completeFeatureIndices, sortedSU, ~] = ...
    featureSelection_FCBF(Xtrain2, Y1train2, redundancyTolerance);

% Variables seleccionadas por su relevancia
var_names_selected = var_names(selectedFeatureIndices);

% Reducción de los conjuntos de datos a las variables seleccionadas
Xtrain_cc = Xtrain_est(:, selectedFeatureIndices);
Xtest_cc = Xtest_est(:, selectedFeatureIndices);

% Guardado de los conjuntos reducidos
save('Xtrain_cc.mat', 'Xtrain_cc');
save('Xtest_cc.mat', 'Xtest_cc');


%% ------------------------------------------------------------------------
%% ----------VISUALIZACIÓN DE LA RELEVANCIA DE LAS VARIABLES--------------
%% ------------------------------------------------------------------------

% Se representa gráficamente la relevancia de las variables según el índice
SU (Symmetrical Uncertainty)

figure;
bar(sortedSU);
title('Relevancia de las variables según Symmetrical Uncertainty');
```

```
ylabel('Valor de SU');
set(gca, 'XTick', 1:length(var_names(completeFeatureIndices)), ...
         'XTickLabel', var_names(completeFeatureIndices), ...
         'XTickLabelRotation', 45);
grid on;

%% ---------------------------------------------------------------------
%% ----------GUARDADO DE CONJUNTOS ORDENADOS POR RELEVANCIA---------------
%% ---------------------------------------------------------------------

% Se reordenan los conjuntos completos según la relevancia de las variables
% (incluso aquellas no seleccionadas final pero con relevancia intermedia).

xtrainprueba = Xtrain_est(:, completeFeatureIndices);
xtestprueba = Xtest_est(:, completeFeatureIndices);
save('xtrainprueba.mat', 'xtrainprueba');
save('xtestprueba.mat', 'xtestprueba');



function [Xtrain, Y1train, Y2train, Xtest, Y1test, Y2test] =
crearGruposBalanceados(MD, porcentajeEntrenamiento, numIter)
% Función que divide una tabla de datos en conjuntos de entrenamiento y test
% manteniendo el equilibrio entre grupos según variables clave (SEXO, EDAD,
VALORACION).
%
% Parámetros de entrada:
% - MD: tabla de datos completa (imputada)
% - porcentajeEntrenamiento: proporción de datos para entrenamiento (ej. 0.5)
% - numIter: número de iteraciones para encontrar la mejor división
%
% Parámetros de salida:
% - Xtrain, Xtest: subconjuntos de variables predictoras
% - Y1train, Y2train, Y1test, Y2test: variables de salida (objetivos)

    % Valores por defecto si no se especifican
    if nargin < 2
        porcentajeEntrenamiento = 0.5;
    end
    if nargin < 3
        numIter = 200;
    end

    % Inicialización
    num_var = size(MD, 2);
    n = height(MD);
    nTrain = floor(n * porcentajeEntrenamiento);
    rng('default');  % Para reproducibilidad
    best_pvalue = 0; % Mayor valor promedio de p encontrado

    % Iteración para encontrar la división más equilibrada
    for i = 1:numIter
        idx = randperm(n);
        train_idx = idx(1:nTrain);
        test_idx = idx(nTrain+1:end);

        train_set = MD(train_idx, :);
        test_set = MD(test_idx, :);
```

```
        % Evaluar el balance entre conjuntos mediante pruebas estadísticas
        % 1. SEXO - prueba de chi-cuadrado
        sexo = [train_set.SEXO; test_set.SEXO];
        grupo = [repmat("train", height(train_set), 1); repmat("test",
height(test_set), 1)];
        [~, ~, p_sexo] = crosstab(grupo, sexo);

        % 2. VALORACION_ANALGOSEDACION_2 - prueba de chi-cuadrado
        [~, ~, p_valoracion] = crosstab(train_set.('VALORACIÓN ANALGOSEDACIÓN
2'), ...
                                        test_set.('VALORACIÓN ANALGOSEDACIÓN
2'));

        % 3. EDAD - prueba no paramétrica de Kruskal-Wallis
        edades = [train_set.EDAD; test_set.EDAD];
        grupos = [repmat("train", height(train_set), 1); repmat("test",
height(test_set), 1)];
        p_edad = kruskalwallis(edades, grupos, 'off');

        % Promedio de p-values para medir balance global
        avg_p = mean([p_sexo, p_valoracion, p_edad]);

        % Actualizar la mejor división si es más equilibrada
        if avg_p > best_pvalue
            best_pvalue = avg_p;
            best_train = train_set;
            best_test = test_set;
        end
    end

    % Separación en variables predictoras (X) y de salida (Y1, Y2)
    Xtrain = best_train(:, 1:end-2);
    Y1train = best_train(:, num_var);
    Y2train = best_train(:, num_var-1);
    Xtest = best_test(:, 1:end-2);
    Y1test = best_test(:, num_var);
    Y2test = best_test(:, num_var-1);

    % Guardar conjuntos en archivos .mat
    save('Xtrain.mat', 'Xtrain');
    save('Xtest.mat', 'Xtest');
    save('Y1train.mat', 'Y1train');
    save('Y2train.mat', 'Y2train');
    save('Y1test.mat', 'Y1test');
    save('Y2test.mat', 'Y2test');

    % ---------------------------
    % Gráficos para verificar balance
    % ---------------------------

    % 1. Gráfico de barras para distribución de SEXO
    [sexo_counts, ~, ~] = crosstab([Xtrain.SEXO; Xtest.SEXO], ...
        [repmat("Entrenamiento", height(Xtrain), 1); repmat("Test",
height(Xtest), 1)]);

    figure;
    bar(sexo_counts);
    set(gca, 'XTickLabel', {'Hombre', 'Mujer'});
    legend({'Entrenamiento', 'Test'}, 'Location', 'best');
```

```
    title('Distribución de Sexo en Entrenamiento y Test');
    ylabel('Frecuencia');
    xlabel('Sexo');

    % 2. Boxplot para comparación de EDAD
    figure;
    boxplot([Xtrain.EDAD; Xtest.EDAD], ...
        [repmat("Entrenamiento", height(Xtrain), 1); repmat("Test",
height(Xtest), 1)]);
    title('Distribución de Edad en Entrenamiento y Test');
    ylabel('Edad');

    % 3. Gráfico de barras para VALORACIÓN
    Y1train_cat = categorical(table2array(Y1train));
    Y1test_cat = categorical(table2array(Y1test));
    grupo_val = [repmat("Entrenamiento", height(Y1train), 1); repmat("Test",
height(Y1test), 1)];
    [val_counts, ~, ~] = crosstab([Y1train_cat; Y1test_cat], grupo_val);

    figure;
    bar(val_counts);
    set(gca, 'XTickLabel', categories(Y1train_cat));
    legend({'Entrenamiento', 'Test'}, 'Location', 'best');
    title('Distribución de Valoración de Sedación en Entrenamiento y Test');
    ylabel('Frecuencia');
    xlabel('Valoración');
end



function [selectedFeatureIndices, completeFeatureIndices, sortedSU,
finalSelectedFeatures] = featureSelection_FCBF(Xtrain, classLabels,
redundancyTolerance)
% featureSelection_FCBF realiza la selección de características utilizando el
método FCBF (Fast Correlation-Based Filter).
% Este método selecciona características relevantes basándose en la
información mutua y elimina las características redundantes.
%
% ENTRADAS:
%   * Xtrain: Matriz de características (Sujetos x Características). Cada
fila representa un sujeto y cada columna una característica.
%             Las características se normalizan dentro de la función.
%   * classLabels: Vector columna (Sujetos x 1) con las etiquetas de clase.
%   * redundancyTolerance: Tolerancia a la redundancia (por ejemplo, 1.25
permite un 25% de redundancia).
%
% SALIDAS:
%   * selectedFeatureIndices: Índices de las características seleccionadas,
ordenadas por importancia.
%   * completeFeatureIndices: Conjunto completo de índices ordenados antes de
eliminar la redundancia.
%   * sortedSU: Valores de Incertidumbre Simétrica (SU) ordenados.
%   * finalSelectedFeatures: Índices finales de las características
seleccionadas (sin redundancia).
%
% Versión: 1.1
% Creado: 15 de diciembre de 2016
% Última modificación: 3 de abril de 2025
```

```
% Autor: Grupo de Ingeniería Biomédica (actualizado con mejoras y corrección
de errores)

    % Normalizar características a media cero y varianza unitaria
    normalizedFeatures = zscore(Xtrain);

    % Calcular información mutua y normalizar con Incertidumbre Simétrica
(SU)
    symmetricalUncertainty = [];
    numFeatures = size(normalizedFeatures, 2);

    for featureIdx = 1:numFeatures
        [mutualInfo, entropyFeature, entropyClass] =
mutualInformation(classLabels, normalizedFeatures(:, featureIdx));

        if (entropyFeature + entropyClass) > 0  % Evitar división por cero
            SU = 2 * (mutualInfo / (entropyFeature + entropyClass));
        else
            SU = 0;  % Asignar cero si la entropía es cero
        end

        symmetricalUncertainty = [symmetricalUncertainty, SU];
    end

    % Ordenar los valores de SU de forma descendente
    [sortedSU, sortedFeatureIndices] = sort(symmetricalUncertainty,
'descend');
    sortedFeatureMatrix = normalizedFeatures(:, sortedFeatureIndices);

    % Umbral para considerar características (inicialmente todas)
    featureThreshold = numFeatures;

    % Guardar el conjunto completo de índices ordenados antes de eliminar
redundancia
    completeFeatureIndices = sortedFeatureIndices;
    selectedFeatureIndices = sortedFeatureIndices;  % Inicializar lista de
características seleccionadas

    % Eliminar características redundantes de forma iterativa
    primaryIdx = 1;

    while primaryIdx <= featureThreshold
        if selectedFeatureIndices(primaryIdx) ~= 0  % Procesar solo índices
válidos
            dominantFeature = sortedFeatureMatrix(:, primaryIdx);
            secondaryIdx = primaryIdx + 1;

            while secondaryIdx <= featureThreshold
                if selectedFeatureIndices(secondaryIdx) ~= 0
                    secondaryFeature = sortedFeatureMatrix(:, secondaryIdx);
                    [mutualInfo, entropyDominant, entropySecondary] =
mutualInformation(dominantFeature, secondaryFeature);

                    if (entropyDominant + entropySecondary) > 0
                        SU_ij = 2 * (mutualInfo / (entropyDominant +
entropySecondary));

                        % Marcar como redundante si excede la tolerancia
```

```
                        if SU_ij >= redundancyTolerance *
max(sortedSU(secondaryIdx), eps)
                            selectedFeatureIndices(secondaryIdx) = 0;
                        end
                    end
                end
                secondaryIdx = secondaryIdx + 1;
            end
        end
        primaryIdx = primaryIdx + 1;
    end

    % Seleccionar solo las características no redundantes
    selectedFeatureMask = selectedFeatureIndices ~= 0;
    finalSelectedFeatures = selectedFeatureIndices(selectedFeatureMask);
    selectedFeatureIndices = finalSelectedFeatures;  % Asignar al valor de
salida
end



function [mutualInfo, entropyX, entropyY] = mutualInformation(featureX,
featureY)
% mutualInformation calcula la información mutua entre dos variables
continuas
% utilizando estimación de densidad mediante kernel (KDE).
%
% Entradas:
% - featureX: Vector de características X (Nx1)
% - featureY: Vector de características Y (Nx1)
%
% Salidas:
% - mutualInfo: Información mutua entre X e Y
% - entropyX: Entropía marginal de X
% - entropyY: Entropía marginal de Y

    % Número de muestras
    numSamplesX = length(featureX);
    numSamplesY = length(featureY);

    % Normalización a media 0 y varianza 1
    featureX = (featureX - mean(featureX)) / std(featureX);
    featureY = (featureY - mean(featureY)) / std(featureY);

    % Número de bins para estimar la densidad
    numBins = 50;

    % Estimación de la PDF marginal de X mediante KDE
    xBins = linspace(min(featureX), max(featureX), numBins);
    pdfX = zeros(size(xBins));
    for i = 1:length(xBins)
        diffX = xBins(i) - featureX;
        bandwidthX = ((4 / (3 * numSamplesX))^(1/5)) * std(diffX);  % Regla
de Silverman
        kernelX = (1 / (sqrt(2 * pi) * bandwidthX)) * exp(- (diffX .^ 2) / (2
* bandwidthX^2));
        pdfX(i) = mean(kernelX);
    end
```

```
    % Estimación de la PDF marginal de Y
    yBins = linspace(min(featureY), max(featureY), numBins);
    pdfY = zeros(size(yBins));
    for i = 1:length(yBins)
        diffY = yBins(i) - featureY;
        bandwidthY = ((4 / (3 * numSamplesY))^(1/5)) * std(diffY);
        kernelY = (1 / (sqrt(2 * pi) * bandwidthY)) * exp(- (diffY .^ 2) / (2
* bandwidthY^2));
        pdfY(i) = mean(kernelY);
    end

    % Cálculo de entropías marginales (en base 2)
    entropyX = -sum(pdfX .* log2(pdfX + eps)) / length(xBins);
    entropyY = -sum(pdfY .* log2(pdfY + eps)) / length(yBins);

    % Estimación conjunta de la PDF de X e Y
    jointPdf = zeros(numBins, numBins);
    for i = 1:length(xBins)
        for j = 1:length(yBins)
            diffX = xBins(i) - featureX;
            diffY = yBins(j) - featureY;
            bandwidthX = ((4 / (3 * numSamplesX))^(1/5)) * std(diffX);
            bandwidthY = ((4 / (3 * numSamplesY))^(1/5)) * std(diffY);
            kernelX = (1 / (sqrt(2 * pi) * bandwidthX)) * exp(- (diffX .^ 2)
/ (2 * bandwidthX^2));
            kernelY = (1 / (sqrt(2 * pi) * bandwidthY)) * exp(- (diffY .^ 2)
/ (2 * bandwidthY^2));
            jointKernel = kernelX .* kernelY;
            jointPdf(i, j) = mean(jointKernel);
        end
    end

    % Cálculo de información mutua usando la definición de entropía conjunta
    mutualInfo = (10 / (numBins - 1)^2) * ...
        sum(sum(jointPdf .* log2(jointPdf ./ (pdfX' * pdfY + eps))));  %
Evita log(0)
end
```

4. “entrenamiento_validacion_XAI”

```
%% CARGA DE DATOS PREPROCESADOS
% Se cargan las matrices necesarias para el entrenamiento y validación
% del modelo, incluyendo etiquetas originales y ajustadas (Y1, Y2),
% así como conjuntos de entrenamiento y prueba normalizados y reducidos.

load('Y1train.mat')
load('Y2train.mat')
load('Y1test.mat')
load('Y2test.mat')
load('Xtrain_cc.mat')      % Conjunto reducido y estandarizado
load('Xtest_cc.mat')
load('xtestprueba.mat')    % Conjunto completo ordenado por relevancia
load('xtrainprueba.mat')

%% COMPARACIÓN ENTRE ETIQUETAS Y1 vs Y2
% Se comparan las métricas de rendimiento entre las dos variantes de la
% variable objetivo, lo que permite determinar si existe mejora en la
% clasificación tras ajustar etiquetas.
```

```
metricas_rendimiento(Y1train, Y2train);

%% ENTRENAMIENTO CON REGRESIÓN LOGÍSTICA MULTINOMIAL
% Se entrena y evalúa el modelo utilizando regresión logística (LR) con
% ambos tipos de etiquetas y conjuntos de datos (reducido y completo).

% Conjunto reducido etiqueta 1
[bestAcc1, bestN1] = acc_caracteristicas(Xtrain_cc, Y1train, Xtest_cc,
Y1test, 'LR');
[~,~] = trainAndValidateModel('LR', Xtrain_cc, Xtest_cc, Y1train, Y1test);

% Conjunto reducido etiqueta 2
[bestAcc2, bestN2] = acc_caracteristicas(Xtrain_cc, Y2train, Xtest_cc,
Y2test, 'LR');
[~,~] = trainAndValidateModel('LR', Xtrain_cc, Xtest_cc, Y2train, Y2test);

% Conjunto completo ordenado
[bestAcc4, bestN4] = acc_caracteristicas(xtrainprueba, Y1train, xtestprueba,
Y1test, 'LR'); %etiqueta 1
[bestAcc3, bestN3] = acc_caracteristicas(xtrainprueba, Y2train, xtestprueba,
Y2test, 'LR'); %etiqueta 2

%% ANÁLISIS CON LDA (ANÁLISIS DISCRIMINANTE LINEAL)
% Se entrena el modelo utilizando LDA, un algoritmo lineal especialmente
% útil para clasificación multiclase con reducción de dimensionalidad.

% Conjunto reducido etiqueta 1
[bestAcc5, bestN5] = acc_caracteristicas(Xtrain_cc, Y1train, Xtest_cc,
Y1test, 'LDA');
[~,~] = trainAndValidateModel('LDA', Xtrain_cc, Xtest_cc, Y1train, Y1test);

% Conjunto reducido etiqueta 2
[bestAcc6, bestN6] = acc_caracteristicas(Xtrain_cc, Y2train, Xtest_cc,
Y2test, 'LDA');
[~,~] = trainAndValidateModel('LDA', Xtrain_cc, Xtest_cc, Y2train, Y2test);

% Conjunto completo ordenado
[bestAcc8, bestN8] = acc_caracteristicas(xtrainprueba, Y1train, xtestprueba,
Y1test, 'LDA'); %etiqueta 1
[bestAcc7, bestN7] = acc_caracteristicas(xtrainprueba, Y2train, xtestprueba,
Y2test, 'LDA'); %etiqueta 2

%% ENTRENAMIENTO CON AdaBoostM2
% AdaBoostM2 es un algoritmo de ensamblado que mejora la precisión mediante
% el ajuste secuencial de clasificadores débiles.

% Conjunto reducido etiqueta 1
[bestAcc13, bestN13] = acc_caracteristicas(Xtrain_cc, Y1train, Xtest_cc,
Y1test, 'AdaBoostM2');
[correctIdx, model] = trainAndValidateModel('AdaBoostM2', Xtrain_cc,
Xtest_cc, Y1train, Y1test);

% Conjunto reducido etiqueta 2
[bestAcc14, bestN14] = acc_caracteristicas(Xtrain_cc, Y2train, Xtest_cc,
Y2test, 'AdaBoostM2');
[~,~] = trainAndValidateModel('AdaBoostM2', Xtrain_cc, Xtest_cc, Y2train,
Y2test);
```

```
% Conjunto completo ordenado
[bestAcc15, bestN15] = acc_caracteristicas(xtrainprueba, Y1train,
xtestprueba, Y1test, 'AdaBoostM2'); %etiqueta 1
[bestAcc16, bestN16] = acc_caracteristicas(xtrainprueba, Y2train,
xtestprueba, Y2test, 'AdaBoostM2'); %etiqueta 2

%% INTERPRETABILIDAD: CÁLCULO DE VALORES DE SHAPLEY
% Se calculan los valores de Shapley para estimar la contribución individual
% de cada variable a la predicción del modelo AdaBoostM2.

% Se crea un modelo base de árbol con profundidad limitada
t = templateTree('MaxNumSplits', 5);

% Cálculo paralelo de valores de Shapley para los datos correctamente
clasificados
[shap_result_avg, shap_values, mean_shap_values, ...
 sorted_features, idx, shap_values_matrix, explainer] = ...
 parallelCalculateShapleyValues_v2(model, Xtrain_cc(correctIdx,:),
Xtest_cc(correctIdx,:));

%% VISUALIZACIÓN DE VALORES DE SHAPLEY
% Se generan representaciones gráficas que ayudan a interpretar la
% importancia relativa de las variables en el modelo.

plotShapleyResults(mean_shap_values, shap_values_matrix, Y1test, correctIdx);



function metricas_rendimiento(y_test, Ypred, scores)
% Esta función calcula y visualiza métricas de rendimiento para modelos de
clasificación.
%
% Entradas:
% - y_test: etiquetas reales del conjunto de test (tabla o array)
% - Ypred: etiquetas predichas por el modelo
% - scores (opcional): probabilidades o puntuaciones por clase para cálculo
de AUC y curvas ROC
%
% Salidas:
% - Se muestran gráficamente y por consola métricas por clase y globales
% - Se genera una curva ROC por clase si se proporciona 'scores'

    % Conversión a arrays si las entradas son tablas
    if istable(y_test)
        y_test = table2array(y_test);
    end
    if istable(Ypred)
        Ypred = table2array(Ypred);
    end

    clases = unique(y_test);  % Clases presentes

    % 1. Matriz de confusión
    confMat = confusionmat(y_test, Ypred);
    total = sum(confMat, 'all');

    figure('Position', [100, 100, 300, 250]);
    confusionchart(confMat, string(clases));
    title('Matriz de Confusión');
```

```
    % 2. Inicialización de métricas por clase
    n_clases = length(clases);
    TP = zeros(n_clases, 1);
    FP = zeros(n_clases, 1);
    FN = zeros(n_clases, 1);
    TN = zeros(n_clases, 1);

    accuracy_clase = zeros(n_clases, 1);
    sensibilidad = zeros(n_clases, 1);  % Recall
    especificidad = zeros(n_clases, 1);
    PPV = zeros(n_clases, 1);  % Precisión
    NPV = zeros(n_clases, 1);
    f1 = zeros(n_clases, 1);
    aucs = NaN(n_clases, 1);
    lrpos = NaN(n_clases, 1);  % Likelihood Ratio positivo
    lrneg = NaN(n_clases, 1);  % Likelihood Ratio negativo

    calcularROC = nargin > 2 && ~isempty(scores);  % Verificar si se
calcularán curvas ROC

    % 3. Cálculo métrico por clase
    for i = 1:n_clases
        TP(i) = confMat(i, i);
        FP(i) = sum(confMat(:, i)) - TP(i);
        FN(i) = sum(confMat(i, :)) - TP(i);
        TN(i) = total - TP(i) - FP(i) - FN(i);

        PPV(i) = TP(i) / (TP(i) + FP(i) + eps);
        NPV(i) = TN(i) / (TN(i) + FN(i) + eps);
        sensibilidad(i) = TP(i) / (TP(i) + FN(i) + eps);
        especificidad(i) = TN(i) / (TN(i) + FP(i) + eps);
        f1(i) = 2 * (PPV(i) * sensibilidad(i)) / (PPV(i) + sensibilidad(i) +
eps);
        accuracy_clase(i) = (TP(i) + TN(i)) / (total + eps);

        if calcularROC
            y_binaria = y_test == clases(i);
            [~, ~, ~, auc] = perfcurve(y_binaria, scores(:, i), true);
            aucs(i) = auc;
            lrpos(i) = sensibilidad(i) / (1 - especificidad(i) + eps);
            lrneg(i) = (1 - sensibilidad(i)) / (especificidad(i) + eps);
        end
    end

    % 4. Métricas globales
    po = sum(diag(confMat)) / total;
    filas = sum(confMat, 2);      % Totales por clase real
    columnas = sum(confMat, 1);   % Totales por clase predicha
    pe = sum((filas .* columnas')) / (total^2);
    kappa = (po - pe) / (1 - pe);

    accuracy = sum(Ypred == y_test) / height(y_test);
    sensibilidad_general = sum(TP) / (sum(TP) + sum(FN) + eps);
    especificidad_general = sum(TN) / (sum(TN) + sum(FP) + eps);
    PPV_general = sum(TP) / (sum(TP) + sum(FP) + eps);
    NPV_general = sum(TN) / (sum(TN) + sum(FN) + eps);
    LRpos_general = sum(TP) / (sum(FP) + eps);
    LRneg_general = sum(FN) / (sum(TN) + eps);
```

```
    F1score_general = 2 * (PPV_general * sensibilidad_general) / (PPV_general
+ sensibilidad_general + eps);

    % 5. Resultados globales por consola
    fprintf('kappa: %.2f\n', kappa);
    fprintf('Accuracy: %.2f%%\n', accuracy * 100);
    fprintf('Sensibilidad general: %.2f%%\n', sensibilidad_general * 100);
    fprintf('Especificidad general: %.2f%%\n', especificidad_general * 100);
    fprintf('PPV general: %.2f%%\n', PPV_general * 100);
    fprintf('NPV general: %.2f%%\n', NPV_general * 100);
    fprintf('LR+ general: %.2f\n', LRpos_general);
    fprintf('LR- general: %.2f\n', LRneg_general);
    fprintf('F1 score general: %.2f%%\n', F1score_general * 100);

    % 6. Curvas ROC por clase (si se proporcionaron 'scores')
    if calcularROC
        figure(); hold on;
        for i = 1:n_clases
            y_bin = y_test == clases(i);
            [Xroc, Yroc] = perfcurve(y_bin, scores(:, i), true);
            plot(Xroc, Yroc, 'LineWidth', 2, ...
                'DisplayName', sprintf('Clase %s (AUC = %.2f)',
string(clases(i)), aucs(i)));
        end
        plot([0, 1], [0, 1], 'k--');  % Línea diagonal
        xlabel('False Positive Rate');
        ylabel('True Positive Rate');
        title('Curva ROC por clase');
        legend('Location', 'SouthEast'); grid on; axis square;
        hold off;
    end

    % 7. Tabla resumen con métricas por clase
    T = table(clases, TP, FP, FN, TN, accuracy_clase, sensibilidad, ...
              especificidad, PPV, NPV, f1, aucs, lrpos, lrneg, ...
              'VariableNames',
{'Clase','TP','FP','FN','TN','Acc','Se','Sp','PPV','NPV', ...
                                  'F1Score','AUC','LR+','LR-'});
    disp(T);
end



function [bestAcc, bestN] = acc_caracteristicas(X_train, y_train, X_test,
y_test, classifierType)
% Esta función evalúa la precisión de un modelo en función del número de
variables (características) utilizadas.
%
% Entradas:
% - X_train, y_train: datos de entrenamiento (pueden ser tablas o arrays)
% - X_test, y_test: datos de test (igualmente, tabla o array)
% - classifierType: tipo de clasificador ('LDA', 'LR', 'SVM', 'AdaBoostM2')
%
% Salidas:
% - bestAcc: mejor precisión (accuracy) alcanzada
% - bestN: número de características que generó la mejor precisión

    % Conversión a matrices si se entregan como tablas
    if istable(X_test),  X_test = table2array(X_test);  end
```

```
    if istable(y_test),  y_test = table2array(y_test);  end
    if istable(X_train), X_train = table2array(X_train); end
    if istable(y_train), y_train = table2array(y_train); end

    maxFeatures = size(X_train, 2);            % Número total de
características
    accuracies = zeros(maxFeatures, 1);        % Vector para guardar
precisión por número de variables

    % Entrenamiento incremental desde 1 hasta maxFeatures
    for n = 1:maxFeatures
        % Entrenar modelo con las n primeras características
        switch classifierType
            case 'LDA'
                model = fitcdiscr(X_train(:, 1:n), y_train);
            case 'LR'
                model = fitmnr(X_train(:, 1:n), y_train);  % Regresión
logística multinomial
            case 'SVM'
                model = fitcecoc(X_train(:, 1:n), y_train); % SVM multiclase
con ECOC
            case 'AdaBoostM2'
                t = templateTree('MaxNumSplits', 5);
                model = fitcensemble(X_train(:, 1:n), y_train, ...
                    'Method', 'AdaBoostM2', ...
                    'NumLearningCycles', 100, ...
                    'Learners', t, ...
                    'LearnRate', 0.1);
            otherwise
                error('Tipo de clasificador no soportado.');
        end

        % Evaluación sobre test
        y_pred = predict(model, X_test(:, 1:n));
        accuracies(n) = sum(y_pred == y_test) / length(y_test);
    end

    % Seleccionar mejor resultado
    [bestAcc, bestN] = max(accuracies);

    % ----------------------------
    % Visualización de resultados
    % ----------------------------
    figure;
    plot(1:maxFeatures, accuracies, '-o', 'LineWidth', 2); hold on;
    plot(bestN, bestAcc, 'ro', 'MarkerSize', 10, 'LineWidth', 2);
    plot(1:maxFeatures, repmat(accuracies(end), maxFeatures, 1), 'k--');
    trend = smoothdata(accuracies, 'gaussian', 5);
    plot(1:maxFeatures, trend, 'Color', [0.8 0.4 0.4], 'LineWidth', 2);
    hold off;

    xlabel('Número de características incluidas');
    ylabel('Accuracy');
    title('Accuracy vs. número de características');
    legend({'Accuracy por número de características', ...
            'Mejor accuracy', ...
            sprintf('Accuracy con %d características', maxFeatures), ...
            'Tendencia suavizada'}, ...
            'Location', 'SouthEast');
```

```
    grid on;
end



function [correctIdx, model] = trainAndValidateModel(classifierType, X_train,
X_test, y_train, y_test)
% Esta función entrena un modelo de clasificación según el tipo especificado
% y evalúa su rendimiento sobre el conjunto de test.
%
% Entradas:
% - classifierType: cadena con el nombre del clasificador ('LDA', 'LR',
'SVM', 'AdaBoostM2')
% - X_train, y_train: datos de entrenamiento
% - X_test, y_test: datos de test
%
% Salidas:
% - correctIdx: índice lógico de aciertos del modelo sobre el test
% - model: modelo entrenado

    % Entrenamiento según el tipo de clasificador
    switch classifierType
        case 'LDA'
            model = fitcdiscr(X_train, y_train);  % Análisis Discriminante
Lineal

        case 'LR'
            model = fitmnr(X_train, table2array(y_train));  % Regresión
Logística Multinomial

        case 'SVM'
            model = fitcecoc(X_train, y_train);   % Máquina de Vectores de
Soporte (ECOC para multiclase)

        case 'AdaBoostM2'
            t = templateTree('MaxNumSplits', 5);  % Plantilla de árbol simple
            model = fitcensemble(X_train, y_train, ...
                'Method', 'AdaBoostM2', ...
                'NumLearningCycles', 100, ...
                'Learners', t, ...
                'LearnRate', 0.1);

        otherwise
            error('Tipo de clasificador no soportado');
    end

    % Predicción sobre el conjunto de test
    [Ypred, scores] = predict(model, table2array(X_test));

    % Cálculo de métricas con función auxiliar
    metricas_rendimiento(y_test, Ypred, scores);

    % Conversión a array si y_test es una tabla
    if istable(y_test)
        y_test_array = table2array(y_test);
    else
        y_test_array = y_test;
    end
```

```
    % Identificación de muestras correctamente clasificadas
    correctIdx = Ypred == y_test_array;
end




function
[sorted_shap_values,shap_values,mean_shap_values,sorted_features,idx,
shap_values_matrix,explainer] = parallelCalculateShapleyValues_v2(model,
X_train, X_test)
    % Calculate Shapley values for the given model

    % Create an explainer for the model using the training data
    explainer = shapley(model, X_train);

    num_samples = size(X_test, 1);
    shap_values = cell(num_samples, 1);  % Use cell array to store results

    % Use parallel computing if available
    for i = 1:num_samples

        % Calculate Shapley values for each sample and store them
independently
        shap_result = fit(explainer, X_test(i, :));  % Calculate Shapley
values for sample i

        % Store Shapley values in a separate cell for each sample (without
overwriting)
        shap_values{i} = shap_result.ShapleyValues{:, 2}; % Store Shapley
values in a cell
    end

    % Convert cell array to matrix
    shap_values_matrix = cell2mat(shap_values');  % Combine all cell values
into a matrix

    % Calculate mean absolute Shapley values for each feature
    mean_shap_values =  abs(mean(shap_values_matrix,2))';

    % Sort Shapley values and feature names
    features = X_train.Properties.VariableNames;
    [sorted_shap_values, idx] = sort(abs(mean_shap_values), 'descend');
    sorted_features.name = features(idx);
    sorted_features.order = idx;

end




function plotShapleyResults(mean_shap_values, shap_values_matrix, Ytest,
correctIdx)
    % Función para graficar los valores de Shapley
    features={'Peso','Motivo de ingreso','Tensión arterial
media','Ventilación mecánica','Dosis de aminas','Dosis sevoflurano'};
    num_features = length(features);
    [~, idx] = sort(abs(mean_shap_values), 'descend');
    mean_shap_values = mean_shap_values(idx);
```

```
    % Gráfico de barras para importancia promedio de características
    figure;
    barh(fliplr(mean_shap_values), 'FaceColor', [0.2, 0.4, 0.8], 'EdgeColor',
'k', 'LineWidth', 1.2);
    grid on;
    xlabel('Mean absolute Shapley Values', 'FontSize', 12, 'FontWeight',
'bold');
    ylabel('Features', 'FontSize', 12, 'FontWeight', 'bold');
    title('Shapley Importance Plot', 'FontSize', 14, 'FontWeight', 'bold');
    set(gca, 'YTickLabel', fliplr(features), 'YTick', 1:length(features),
'FontSize', 10);
    ax = gca;
    ax.XColor = 'k';
    ax.YColor = 'k';
    ax.LineWidth = 1.2;

    % SHAP Summary Plot para cada clase

    figure;

    Ypredict=table2array(Ytest);
    Ycorrectos=Ypredict(correctIdx);
    indxCorrectosClase0=Ycorrectos==0;
    indxCorrectosClase1=Ycorrectos==1;
    indxCorrectosClase2=Ycorrectos==2;


    % Subplot 1: SHAP Summary Plot para la clase 0
    subplot(1,3,1)
    shap_values_class0 = shap_values_matrix(idx(1:num_features),
indxCorrectosClase0);
    [num_features, num_samples] = size(shap_values_class0);
    feature_idx = repmat(1:num_features, num_samples, 1);
    shap_values_class0 = shap_values_class0';

    scatter(shap_values_class0(:), flip(feature_idx(:)) +
0.09*randn(size(feature_idx(:))), 25, shap_values_class0(:), 'filled',
'MarkerEdgeColor', 'k');
    colorbar;
    colormap('cool');
    grid on;
    ylabel('Features', 'FontSize', 12, 'FontWeight', 'bold');
    xlabel('Shapley Value', 'FontSize', 12, 'FontWeight', 'bold');
    title('SHAP Summary Plot (Clase 0)', 'FontSize', 14, 'FontWeight',
'bold');
    set(gca, 'YTickLabel', fliplr(features(1:num_features)), 'YTick',
1:num_features, 'FontSize', 10);
    ax = gca;
    ax.XColor = 'k';
    ax.YColor = 'k';
    ax.LineWidth = 1.2;

    % Subplot 2: SHAP Summary Plot para la clase 1
    subplot(1,3,2)
    shap_values_class1 = shap_values_matrix(idx(1:num_features),
indxCorrectosClase1);
    [num_features, num_samples] = size(shap_values_class1);
    feature_idx = repmat(1:num_features, num_samples, 1);
```

```
    shap_values_class1 = shap_values_class1';

    scatter(shap_values_class1(:), flip(feature_idx(:)) +
0.09*randn(size(feature_idx(:))), 25, shap_values_class1(:), 'filled',
'MarkerEdgeColor', 'k');
    colorbar;
    colormap('cool');
    grid on;
    ylabel('Features', 'FontSize', 12, 'FontWeight', 'bold');
    xlabel('Shapley Value', 'FontSize', 12, 'FontWeight', 'bold');
    title('SHAP Summary Plot (Clase 1)', 'FontSize', 14, 'FontWeight',
'bold');
    set(gca, 'YTickLabel', fliplr(features(1:num_features)), 'YTick',
1:num_features, 'FontSize', 10);
    ax = gca;
    ax.XColor = 'k';
    ax.YColor = 'k';
    ax.LineWidth = 1.2;

    % Subplot 3: SHAP Summary Plot para la clase 2
    subplot(1,3,3)
    shap_values_class2 = shap_values_matrix(idx(1:num_features),
indxCorrectosClase2);
    [num_features, num_samples] = size(shap_values_class2);
    feature_idx = repmat(1:num_features, num_samples, 1);
    shap_values_class2 = shap_values_class2';

    scatter(shap_values_class2(:), flip(feature_idx(:)) +
0.09*randn(size(feature_idx(:))), 25, shap_values_class2(:), 'filled',
'MarkerEdgeColor', 'k');
    colorbar;
    colormap('cool');
    grid on;
    ylabel('Features', 'FontSize', 12, 'FontWeight', 'bold');
    xlabel('Shapley Value', 'FontSize', 12, 'FontWeight', 'bold');
    title('SHAP Summary Plot (Clase 2)', 'FontSize', 14, 'FontWeight',
'bold');
    set(gca, 'YTickLabel', fliplr(features(1:num_features)), 'YTick',
1:num_features, 'FontSize', 10);
    ax = gca;
    ax.XColor = 'k';
    ax.YColor = 'k';
    ax.LineWidth = 1.2;


end
```