Universidad de Valladolid

Facultad de Medicina

Trabajo Fin de Grado
Grado en Ingeniería Biomédica

Inteligencia artificial aplicada a datos genéticos de pacientes con sepsis

Autor:
D. David Segovia Rodríguez

Tutores:
Dra. Dª. Rocío López Herrero
Dr. D. Fernando Vaquerizo Villar

Valladolid, Septiembre de 2024

TÍTULO:	**Inteligencia artificial aplicada a datos genéticos de pacientes con sepsis**
AUTOR:	**D. David Segovia Rodríguez**
TUTOR/ES:	**Dra. Dª. Rocío López Herrero**
	Dr. D. Fernando Vaquerizo Villar
DEPARTAMENTO:	**Cirugía, Oftalmología, Otorrinolaringología y Fisioterapia**

Tribunal

PRESIDENTE:	**Dr. D. Eduardo Tamayo Gómez**
SECRETARIA:	**Dra. Dª. Rocío López Herrero**
VOCAL:	**Dr. D. Javier Gómez Pilar**
P. SUPLENTE:	**Dra. Dª. Estefanía Gómez Pesquera**
S. SUPLENTE:	**Dra. Dª. Esther Gómez Sánchez**

FECHA:	**Septiembre de 2024**
CALIFICACIÓN:	

Agradecimientos

Quisiera empezar agradeciendo a mis tutores de TFG, Fernando Vaquerizo Villar y Rocío Lopez Herrero, por brindarme la oportunidad de realizar este trabajo. Su experiencia y ayuda han sido fundamentales para el desarrollo y finalización del mismo.

También me gustaría expresar mi gratitud a la Universidad de Valladolid por ofrecerme un entorno académico de calidad, en particular, a la Facultad de Medicina y a la Escuela de Ingenieros Industriales. Un reconocimiento especial a Roberto Hornero Sánchez, creador y antiguo coordinador del grado de Ingeniería Biomédica, por su visión y dedicación al establecer una base sólida para el grado. Al igual que a Jesús Poza Crespo, el actual coordinador del grado, por su continuo liderazgo y compromiso con nosotros.

Agradezco especialmente también a mis compañeros de clase por las experiencias vividas durante la carrera y por crear un ambiente de trabajo motivador y estimulante. Me alegra especialmente que a muchos de ellos les pueda considerar amigos y estoy seguro de que estarán presentes en otras etapas de mi vida.

A mi familia, por su constante apoyo emocional y motivación. Gracias por creer en mí y por brindarme el ánimo necesario para superar los desafíos de este camino académico.

Y finalmente, quiero dar las gracias a todos los participantes del estudio previo cuyos resultados se han usado para la elaboración de mi TFG. Sin su participación y trabajo, este trabajo no habría sido posible.

Muchas gracias a todos.

Resumen

La sepsis se define como una afección médica grave y potencialmente mortal caracterizada por una respuesta inflamatoria desregulada ante una infección. Un diagnóstico precoz y un tratamiento adecuado son esenciales para mejorar el pronóstico. En este contexto, este trabajo de fin de grado (TFG) explora el uso de algoritmos de inteligencia artificial (IA) para identificar clústeres de pacientes con sepsis basados en perfiles genéticos y clínicos, con el objetivo de personalizar el tratamiento y mejorar los resultados clínicos.

Los datos genéticos empleados constan de 3,761 SHAP (*SHapley Additive exPlanations) values* de SNPs (*Single Nucleotide Polymorphisms*) de 187 pacientes españoles, obtenidos de un estudio previo. En dicho estudio se aplicó una metodología de *eXplainable Artificial Intelligence* (XAI), para obtener, para cada paciente, la contribución de cada SNP a la predicción de sepsis en forma de SHAP *values*. A partir de los SHAP *values*, se ha realizado un análisis detallado de las contribuciones genéticas a la sepsis mediante los métodos de *clustering* K-means y DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*), que destacan por su robustez y efectividad para obtener clústeres a partir de los SHAP *values* de cada SNP en cada sujeto. Se llevó a cabo un estudio conjunto detallado de estas dos técnicas, asegurando que los resultados de *clustering* fueran fiables y precisos. Para determinar el número óptimo de clústeres (*K*) para el algoritmo K-means, se utilizaron tanto el método del codo como el método de la silueta, proporcionando resultados de *clustering* validados y precisos.

El análisis reveló 3 clústeres distintos de pacientes con sepsis, cada uno asociado con perfiles genéticos y complicaciones clínicas específicas. El Clúster 0, con 50 sujetos, está asociado al SNP rs74707084 del gen SYNPR, que afecta la comunicación neuronal. Este clúster presenta alta mortalidad a 90 días y prolongada ventilación mecánica, sugiriendo dificultades en la recuperación post-UCI debido a la disfunción en la comunicación neuronal y la respuesta inmune. El Clúster 1, con 100 sujetos, está asociado con el SNP rs1575081785 del gen RBSN, implicado en el tráfico de vesículas y la regulación de la inflamación y la respuesta inmune. Este clúster también muestra la menor mortalidad y duración de ventilación mecánica, indicando una mejor respuesta clínica gracias a un manejo médico intensivo. El Clúster 2, con 37 sujetos, está asociado al SNP rs17653532 del gen PRIM2, crucial para la replicación del ADN (Ácido DesoxirriboNucleico). Este clúster tiene las peores variables clínicas, con alta mortalidad y ventilación mecánica prolongada, reflejando complicaciones graves relacionadas con defectos en la reparación del ADN y la respuesta inmune.

El análisis llevado a cabo en este TFG concluye que la identificación de estos clústeres puede facilitar la personalización de las estrategias de tratamiento de la sepsis, permitiendo predicciones más precisas de los resultados clínicos y una mejor planificación de recursos en entornos hospitalarios.

Palabras clave

Sepsis, Inteligencia artificial (IA), SNP (*Single Nucleotide Polymorphism*), SHAP *values, clustering*, K-means.

Abstract

Sepsis is defined as a serious and potentially life-threatening medical condition characterized by a dysregulated inflammatory response to an infection. Early diagnosis and appropriate treatment of sepsis are essential for improving its prognosis. In this context, this final degree project explores the use of artificial intelligence (AI) algorithms to identify clusters of sepsis patients based on genetic and clinical profiles, with the aim of personalizing treatment and improving clinical outcomes.

The genetic data used in this project consists of 3,761 SHAP (SHapley Additive exPlanations) values from SNPs (Single Nucleotide Polymorphisms) of 187 Spanish patients, obtained from a previous study. In that study, an eXplainable Artificial Intelligence (XAI) methodology was applied to obtain, for each patient, the contribution of each SNP to the prediction of sepsis in the form of SHAP values. Based on these SHAP values, a detailed analysis of the genetic contributions to sepsis was conducted using the K-means and DBSCAN (Density-Based Spatial Clustering of Applications with Noise) clustering methods, which are known for their robustness and effectiveness in obtaining clusters from the SHAP values of each SNP in each subject. A comprehensive joint study of these two techniques was carried out, ensuring that the clustering results were reliable and accurate. To determine the optimal number of clusters (K) for the K-means algorithm, both the elbow method and the silhouette method were used, providing validated and precise clustering results.

The analysis revealed three distinct clusters of sepsis patients, each associated with specific genetic profiles and clinical complications. Cluster 0, with 50 subjects, is associated with the SNP rs74707084 in the SYNPR gene, which affects neuronal communication. This cluster presents high 90-day mortality and prolonged mechanical ventilation, suggesting difficulties in post-ICU recovery due to dysfunction in neuronal communication and immune response. Cluster 1, with 100 subjects, is associated with the SNP rs1575081785 in the RBSN gene, involved in vesicle trafficking and the regulation of inflammation and immune response. This cluster also shows the lowest mortality and duration of mechanical ventilation, indicating a better clinical response due to intensive medical management. Cluster 2, with 37 subjects, is associated with the SNP rs17653532 in the PRIM2 gene, which is crucial for DNA (DeoxyriboNucleic Acid) replication. This cluster has the worst clinical variables, with high mortality and prolonged mechanical ventilation, reflecting severe complications related to defects in DNA repair and immune response.

The analysis conducted in this project concludes that identifying these clusters can facilitate the personalization of sepsis treatment strategies, allowing for more accurate predictions of clinical outcomes and better resource planning in hospital settings.

Keywords

Sepsis, artificial intelligence (AI), SNP (*Single Nucleotide Polymorphism*), SHAP values, clustering, K-means.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1. Introducción

1.1 Sepsis

La sepsis se define como una afección médica grave y potencialmente mortal (Singer et al., 2016; World Health Organization, 2024). La sepsis surge cuando la respuesta del cuerpo a una infección, bacteriana, viral, fúngica o parasitaria, daña sus propios tejidos y órganos, produciendo una disfunción orgánica y pudiendo llegar a provocar una insuficiencia multiorgánica o incluso la muerte (Singer et al., 2016; World Health Organization, 2024). Además, es una de las causas principales de mortalidad en pacientes de todo el mundo, según la organización mundial de la salud (WHO) (World Health Organization, 2024). Se estima que la incidencia de sepsis es de aproximadamente 189 casos por cada 100,000 adultos al año, con una tasa de mortalidad elevada, alrededor del 26.7%, siendo aún mayor en pacientes con shock séptico (Fleischmann-Struzek et al., 2020). Además, esta afección conlleva un gasto sanitario elevado, llegando a superar el coste medio por episodio de US$22,100 (van den Berg et al., 2022).

A lo largo de la historia, la definición de sepsis ha ido evolucionando conforme se ha avanzado en el conocimiento de la enfermedad. La primera definición formal, conocida como Sepsis-1, surgió en 1991 y definió la sepsis como un síndrome de respuesta inflamatoria sistémica (SIRS) en respuesta a una infección. SIRS se caracteriza por dos o más de los siguientes criterios: temperatura >38°C o <36°C, frecuencia cardíaca >90 latidos por minuto, frecuencia respiratoria >20 respiraciones por minuto o PaCO2 <32 mm Hg (4.3 kPa), recuento de glóbulos blancos >12,000/mm³ o <4,000/mm³, o >10% de bandas inmaduras (Singer et al., 2016). Sin embargo, esta definición presentaba limitaciones significativas, como un enfoque excesivo en la inflamación y la idea errónea de que la sepsis sigue un continuo desde la sepsis grave hasta el shock séptico. Además, los criterios de SIRS demostraron una especificidad y sensibilidad inadecuadas para identificar de manera precisa la sepsis, lo que generó inconsistencias en la incidencia reportada y la mortalidad observada (Singer et al., 2016). Posteriormente, en 2001, la definición fue revisada bajo la denominación Sepsis-2, que incorporó no solo criterios clínicos sino también marcadores de laboratorio, con el objetivo de mejorar la precisión en el diagnóstico (Singer et al., 2016). A pesar de estos avances, persistieron problemas relacionados con la terminología y la identificación de disfunción orgánica, lo que llevó a la coexistencia de múltiples definiciones y términos para sepsis, shock séptico y disfunción orgánica (Singer et al., 2016).. Finalmente, en 2016, se introdujo la revisión más reciente, conocida como Sepsis-3, que redefinió la sepsis como una disfunción orgánica potencialmente mortal causada por una respuesta desregulada del huésped a la infección. Con esta nueva definición, el término "sepsis grave" fue eliminado por considerarse redundante, ya que la sepsis misma implica un riesgo significativo de disfunción orgánica y mortalidad. La escala SOFA (*Sequential Organ Failure Assessment*) y qSOFA (*quick* SOFA) fueron adoptadas como herramientas clave para evaluar la gravedad de la sepsis y su impacto en la mortalidad del paciente, abordando así las limitaciones de las definiciones anteriores y mejorando la especificidad y sensibilidad en el diagnóstico de esta condición crítica (Singer et al., 2016).

1.1.1 Diagnóstico

La sepsis puede detectarse mediante un cambio de 2 puntos en la puntuación total SOFA (Singer et al., 2016) como consecuencia de la infección (ver Tabla 1.1). Esto implica un riesgo de mortalidad global de aproximadamente el 10% en una población hospitalaria general con sospecha de infección.

Parámetros	**0**	**1**	**2**	**3**	**4**
Respiración: PaO_2/FiO_2 (mmHg)	≥400	<400	300	<200 con ventilación mecánica	<100 con ventilación mecánica
Sistema nervioso central: Escala de coma de Glasgow	15	13-14	10-12	6-9	<6
Sistema cardiovascular (PAM o dosis de catecolaminas)	PAM ≥ 70mmHg	PAM<70 mmHg	Dop < 5 o Dob (cualquier dosis)	Dop 5.1 -15 o Epi ≤ 0.1 o Nor ≤0.1	Dop >15 o Epi > 0.1o Nor > 0.1
Hígado: bilirrubina (mg/dl)	<1.2	1.2 – 1.9	2.0 – 5.9	6.0-11.9	> 12.0
Coagulación: plaquetas$\times 10^3/\mu L$	>150	<150	< 100	< 50	< 20
Sistema renal: creatinina (mg/dL), aclaramiento, (mL/d)	<1.2	1.2-1.9	2.0-3.4	3.5-4.9 < 500	> 5.0 < 200

Tabla 1.1 *Puntuación secuencial para la evaluación de la insuficiencia orgánica (SOFA). Abreviaturas: PaO2, presión parcial de oxígeno; FiO2, presión parcial de oxígeno; PAM, presión arterial media; Dop, dopamina; Dob, dobutamina; Epi, epinefrina; Nor. Tabla procedente de: https://multimedia.elsevier.es/PublicationsMultimediaV1/item/multimedia/S0034935612002423:mmc1.pdf?idApp=UINPBA00004N*

Por otra parte, en entornos fuera del hospital, en el departamento de emergencias o en salas generales de hospitales, los pacientes adultos con sospecha de infección pueden ser rápidamente identificados como más propensos a resultados adversos típicos de la sepsis si presentan al menos dos de los siguientes criterios clínicos, que constituyen la escala qSOFA: una frecuencia respiratoria de 22 respiraciones por minuto o más, alteración del estado mental o una presión arterial sistólica de 100 mmHg o menos (Singer et al., 2016).

El shock séptico se define como un subtipo de sepsis caracterizado por alteraciones circulatorias, celulares y metabólicas extremadamente graves, que conllevan un riesgo de mortalidad significativamente mayor en comparación con la sepsis sola. Los pacientes con shock séptico se identifican clínicamente por la necesidad de vasopresores para mantener una presión arterial media de al menos 65 mm Hg y por niveles elevados de lactato sérico superiores a 2 mmol/L (>18 mg/dL) en ausencia de hipovolemia. Esta combinación de criterios se asocia con tasas de mortalidad hospitalaria superiores al 40% (Singer et al., 2016).

De cara al diagnóstico también es importante tener en cuenta los síntomas y los factores de riesgo (World Health Organization, 2024). Entre los factores de riesgo encontramos:

- Adultos mayores de 65 años.
- Personas con afecciones crónicas, como diabetes, enfermedad pulmonar, cáncer y enfermedad renal.
- Personas con sistemas inmunitarios debilitados, pacientes de unidad de cuidados intensivos (UCI) u hospitalizados.
- Mujeres embarazadas.
- Niños menores de un año.

Asimismo, los síntomas más comunes son:

- Frecuencia cardíaca elevada, pulso débil o hipotensión arterial.
- Dificultad para respirar.
- Confusión o desorientación.
- Dolor o molestia extrema.
- Fiebre, escalofríos o sensación de mucho frío.
- Piel húmeda o sudorosa.

1.1.2 Formación y fisiopatología

El desarrollo de la sepsis puede describirse en las siguientes etapas (Arora et al., 2023; Jarczak et al., 2021):

1. **Activación del Sistema Inmune Innato.** Ante una infección se produce una respuesta inmunitaria nata. Los patrones moleculares asociados a patógenos se unen a receptores de reconocimiento de patrones en las células inmunes, como los macrófagos, las células dendríticas y los neutrófilos. Estas producen las citocinas, como la interleucina (IL)-1, IL-6, el factor de necrosis tumoral-alfa (TNF-α) y el interferón-gamma (IFN-γ). Dichas citocinas juegan roles cruciales en la coordinación de la respuesta inmune innata y adaptativa, promoviendo la inflamación, la activación y reclutamiento de otras células inmunitarias al sitio de la infección, y modulando la respuesta adaptativa para eliminar el patógeno.
2. **Respuesta Inflamatoria Sistémica.** En algunos casos esta respuesta inmunológica es excesiva lo que induce inflamación en todo el cuerpo, no solo en el sitio de la infección. La inflamación sistémica daña el endotelio (revestimiento interno de los vasos sanguíneos), lo que provoca una mayor permeabilidad vascular y fuga de líquido hacia los tejidos. Esto activa el sistema de coagulación sanguínea, lo que puede llevar a la formación de microtrombos en los vasos sanguíneos pequeños y eventualmente a la coagulación intravascular diseminada.
3. **Daño endotelial y disfunción microcirculatoria**. El daño endotelial produce vasodilatación y fuga de líquidos, causando una disminución de la presión arterial (hipotensión). Junto con los microtrombos, esto reduce el flujo sanguíneo a los órganos y tejidos, provocando hipoxia (falta de oxígeno) y daño tisular. Si esto ocurre en varios órganos estaremos hablando de un fallo multiorgánico. Los órganos comúnmente afectados incluyen los pulmones (síndrome de distrés respiratorio agudo), los riñones (insuficiencia renal aguda), el hígado (disfunción hepática) y el cerebro (encefalopatía séptica).

1.1.3 Tratamiento

Para el tratamiento de la sepsis se toma como referencia las últimas directrices de la campaña SSC (*Surviving Sepsis Campaign*) publicadas en 2021, las cuales se centran en varios aspectos clave para mejorar la identificación temprana y el manejo efectivo de los pacientes con sepsis y shock séptico (Evans et al., 2021; Srzić et al., 2022). El tratamiento recomendado consta de (Evans et al., 2021; Srzić et al., 2022):

1. **Resucitación hidroelectrolítica**. Consiste en administrar al menos 30 mL/kg de fluidos cristaloides intravenosos dentro de las primeras tres horas en pacientes con hipoperfusión inducida por sepsis o shock séptico. En este sentido, se prefiere el uso de cristaloides balanceados sobre la solución salina normal.
2. **Administración de antibióticos**. Se recomienda iniciar antibióticos de amplio espectro inmediatamente o dentro de una hora desde el reconocimiento de shock séptico. En casos de

sepsis grave, sin shock, se deben considerar causas no infecciosas (como pancreatitis aguda o traumatismo), pero es crucial administrar los antibióticos dentro de las tres horas posteriores al reconocimiento de la sepsis.

3. **Soporte hemodinámico**. El objetivo inicial es mantener una presión arterial media de al menos 65 mmHg en pacientes con shock séptico. Si no se logra mantener la presión arterial dentro de la normalidad con la resucitación hídrica, se recomienda el uso de noradrenalina como vasopresor de primera línea. En casos donde la presión arterial media no se mantenga adecuadamente con noradrenalina, se sugiere añadir vasopresina para complementar el tratamiento.
4. **Manejo de la función cardíaca**. En pacientes con shock y disfunción cardíaca persistente, se recomienda el uso de dobutamina.
5. **Manejo respiratorio**. En pacientes con fallo respiratorio inducido por sepsis, se sugiere el uso de oxigenoterapia nasal de alto flujo. Si la oxigenación no es suficiente o en casos de síndrome de distrés respiratorio agudo severo, se recomienda el uso de ventilación mecánica invasiva.
6. **Seguimiento Post-Cuidados Intensivos**. Se enfatiza la importancia de evaluar y seguir a los supervivientes de sepsis o shock séptico para problemas físicos, cognitivos y emocionales después del alta hospitalaria.

1.2 Genética

La genética es el estudio de cómo se trasmiten las características y rasgos de los organismos en los genes de una generación a otra (Anna C. Edens Hurst, 2022). El ser humano tiene 46 cromosomas, 2 cromosomas que determinan su sexo (cromosomas X e Y) y 22 pares de cromosomas no sexuales (autosómicos) (Anna C. Edens Hurst, 2022). Estos cromosomas se componen de genes, la unidad básica de la herencia (NIH, 2024), son segmentos de ácido desoxirribonucleico (ADN) que contienen la información necesaria para especificar los rasgos físicos y biológicos (NIH, 2019). La mayoría de los genes codifican para proteínas específicas, o segmentos de proteínas, que tienen diferentes funciones en el organismo.

A su vez, la estructura del ADN está constituida por nucleótidos. Estos se componen de una molécula de azúcar (ya sea ribosa en el ARN o desoxirribosa en el ADN) unida a un grupo fosfato y a una base nitrogenada. Las bases del ADN son la adenina (A), citosina (C), guanina (G) y timina (T) (NIH, 2019). A veces se produce un cambio o mutación en uno varios genes pudiendo llegar a alterar las instrucciones para fabricar las proteínas y haciendo que las proteínas no funcionen correctamente o falten. Esto puede producir enfermedades, como la sepsis (Qiao et al., 2018).

Las variaciones genéticas más comunes en las personas son los polimorfismos genéticos, en concreto, las que afectan afectar a un solo nucleótido (llamados SNPs, por sus siglas en inglés, *Single Nucleotide Polymorphisms*) y se pueden encontrar en regiones codificantes (exones), no codificantes (intrones) o en regiones intergénicas (entre genes). Para ser consideradas SNPs, estas variaciones deben ocurrir en al menos el 1% de la población (Edwards et al., 2007).

La mayoría de los SNP no afectan a la salud ni el desarrollo. No obstante, algunas de estas variaciones genéticas son cruciales en la investigación de la salud humana. Los SNP pueden predecir cómo responderá una persona a ciertos medicamentos, su vulnerabilidad a factores ambientales como toxinas y el riesgo de contraer enfermedades (Edwards et al., 2007). También se utilizan para rastrear la herencia de enfermedades ligadas a variantes genéticas en las familias y llevar cabo estudios para identificar SNP asociados con enfermedades complejas como las cardiopatías, la diabetes, el cáncer y la sepsis (Edwards et al., 2007).

En el contexto del diagnóstico y pronóstico de la sepsis, se ha demostrado que la respuesta inmune del huésped a los agentes microbianos está influenciada por la variación genética (Skibsted et al., 2013). Asimismo, tecnologías avanzadas como los estudios de asociación del genoma completo (llamados GWAS, por sus siglas en inglés, *Genome Wide Association Studies*) (Rosier et al., 2021) o la creación de puntuaciones de riesgo poligénico (D'Urso et al., 2020; Engoren et al., 2022) han permitido identificar variaciones genéticas que predisponen a los individuos a la sepsis, así como su gravedad y la respuesta al tratamiento. Además, algunas variantes genéticas se han asociado con menor supervivencia en pacientes con sepsis (Hernandez-Beeftink et al., 2022). Por tanto, identificar estas variantes puede ayudar a evaluar el pronóstico de los pacientes y ajustar las terapias para mejorar los resultados clínicos.

1.3 Inteligencia artificial

El término "inteligencia artificial" (IA) fue acuñado en una conferencia en la Universidad de Dartmouth organizada por John McCarthy en 1956 (Sanabria-Navarro et al., 2023), y es una tecnología que permite que los ordenadores resuelvan problemas simulando la inteligencia y las capacidades humanas de resolución de problemas (IBM, 2024). Desde entonces ha ido evolucionando y ha conseguido hitos como en 1997, cuando DeepBlue, la supercomputadora creada por la empresa IBM, se enfrentó y gano al campeón mundial de ajedrez Gary Kasparov siendo la primera vez que una maquina conseguía ganar a un campeón mundial (IBM, 2024).

1.3.1 *Machine Learning*

Machine learning (ML) o aprendizaje automático, es un subcampo de la IA que se centra en el desarrollo de algoritmos y técnicas que permiten a los ordenadores aprender patrones y tomar decisiones a partir de datos, sin intervención humana explícita (El Naqa et al., 2015; Lecun et al., 2015). Aquí se destacan tres enfoques principales:

- **Aprendizaje Supervisado**. Este enfoque implica entrenar un modelo utilizando datos etiquetados, es decir, datos donde se conoce la respuesta correcta. El modelo aprende a hacer predicciones o clasificaciones basadas en ejemplos de entrenamiento previamente etiquetados. Algunos ejemplos de aplicaciones incluyen reconocimiento de voz, clasificación de imágenes y predicción de ventas.

- **Aprendizaje No Supervisado**. El modelo se entrena con datos no etiquetados, buscando patrones intrínsecos o estructuras ocultas en los datos. Se utiliza para agrupar datos similares en clústeres (*clustering*) o para reducir la dimensionalidad de los datos (análisis de componentes principales). Ejemplos de aplicación son la segmentación de clientes, el análisis de redes sociales y la detección de anomalías.

- **Aprendizaje por Refuerzo**. Este enfoque consta de un agente o modelo encargado de aprender a tomar decisiones secuenciales en un entorno con el objetivo de maximizar una recompensa acumulada a lo largo del tiempo. A diferencia del aprendizaje supervisado, donde el modelo se entrena con un conjunto de datos etiquetados, el aprendizaje por refuerzo se basa en la interacción continua del agente con su entorno. El agente recibe retroalimentación en forma de recompensas o castigos basados en las acciones que realiza, lo que le permite ajustar su política para tomar decisiones futuras más efectivas.

1.3.2 *Deep Learning*

Posteriormente, surgió el *deep learning* (DL), una subcategoría del ML, que ha revolucionado la IA en las últimas décadas. El DL se basa en el uso de redes neuronales artificiales con múltiples capas (profundas) para modelar y aprender patrones complejos de los datos. Las redes neuronales profundas han demostrado ser altamente efectivas para procesar grandes volúmenes de datos y

extraer características relevantes en tareas como el reconocimiento de voz, la traducción automática y la detección de objetos en imágenes. Esto ha impulsado su adopción en diversas aplicaciones de IA (El Naqa et al., 2015; Lecun et al., 2015).

1.3.3 IA en la sepsis

La inteligencia artificial ha evolucionado significativamente desde sus inicios, y las técnicas de ML y DL han transformado la capacidad de los modelos para resolver problemas complejos. Sin embargo, la interpretabilidad de estos modelos sigue siendo un desafío, especialmente cuando se aplican en campos críticos como la medicina (Dwivedi et al., 2023).

Aquí es donde entra en juego *eXplainable Artificial Intelligence* (XAI). XAI nos permite entender y confiar en las decisiones tomadas por los modelos de IA, proporcionando explicaciones claras y comprensibles (Dwivedi et al., 2023). En particular, *SHapley Additive exPlanations* (SHAP) de (Lundberg & Lee, 2017) es un método para explicar predicciones individuales que descompone la predicción de un modelo en contribuciones atribuibles a cada característica.

Aplicar SHAP en el contexto de la sepsis permite obtener SHAP *values* de los SNPs para hacer *clustering* de pacientes. Esto no solo facilita una mejor comprensión de cómo los SNPs específicos influyen en la sepsis, sino que también puede ayudar a identificar subgrupos de pacientes con características genéticas similares. Esta información es crucial para personalizar tratamientos, mejorar los pronósticos y optimizar la gestión de los recursos en el sistema de salud. En resumen, la combinación de XAI y el *clustering* nos ofrece herramientas poderosas para avanzar en el diagnóstico y tratamiento personalizado de enfermedades complejas como la sepsis.

1.4 Hipótesis y objetivos

Para la realización de este trabajo de fin de grado (TFG) se plantea la hipótesis de que los algoritmos de IA son de gran utilidad a la hora de identificar distintos subgrupos de pacientes de sepsis con distinto perfil genético. En base a esta hipótesis, el objetivo principal del TFG es identificar, mediante técnicas de IA, clústeres de pacientes con sepsis con distinto perfil clínico y genético. Para lograr este objetivo, se va a clasificar a estos pacientes en grupos, mediante métodos de *clustering*, según la importancia de sus SNPs en el desarrollo de la sepsis. Es importante destacar que este TFG parte de los resultados proporcionados en (López Herrero et al., 2024), donde, mediante XAI, se obtienen los SHAP *values* que miden la contribución de los SNPs de cada sujeto a la predicción de sepsis.

Para conseguir el objetivo principal del TFG, se plantean los siguientes objetivos específicos:

1. Realizar una revisión bibliográfica sobre las técnicas de *clustering* aplicadas a partir de datos genéticos de pacientes con sepsis y/o de SHAP *values*.
2. Seleccionar las técnicas de *clustering* más apropiadas para la identificación de distintos grupos de pacientes de sepsis con distinto perfil genético a partir de nuestros datos (SHAP *values* de los SNPs en pacientes con sepsis).
3. Aplicar las técnicas seleccionadas y analizar su comportamiento según los distintos parámetros de configuración (hiperparámetros) de cada una de ellas para seleccionar los mejores clústeres de pacientes con sepsis.
4. Identificar los sujetos que pertenecen a cada clúster y realizar un análisis estadístico de las variables clínicas de cada clúster.
5. Identificar los SNPs que tienen mayor peso en cada clúster y realizar un análisis funcional de los genes de los SNPs escogidos para cada clúster.
6. Discutir y extraer conclusiones a partir de los resultados del análisis clínico y funcional, así como proponer líneas futuras de investigación.

1.5 Planificación y estructura del TFG

1.5.1 Planificación

Con el propósito de cumplir todos los objetivos del TFG, el proceso de este se ha desarrollado en 3 fases:

1. Adquisición de conocimientos y planteamiento del problema, que consta de los siguientes pasos:
    - Búsqueda bibliográfica de información de la sepsis, etiología, fisiopatología y tratamiento; así como la descripción de los SNPs.
    - Búsqueda bibliográfica de métodos de IA: ML, DL y XAI.
    - Lectura y comprensión del trabajo de investigación previo cuyos resultados son el punto de partida de este TFG (López Herrero et al., 2024).
    - Búsqueda bibliográfica de métodos empleados para *clustering*: *clustering* con genes y *clustering* con SHAP *values*.
    - Repaso de conocimientos de Python y consulta de las librerías necesarias para implementar los métodos de *clustering*, como sklearn.
2. Obtención de los clústeres de pacientes, que consta de los siguientes pasos:
    - Implementación de los métodos de *clustering* y elección del más optimo junto a sus hiperparámetros.
    - Obtención de los 5 SNPs más importantes en cada clúster junto a su correspondiente análisis genético.
    - Análisis estadístico de los datos clínicos de los pacientes por clúster.
3. Interpretación de los resultados e informe:
    - Discusión y extracción de conclusiones a partir de los resultados del análisis funcional y el análisis clínico de los distintos clústeres.
    - Redacción de la memoria del TFG.

1.5.2 Estructura del TFG

El TFG se divide en los siguientes 6 capítulos:

- Capítulo 1: Introducción, en el que se estudia la enfermedad a tratar la sepsis, se aborda su etiología, fisiopatología y su tratamiento. También se trata la genética y el concepto de IA y como esta puede ser útil para tratar nuestro problema. Además, se describen los objetivos e hipótesis del trabajo.
- Capítulo 2: Estado del arte, en el que se define que es el *clustering* y se buscan los métodos de *clustering* más empleados en genes y con SHAP *values*. Posteriormente se comparan los ventajas y desventajas de cada uno para escoger la metodología más adecuada para nuestro caso.
- Capítulo 3: Materiales y métodos, que primero describe la base de datos a emplear los análisis previos hecho sobre ella para la detección de la sepsis. A continuación, describe la metodología de *clustering* aplicada, así como el análisis clínico y funcional de los clústeres.
- Capítulo 4: Resultados, que muestra los resultados de los métodos de *clustering* aplicados, así como los resultados derivados del análisis clínico y funcional de cada clúster.
- Capítulo 5: Discusión, que consta de una interpretación de los resultados obtenidos en base a la bibliografía existente en pacientes con sepsis, así como de la identificación de las principales limitaciones del trabajo.

- Capítulo 6: Conclusiones y líneas futuras, que resalta las principales conclusiones del trabajo, así como posibles futuras investigaciones en el campo de la sepsis.

Capítulo 2. Estado del arte

2.1 Introducción al *clustering*

El *clustering* es una técnica usada en el ámbito de la ciencia de datos para organizar un conjunto de datos en grupos, o clústeres, donde los objetos en el mismo grupo son más similares entre sí que aquellos en diferentes grupos (Ezugwu et al., 2022). Este proceso, que se basa en maximizar la similitud intraclúster y minimizar la similitud interclúster, es crucial para descubrir estructuras naturales en los datos sin necesidad de etiquetas predefinidas. Dentro del *clustering*, se distinguen dos categorías (Ghosal et al., 2020):

- *Hard clustering*: Donde cada elemento solo puede pertenecer a un grupo.
- *Soft clustering*: Donde cada elemento puede pertenecer a varios grupos.

A pesar de que existen varios tipos de algoritmos todos persiguen los siguientes objetivos (Ezugwu et al., 2022):

1. **Identificar patrones ocultos**, lo cual permite descubrir estructuras subyacentes en los datos que no son evidentes a simple vista.
2. **Reducción de dimensionalidad**, ayudando a reducir la complejidad de los datos, facilitando el análisis y visualización.
3. **Segmentación de datos**, que consiste en dividir un conjunto de datos en grupos más pequeños, o segmentos, que comparten características similares. En medicina, permite clasificar pacientes según síntomas o respuestas a tratamientos para ofrecer atención médica personalizada.

En función de la aplicación concreta, existen diversos tipos de *clustering*, cada uno de ellos con una metodología distinta para identificar los grupos, con sus ventajas e inconvenientes. Algunos de ellos son (Ghosal et al., 2020; Saxena et al., 2017):

- ***Clustering* Particional.** Enfoque iterativo que busca similitudes entre los puntos intra-clúster con respecto a sus distancias desde el centroide del clúster. Asume que cada clúster debe tener al menos un punto y que cada punto tiene que pertenecer al menos a un clúster. Algunos de los métodos dentro de este enfoque son:

- K-means: Asigna datos a clústeres minimizando la suma de distancias cuadráticas de los puntos a los centros del clúster.
- PAM (*Partitioning Around Medoids*): Similar a K-means, pero utiliza puntos de datos reales como centros de clústeres, haciéndolo más robusto a los valores espúreos (*outliers*).

- ***Clustering* Jerárquico.** Construye una jerarquía de clústeres utilizando métodos aglomerativos o divisivos:

- Aglomerativo. Comienza con cada punto como un clúster individual y fusiona los clústeres más similares sucesivamente.
- Divisivo. Comienza con un solo clúster que incluye todos los puntos y divide sucesivamente en clústeres más pequeños.

- ***Clustering* Basado en Densidad.** Métodos que identifican clústeres basados en áreas de alta densidad de puntos. Tiene la ventaja de eliminar *outliers* o ruido al usar las zonas de alta densidad como clúster y las zonas de baja densidad como espacio de separación entre estos.

Un método dentro de este enfoque es DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*), que encuentra clústeres de formas arbitrarias basándose en la densidad de puntos en una región.

- **Modelos Basados en Gráficos.** Utilizan grafos para representar relaciones entre los datos y agrupan nodos del grafo en clústeres. En este enfoque se encuentra el método *Spectral Clustering,* que utiliza propiedades del espectro (valores propios) del grafo de afinidad de los datos para realizar el agrupamiento.
- **Modelos Probabilísticos**. Métodos que asumen que los datos son generados por una combinación de distribuciones subyacentes. Emplean métodos matemáticos y algoritmos para definir los clústeres. En este enfoque se encuentra el método *Gaussian Mixture Models* (GMM), que supone que los datos provienen de una mezcla de varias distribuciones gaussianas y utiliza el algoritmo *Expectation-Maximization* para encontrar los parámetros.

2.2 Algoritmos de *clustering*

Antes de decidir el algoritmo que se va a emplear hay que describirlos y especificar que hiperparámetros son necesarios en cada uno. La elección de los algoritmos descritos está basada en su incidencia y efectividad demostrada en la revisión bibliográfica.

2.2.1 K-means

K-means es un algoritmo particional muy utilizado cuyo funcionamiento se basa en minimizar la suma de distancias entre las muestras y el centroide al que pertenecen. Como argumentos de entrada el algoritmo necesita únicamente los datos y el número de clústeres (*K)* (Francisco Sanz, 2024; Sinaga & Yang, 2020). El algoritmo K-means consta de las siguientes ventajas (Francisco Sanz, 2024; Sinaga & Yang, 2020):

- Eficiencia computacional y escalabilidad: K-means es conocido por su simplicidad y rapidez. La implementación es directa y el tiempo de ejecución es relativamente bajo, incluso con grandes conjuntos de datos. Esto lo convierte en una opción práctica y efectiva para tareas de *clustering* donde la velocidad es una consideración importante. Además, es eficiente para grandes volúmenes de datos, especialmente cuando el número de clústeres es pequeño.

- Facilidad de interpretación: Los resultados de K-means son fáciles de interpretar, ya que cada punto de datos se asigna al centroide más cercano. Esta asignación directa permite una comprensión clara de cómo se forman los clústeres, facilitando la toma de decisiones basada en los resultados del *clustering*.

- Convergencia y estabilidad: Aunque K-means puede converger a óptimos locales, su comportamiento es predecible y garantiza la convergencia en cada ejecución. Esto proporciona una base sólida para la reproducibilidad y la estabilidad de los resultados, aspectos cruciales en análisis de datos.

- Flexibilidad: K-means puede adaptarse a diferentes tipos de datos y medidas de distancia. Esto le permite ser aplicado en una variedad de contextos y con diferentes métricas de similitud, aumentando su versatilidad.

Asimismo, K-means consta de las siguientes desventajas (Francisco Sanz, 2024; Sinaga & Yang, 2020):

- Número de clústeres predefinido: El usuario debe especificar el número de clústeres antes de ejecutar el algoritmo, lo que puede no ser siempre intuitivo.

- Sensibilidad a la inicialización: Diferentes inicializaciones pueden llevar a diferentes resultados finales, por lo que puede ser necesario ejecutar el algoritmo varias veces con diferentes inicializaciones.
- Sensible al ruido en los datos, lo que puede alterar los resultados.

2.2.2 *Partitioning Around Medoids*

Al igual que K-means, PAM es un método particional pero en el que en lugar de usar la media de los puntos en un clúster, se usa un punto real del clúster llamado "medoide" como centroide (Reynolds et al., 2004). Este método es más robusto ante *outliers* y datos ruidosos porque el medoide es un punto real en lugar de una media calculada. Al igual que K-means, requiere de dos argumentos, la base de datos y el número de clústeres *K* (Reynolds et al., 2004).

2.2.3 *Hierarchical Clustering*

El *clustering* jerárquico (*Hierarchical Clustering*, HC), en concreto el aglomerativo, construye la jerarquía de clústeres de manera de menos a más. Este algoritmo comienza con cada punto de datos como un clúster individual y luego combina los clústeres más similares de manera iterativa hasta que todos los puntos se encuentran en un solo clúster o se alcanza un número deseado de clústeres. Para ello, es necesario definir el número de clústeres deseado o el umbral de distancia para determinar dónde cortar el dendrograma y formar los clústeres, (si no se fija el algoritmo parará cuando solo haya un clúster), el tipo de enlace (que define cómo se calcula la distancia entre clústeres a medida que se agrupan) y la métrica utilizada para calcular las distancias entre puntos (Bouguettaya et al., 2015; Fatih Karabiber, 2024). El algoritmo *clustering* jerárquico aglomerativo consta de las siguientes ventajas (Bouguettaya et al., 2015; Fatih Karabiber, 2024):

- Visualización clara: Produce un dendrograma que proporciona una representación visual de la estructura jerárquica de los datos.
- Flexibilidad: No requiere especificar el número de clústeres a priori; este se puede determinar observando el dendrograma.
- Captura de la jerarquía natural: Puede capturar la estructura jerárquica natural en los datos.

Asimismo, este algoritmo consta de las siguientes desventajas (Bouguettaya et al., 2015; Fatih Karabiber, 2024):

- Escalabilidad: Puede ser computacionalmente costoso y no es adecuado para conjuntos de datos muy grandes.
- Sensibilidad a *outliers*: Puede ser influenciado por *outliers* y ruido en los datos.
- Irreversibilidad: Una vez que dos clústeres se combinan, no pueden separarse, lo cual puede no ser óptimo en algunos casos.

2.2.4 *Density-Based Spatial Clustering of Applications with Noise*

DBSCAN es un algoritmo de *clustering* basado en la densidad que identifica clústeres basándose en la densidad de puntos en el espacio de datos. Las regiones de datos de alta densidad se agruparán en clústeres mientras que las regiones de baja densidad se marcarán como ruido (DataScientest, 2022; Deng, 2020).

A diferencia de otros algoritmos, no requiere que se especifique el número de clústeres a priori, pero sí que hace falta que se fijen otros hiperparámetros como *Eps*, la distancia máxima entre dos puntos para que se consideren en el mismo clúster y *minPts*, el número mínimo de puntos

necesarios para formar un clúster (DataScientest, 2022; Deng, 2020). El algoritmo DBSCAN consta de las siguientes ventajas (DataScientest, 2022; Deng, 2020):

- Detección de formas arbitrarias: Puede encontrar clústeres de formas arbitrarias.
- Manejo de Ruido: Identifica y maneja puntos de ruido.
- No requiere un número de clústeres predefinido: No es necesario definir el número de clústeres a priori.

Asimismo, el algoritmo DBSCAN consta de las siguientes desventajas (DataScientest, 2022; Deng, 2020):

- Sensibilidad a hiperparámetros: Los resultados pueden ser sensibles a los valores de *Eps* y *MinPts*.
- Densidad variable: Dificultad para identificar clústeres en conjuntos de datos con densidad variable.
- Desempeño en altas dimensiones: Menos efectivo en datos de alta dimensión debido a la "maldición de la dimensionalidad".

2.3 Elección del número de clústeres

Como se ha mencionado anteriormente la elección del hiperparámetro *K*, numero de clústeres, es muy importante en algunos métodos como K-means. Es por ello que hemos elegido los dos métodos más conocidos y usados para comparar sus resultados, eligiendo así *K* de la manera más precisa**.**

2.3.1 Método del codo

El método del codo es una técnica visual que se utiliza para encontrar el número óptimo de clústeres en un conjunto de datos. Este método se basa en la suma de las distancias cuadradas dentro del clúster (*Within-Cluster Sum of Squares*, WCSS) para diferentes valores de *K* (Cui, 2020).

El objetivo es buscar el punto, en un gráfico de WCSS contra *K,* donde la disminución de WCSS comienza a ralentizarse, formando un "codo". Este punto sugiere que agregar más clústeres no proporciona una mejora significativa en la compactación de los clústeres. El valor de *K* correspondiente a este punto de inflexión se considera como el número óptimo de clústeres (Cui, 2020).

2.3.2 Método del análisis de la silueta

El análisis de la silueta mide cuán similares son los puntos dentro de un clúster en comparación con los puntos de otros clústeres. La puntuación de la silueta es una métrica que oscila entre -1 y 1, donde los valores más altos indican que los puntos están bien agrupados (scikit-learn, 2024).

El número óptimo de clústeres es aquel que maximiza la puntuación media de la silueta. Los coeficientes de la silueta cerca de +1 indican que la muestra está lejos de los clústeres vecinos. Un valor de 0 indica que la muestra está en o muy cerca del límite de decisión entre dos clústeres vecinos y los valores negativos indican que esas muestras podrían haberse asignado al clúster incorrecto (scikit-learn, 2024).

2.4 *Clustering* en pacientes con sepsis

La sepsis es una enfermedad que tiene múltiples etiologías y estados fisiopatológicos. Es por ello por lo que identificar a que grupo o pertenece cada paciente es fundamental para realizar un correcto diagnóstico y tratamiento. La Tabla 2.1 muestra las principales características de los estudios encontrados en la búsqueda bibliográfica de *clustering* en pacientes con sepsis.

Artículo	**Sujetos**	**Características de entrada al algoritmo**	**S/NS**	**Técnica de *clustering***	**Número de clústeres**	**Objetivo**
Papin et al. (2021)	6,046 pacientes con sepsis	Las primeras 52 dimensiones resultantes del análisis de correspondencias múltiples de 63 variables clínicas. Que explican al menos el 90% de la variabilidad total	NS	HC	6	Identificar clústeres de pacientes con sepsis basados en características clínicas y biológicas disponibles al ingreso en la UCI.
Jang et al. (2022)	2,490 pacientes con sepsis y 16,916 controles sanos	Edad, WBC, NLR, Hb, PLT, DNI, MPXI	NS	K-means y silueta	3 y 4	Identificar factores de riesgo robustos para la sepsis mediante *clustering*.
Maslove et al. (2012)	126 pacientes, 365 genes	Perfiles de expresión génica de neutrófilos	NS	PAM con silueta y HC	2	Identificar subtipos moleculares de sepsis basados en patrones de expresión génica.
Zhang et al. (2020)	685 pacientes adultos con sepsis	50 características representativas de perfil transcriptómico seleccionadas por su relación con la mortalidad	S	K-means con codo y silueta	2	Identificar subgrupos de pacientes con sepsis y desarrollar un modelo predictivo basado en características genéticas.

Tabla 2.1 *Resumen revisión bibliográfica de clustering en pacientes con sepsis. Abreviaturas: S, supervisado; NS, no supervisado. Abreviaturas: WBC, White Blood Cell; NLR, Neutrophil–Lymphocyte Ratio; Hb, HemogloBine; PLT, PLaTelet count; DNI, Delta Neutrophil Index; MPXI, Myeloperoxidase Index.*

2.4.1 *Clustering* Basado en Datos Clínicos y Demográficos

Por un lado, nos hemos encontrado enfoques de *clustering* que utilizan información como la edad, sexo, antecedentes médicos, comorbilidades, signos vitales y resultados de laboratorio para agrupar a los pacientes. Estos enfoques permiten detectar diferentes niveles de riesgo de mortalidad o variaciones en la respuesta al tratamiento. Esto, a su vez, facilita la personalización del cuidado médico, adaptándolo a las características específicas de cada grupo de pacientes. Dentro de la búsqueda bibliográfica realizada, destacan los siguientes estudios:

- El estudio de Papin et al. (2021) identificó seis clústeres distintos entre pacientes con sepsis usando datos clínicos y biológicos recolectados al ingreso en UCI, incluyendo trastornos subyacentes, fuentes de infección y microorganismos causantes. Utilizando un análisis retrospectivo de la cohorte OUTCOMEREA y *clustering* jerárquico, se compararon las tasas de mortalidad a 28 días, 90 días y un año, ajustando por la puntuación SOFA y el año de ingreso. Los clústeres mostraron características únicas y resultados significativamente diferentes en términos de mortalidad.
- El estudio de Jang et al. (2022) identificó factores de riesgo robustos para la sepsis utilizando K-means con datos clínicos y de laboratorio de sepsis y de controles. Se agruparon en tres y cuatro clústeres basados en siete marcadores (Edad, WBC, NLR, Hb, PLT, DNI y MPXI), revelando que mientras el recuento de glóbulos blancos (WBC, *White Blood Counts*) no mostró una asociación significativa con la sepsis en grupos de edad avanzada, los marcadores NLR y DNI fueron predictores robustos. El análisis de clúster permitió identificar eficientemente predictores de sepsis, contribuyendo a detectar pacientes potenciales que podrían ser pasados por alto.

Estos estudios, al obtener clústeres formados por diversos factores como la edad, comorbilidades, tipo de infección y estado inmunológico, muestran que:

- Las características al ingreso pueden predecir diferentes resultados clínicos en pacientes con sepsis.
- La identificación de clústeres específicos puede mejorar la gestión del cuidado al permitir una mayor personalización del tratamiento.
- La consideración de estos clústeres en futuras investigaciones puede aumentar la homogeneidad de los estudios, mejorando la precisión y relevancia de los hallazgos.

2.4.2 *Clustering* Basado en Datos Genómicos

El *clustering* basado en datos genómicos utiliza principalmente datos de expresión génica para identificar subtipos moleculares de enfermedades como la sepsis, (Maslove et al., 2012). Este enfoque se basa en analizar perfiles de expresión génica obtenidos de muestras biológicas, como sangre o tejidos, para identificar patrones específicos que distinguen diferentes subgrupos de pacientes. Este método puede revelar variaciones en la fisiopatología subyacente que no son evidentes a través de parámetros clínicos tradicionales, permitiendo una clasificación más precisa y personalizada de los pacientes, (Maslove et al., 2012). Dentro de la búsqueda bibliográfica realizada se encuentran los siguientes estudios:

- El estudio de Maslove et al. (2012) identificó subtipos moleculares de sepsis a partir de perfiles de expresión génica de neutrófilos de pacientes adultos sépticos mediante *clustering* alrededor de medoides (PAM) y *clustering* jerárquico. Se identificaron dos subtipos: el subtipo 1 mostró una mayor expresión de genes relacionados con vías de señalización inflamatorias y receptores Toll, y una mayor prevalencia de sepsis severa. Además, se observaron diferencias en la expresión de farmacogenes relevantes para tratamientos con hidrocortisona, vasopresina, norepinefrina y drotrecogina alfa.

- El estudio de Zhang et al. (2020) utilizó el algoritmo de *clustering* K-means para identificar subgrupos de pacientes con sepsis, empleando 50 características representativas seleccionadas por su relación con la mortalidad. Asimismo, determinaron el número óptimo de clústeres mediante los métodos del Codo y del Coeficiente de Silueta Promedio, confirmando este número con un enfoque basado en Monte Carlo. En el dataset GSE65682, se identificaron dos clústeres distintos de sepsis con características clínicas y biológicas únicas, y diferencias significativas en mortalidad y respuesta al tratamiento con hidrocortisona. Además, se desarrolló un modelo basado en 5 genes para predecir la pertenencia a estos grupos, facilitando un tratamiento más personalizado y mejorando la gestión clínica de la sepsis.

  El *clustering* basado en datos genómicos ofrece múltiples ventajas en el manejo de la sepsis. Primero, mejora la precisión diagnóstica al identificar subtipos biológicos que los métodos clínicos tradicionales no pueden diferenciar (Seymour et al., 2019). Segundo, proporciona una base para el desarrollo de tratamientos personalizados, optimizando las intervenciones terapéuticas según el perfil molecular de cada paciente (Sarma et al., 2020). Tercero, facilita la comprensión de las vías patogénicas y los mecanismos biológicos subyacentes a la sepsis, lo que puede conducir a la identificación de nuevos objetivos terapéuticos (van der Poll & Opal, 2008). Además, este enfoque permite una mejor evaluación y selección de pacientes para ensayos clínicos, lo que puede mejorar la calidad y la interpretabilidad de los resultados de dichos estudios. Al identificar los subtipos de sepsis que responden mejor a ciertos tratamientos, se pueden diseñar estudios más focalizados y efectivos, incrementando la probabilidad de éxito en el desarrollo de nuevas terapias.

2.5 *Clustering* de SHAP *values*

SHAP de Lundberg y Lee (Lundberg & Lee, 2017), es un método para explicar predicciones individuales. SHAP se basa en los Shapley *values* derivados de la teoría de juegos. El objetivo de SHAP es explicar la predicción de una instancia x calculando la contribución de cada característica a dicha predicción (Christoph Molnar, 2024). Este método de explicación computa los valores de Shapley desde la teoría de juegos cooperativos. En este contexto, los valores de las características de una instancia de datos actúan como jugadores en una coalición, y los valores de Shapley nos indican cómo distribuir de manera justa el "*payout*", la predicción, entre estas características (Christoph Molnar, 2024).

En SHAP, los valores de Shapley se interpretan como un método aditivo de atribución de características, es decir, como un modelo lineal (Christoph Molnar, 2024). En el trabajo previo realizado por López Herrero et al. (2024), se ha aplicado el método Deep SHAP para obtener los SHAP *values* y así medir, para cada paciente, la contribución de cada SNP a la predicción de la sepsis.

El aspecto más importante de utilizar *clustering* con SHAP *values* es la capacidad de identificar patrones y agrupar pacientes de manera precisa basándose en la contribución específica de cada SNP a la predicción (en nuestro caso, la sepsis) (Christoph Molnar, 2024; López Herrero et al., 2024). Esto permite una interpretación clara y detallada de cómo cada característica genética influye en el desarrollo de la enfermedad, facilitando la toma de decisiones clínicas personalizadas y el diseño de tratamientos más efectivos (Johnsen et al., 2021). Además, al utilizar los SHAP *values*, se asegura que los clústeres formados reflejen fielmente las influencias de los SNPs, proporcionando una base sólida y explicable para la segmentación y análisis de los datos genéticos (Cooper et al., 2021).

Este enfoque es también un aspecto innovador, ya que no se había hecho nunca *clustering* de SHAP *values* de SNPs en sepsis. Esto representa un avance significativo en la investigación

genética y médica, ofreciendo nuevas perspectivas y metodologías para el análisis de datos complejos y la comprensión de enfermedades como la sepsis.

La Tabla 2.2 muestra las principales características de los estudios encontrados en la búsqueda bibliográfica de artículos de *clustering* de SHAP *values* para adquirir ideas y conocimiento de cómo aplicarlo a nuestro caso:

- El estudio de Cooper et al. (2021) presenta una metodología innovadora para identificar subgrupos de síntomas en pacientes con COVID-19, utilizando SHAP *values* y *clustering* basado en densidad con HDBSCAN. El objetivo es mejorar la interpretación y la acción clínica mediante la identificación de 16 presentaciones distintas de síntomas entre 2,479 pacientes positivos. Utilizando una combinación de SHAP *values* para la atribución de características y reducción de dimensionalidad con UMAP, HDBSCAN realizó el *clustering* en el espacio de embedding generado. La optimización de los hiperparámetros de HDBSCAN mediante una búsqueda en cuadrícula permitió una asignación adecuada de instancias a los clústeres, minimizando las no asignadas. Este enfoque facilita la comprensión de la variabilidad clínica del COVID-19 y sugiere investigaciones futuras sobre factores demográficos y clínicos comunes dentro de cada grupo y su relación con los resultados clínicos.

- El estudio de Rodrigo Queirós Conceição (2023) aborda los desafíos del análisis de grandes volúmenes de datos utilizando un enfoque innovador que combina modelos de caja negra con SHAP *values* para identificar relaciones complejas y clústeres interpretables en la detección de malware. La investigación destaca la importancia de seleccionar el algoritmo de *clustering* más adecuado, comparando K-means, DBSCAN y GMM, y utiliza etiquetas conocidas y SHAP *values* para evaluar y mejorar la comprensión de los clústeres. El estudio demuestra que este enfoque no solo supera las limitaciones de métodos tradicionales, sino que también ofrece una visión detallada de cómo las variables influyen en la agrupación de datos, facilitando una toma de decisiones más informada.

Como conclusión de la búsqueda, podemos destacar la importancia de utilizar SHAP *values* en conjunción con análisis de *clustering* para obtener perspectivas adicionales sobre cómo las variables afectan a diferentes grupos de datos. Esta metodología no solo facilita la interpretación de los clústeres, sino que también proporciona información valiosa para la toma de decisiones, como la capacidad de realizar un análisis de SHAP *values* dentro del clúster para entender mejor las asociaciones clínicas del modelo. Todo esto se va a tener muy en cuenta a la hora de seleccionar el algoritmo de *clustering* más adecuado en nuestro caso.

Artículo	**Sujetos**	**Características de entrada al algoritmo**	**S/NS**	**Técnica de *clustering***	**Número de clústeres**	**Objetivo**
Cooper et al. (2021)	4063 pacientes de COVID-19	SHAP *values* de 21 síntomas	S	HDBSCAN	16	Identificar y caracterizar diferentes presentaciones de síntomas de COVID-19.
Rodrigo Queirós Conceição (2023)	Comparación de métodos de *clustering* para SHAP *values*: K-means, DBSCAN y GMM para la detección de *malware*.					

Tabla 2.2 *Resumen revisión bibliográfica de clustering con SHAP values. Abreviaturas: S, supervisado; NS, no supervisado.*

2.6 Elección y comparación del método a implementar

La elección del algoritmo de *clustering* más adecuado es fundamental para el análisis y segmentación de datos. Cada algoritmo tiene sus propias características y se adapta de manera diferente según el tipo de datos y el objetivo del análisis (RODRIGO QUEIRÓS CONCEIÇÃO, 2023). En este contexto, se han escogido varios algoritmos para determinar cuál se ajusta mejor a las necesidades del estudio. Entre estos algoritmos, K-means ha demostrado ser una opción robusta y eficiente para la segmentación de datos en muchas aplicaciones. K-means se destaca por su eficiencia, simplicidad y capacidad de interpretación, lo que lo convierte en una opción preferida en muchos contextos (Francisco Sanz, 2024; Sinaga & Yang, 2020). En nuestro caso específico, donde trabajamos con SHAP *values* de SNPs, K-means es especialmente útil debido a las siguientes razones:

- Eficiencia computacional: K-means es rápido y escalable, adecuado para trabajar con grandes volúmenes de datos como los SHAP *values* generados a partir de numerosos SNPs.
- Simplicidad de implementación: Su algoritmo sencillo permite una fácil implementación y comprensión de los resultados, lo cual es crucial en estudios genéticos donde la interpretabilidad es importante.

Además, determinar el número óptimo de clústeres es crucial en K-means. En nuestro análisis, se utilizan tanto el método del codo como el análisis de la silueta. Ambos métodos nos podrán proporcionar información valiosa sobre la estructura de nuestros datos y ayudarán a garantizar que el número de clústeres elegido refleje adecuadamente los patrones presentes en el conjunto de datos. Este enfoque multifacético garantiza que nuestra elección estuviera bien fundamentada.

Con el objetivo de realizar un análisis más robusto, también se ha aplicado DBSCAN para explorar sus capacidades y compararlas con las de K-means. DBSCAN ofrece varias ventajas adicionales:

- Identificación de clústeres de forma arbitraria: A diferencia de K-means, DBSCAN puede identificar clústeres con formas no esféricas, lo cual es útil en datos con estructuras complejas.
- Manejo del ruido: DBSCAN maneja eficientemente el ruido y los puntos atípicos, lo que puede ser esencial en análisis de datos genéticos donde existen variaciones significativas.
- Sin necesidad de especificar el número de clústeres: DBSCAN no requiere la especificación previa del número de clústeres, lo cual puede ser ventajoso cuando la estructura de los datos no es clara.

Por tanto, el uso de DBSCAN, además de K-means, nos permite explorar diferentes perspectivas y asegurarnos de que la segmentación de los datos fuera robusta y precisa. Estos métodos de *clustering*, además de por sus ventajas inherentes, han sido elegidos debido a su uso extendido y recomendado en la bibliografía consultada. Así, se espera que los clústeres identificados proporcionaran información valiosa y accionable para la toma de decisiones basada en los SHAP *values*.

Capítulo 3. Materiales y métodos

3.1 Población bajo estudio

Este TFG se ha llevado a cabo a partir de los resultados obtenidos en un estudio previo (López Herrero et al., 2024). Este llevó a cabo un estudio de cohorte prospectivo que incluyó dos cohortes: una con 750 pacientes con sepsis (GenoSEPSIS) y otra con 3,500 controles poblacionales del Banco Nacional de ADN (BNADN). La Tabla 3.1 muestra las características demográficas de estos sujetos. Los pacientes sépticos representaron el 17.6% (750 pacientes) y tuvieron una mediana de edad más alta en comparación con los controles poblacionales [mediana (rango intercuartil): 72 (61-78) vs. 47 (41-54) años, *p*-valor<0.001] y una mayor proporción de hombres [65.9% vs. 54.4%, p-valor<0.001].

3.1.1 *GenoSEPSIS*

La cohorte de pacientes, GenoSEPSIS, incluye a pacientes adultos que se sometieron a cirugía mayor y fueron admitidos en la unidad de reanimación del Hospital Clínico Universitario de Valladolid y del Hospital Clínico Universitario de Santiago en España (López Herrero et al., 2024). Estos pacientes fueron diagnosticados con sepsis (n=121) o shock séptico (n=629) según las definiciones de SEPSIS-3 (Singer et al., 2016).

Este estudio se realizó de acuerdo con la legislación española vigente sobre investigación biomédica y la Declaración de Helsinki. Además, se obtuvo el consentimiento informado por escrito de todos los participantes o sus representantes, y los Comités de Ética de Investigación de todos los centros participantes aprobaron este estudio (número de aprobación: PI 20-2070).

En la Tabla 3.2 podemos observar las características basales de los sujetos con sepsis en estudio. Se observa que el 83.9% (n=629) tuvo shock séptico y su tasa de mortalidad asociada a 90 días fue del 42.7% (n=320). La distribución de las puntuaciones SOFA y APACHE II fueron 9 (7-11) y 18 (15-22), respectivamente. Un total de 561 pacientes (74.8%) tenían una o varias comorbilidades asociadas, entre ellas enfermedad cardiovascular crónica (257 casos, 34.3%), enfermedad respiratoria crónica (156 casos, 20.8%), hipertensión arterial (318 casos, 42.4%), enfermedad crónica insuficiencia renal (89 casos, 11.8%), insuficiencia hepática crónica (43 casos, 5.7%), diabetes mellitus (166 casos, 22.1%), obesidad (109 casos, 14.5%) e inmunosupresión (73 casos, 9.7%).

	Casos (GenoSEPSIS)	**Controles (BNADN)**	***p*-valor**
N°. (%) datos*	750 (17.6)	3,500 (82.4)	
Genero [n (%)]			<0.001
Mujer	251 (33.5)	1,597 (45.7)	
Hombre	494 (65.9)	1,903 (54.4)	
Género desconocido	5 (0.7)	0 (0.0)	
Edad			<0.001
Años en el momento de la toma de la muestra [años, mediana (rango intercuartil)]	72 (61-78)	47 (41-54)	
Edad desconocida [n (%)]	0 (0.0)	68 (1.9)	

Tabla 3.1 *Características demográficas de los pacientes con sepsis y los controles utilizados en este estudio.*

	Cohorte de Sepsis
N°. (%) datos*	750 (100)
Género [n (%)]	
Mujer	251 (33.5)
Hombre	494 (65.9)
Género desconocido	5 (0.7)
Edad, [años, mediana [RIC]]	
Años en el momento de la toma de la muestra	72 (61-78)
Edad desconocida	0 (0.0)
Comorbilidades [n (%)]	
Enfermedad cardiovascular crónica	257 (34.3)
Enfermedad respiratoria crónica	156 (20.7)
Hipertensión	318 (42.4)
Fallo renal crónico	89 (11.9)
Insuficiencia hepática crónica	43 (5.7)
Diabetes mellitus	166 (22.1)
Obesidad	109 (14.5)
Inmunosupresión	73 (9.7)
Mediciones al diagnóstico mediana [RIC]	
Creatinina (mg/dl)	1.7 (1.1-3.1)
Glóbulos blancos (células/mm^3)	12,685 (7,630-18,290)
Linfocitos (células/mm^3)	7.1 (4.5-11.6)
Neutrófilos (células/mm^3)	87.5 (81.6-91.3)
Puntuación SOFA	9 (7-11)
Puntuación APACHE II	18 (15-22)
Evolución temporal y resultados hospitalarios.	
Duración de la estancia hospitalaria [días, mediana (RIC)]	31 (19-44)
Duración de la estancia en UCI [días, mediana (RIC)]	14 (7-21)
Duración de la ventilación mecánica [días, mediana (RIC)]	9 (2-16)
Sepsis [n (%)]	121 (16.1)
Shock séptico [n (%)]	629 (83.9)
Mortalidad a los 90 días [n (%)]	320 (42.7)

Tabla 3.2 *Características basales de los sujetos con sepsis bajo estudio.*

3.1.2 *BNADN*

La cohorte de control, BNADN, se obtuvo del Banco Nacional de ADN Carlos III, de la Universidad de Salamanca, España. Los sujetos de la población del BNADN eran individuos sanos no relacionados, distribuidos uniformemente a lo largo de las diferentes áreas geográficas de España (López Herrero et al., 2024).

3.2 Genotipado y preprocesado

Para el articulo referencia se obtuvo ADN genómico a partir de sangre periférica y se aisló utilizando el kit Chemagic DNA Blood 100 (PerkinElmer Chemagen Technologies GmbH), siguiendo para ello las instrucciones del fabricante. Las muestras se genotiparon de manera conjunta para las dos cohortes (GenoSEPSIS y BNADN) utilizando el Axiom Spain Biobank Array (Thermo Fisher Scientific) en el nodo de Santiago de Compostela del Centro Nacional de Genotipado (CeGen-ISCIII). Este array contiene aproximadamente 781,759 sondas para genotipar un total de 756,834 SNPs (López Herrero et al., 2024).

A continuación, en López Herrero et al. (2024) se llevó a cabo un procedimiento de control de calidad con PLINK 1.9 tanto en las muestras como en los SNPs genotipados. Se excluyeron las variantes con una frecuencia alélica menor (*minor allele frequency*, MAF) <1%, una tasa de llamada <98%, y los marcadores que se desviaban significativamente del equilibrio de Hardy-Weinberg ($p<10^{-6}$) con ajuste mid-p. Además, se evaluó el exceso de heterocigosidad para eliminar posibles contaminaciones entre muestras, y se filtraron aquellas muestras con más del 2% de variantes faltantes. Posteriormente, se retuvieron los SNPs autosómicos y se eliminaron

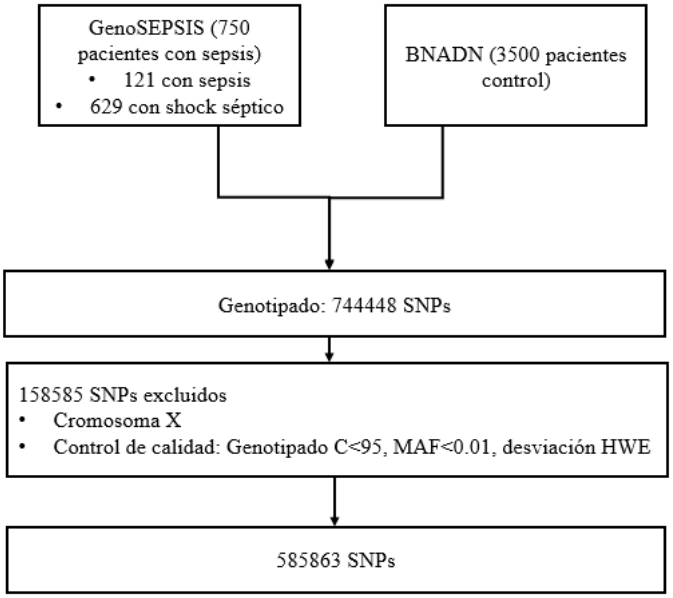

Figura 3.1 *Población de estudio.*

las regiones con alto desequilibrio de ligamiento utilizando el *pruning* de LD (1,000 SNPs por ventana, tamaño de paso de 80 y umbral de r2 de 0.1) para evaluar las relaciones y estimar las proporciones de ascendencia global. Se evaluaron las relaciones de parentesco mediante puntuaciones de identidad por descendencia, y se eliminó un individuo de cada par con PI_HAT>0.25 que mostraba un patrón coherente de Z0, Z1 y Z2 (según los valores teóricos esperados para cada nivel de parentesco). Las componentes principales genéticas se calcularon con PLINK utilizando el subconjunto de variantes podadas por LD (López Herrero et al., 2024).

Después de los análisis de control de calidad, se obtuvieron un total de 585,863 SNPs (ver Figura 3.1). Los análisis de asociación se realizaron con PLINK 1.9, ajustando por edad, sexo y los dos principales componentes principales. Así, se seleccionaron diferentes subconjuntos de SNPs relevantes según distintos umbrales de *p*-valor: 5 x 10^{-2} (33,597 SNPs), 5 x 10^{-3} (3,761 SNPs), 5 x 10^{-4} (495 SNPs), 5 x 10^{-5} (100 SNPs), 5 x 10^{-6} (37 SNPs), 5 x 10^{-7} (30 SNPs), 5 x 10^{-8} (28 SNPs) (López Herrero et al., 2024).

3.3 Explainable Artificial Intelligence

En el estudio desarrollado por (López Herrero et al., 2024), se aplicó una metodología de XAI que consta de dos pasos principales:

- **Predicción de la sepsis.** Primero, se diseñó y evaluó un modelo de DL basado en una red neuronal convolucional (*convolutional neural network*, CNN) para predecir la sepsis utilizando distintos subconjuntos de SNPs relevantes. Para ello, se dividieron los datos en conjuntos de entrenamiento (50%), validación (25%) y test (25%). La metodología propuesta mostró un alto rendimiento en la predicción de sepsis. Concretamente, el modelo entrenado con SNPs con *p*-valor < 5x10^{-3} (3,761 SNPs) obtuvo los mejores resultados, con alcanzó una precisión del 96.4%, un área bajo la curva ROC de 0.985, una sensibilidad del 85.6% y una especificidad del 98.7% en el conjunto de test.

- **Identificación de SNPs relacionados con la sepsis.** A partir del modelo CNN anterior, se aplicó el método Deep SHAP, que proporciona, para cada paciente, la contribución de cada SNP a la predicción de sepsis. De este modo, los SNPs más importantes se determinaron

promediando los SHAP *values* de los pacientes correctamente predichos con sepsis. Concretamente, se identificaron los 20 SNPs más importantes para la predicción de sepsis como aquellos con un mayor SHAP *value,* Figura 3.2, destacando tres SNPs: rs17653532 (SHAP = 0.054), un SNP intrónico en el gen PRIM2; rs1575081785 (SHAP = 0.050), un SNP missense en el gen RBSN; y rs74707084 (SHAP = 0.049), un SNP intrónico en el gen SYNPR (López Herrero et al., 2024).

La Figura 3.3 presenta diagramas de los SHAP *values* para 4 sujetos distintos. En esta figura se puede observar que en cada sujeto predominan SNPs diferentes, lo que indica una variabilidad significativa en la importancia de estos polimorfismos entre los individuos. Esta variabilidad sugiere que los perfiles genéticos de los pacientes con sepsis son heterogéneos, lo que motiva la necesidad de aplicar técnicas de *clustering* de este TFG. El objetivo es identificar subgrupos de pacientes con sepsis con características genéticas similares, lo cual podría tener implicaciones importantes para la personalización de tratamientos y el entendimiento de la predisposición genética a diversas condiciones.

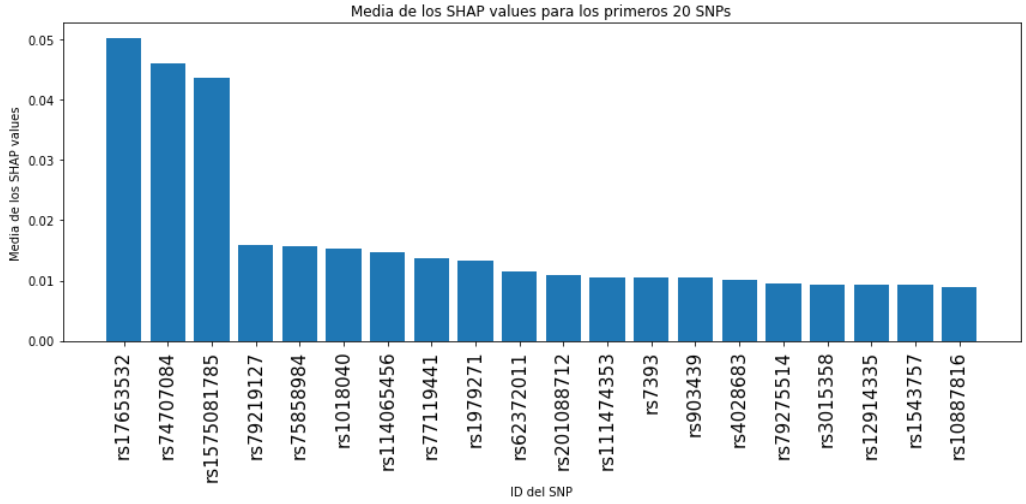

Figura 3.2 *Media de los SHAP values para los primeros 20 SNPs.*

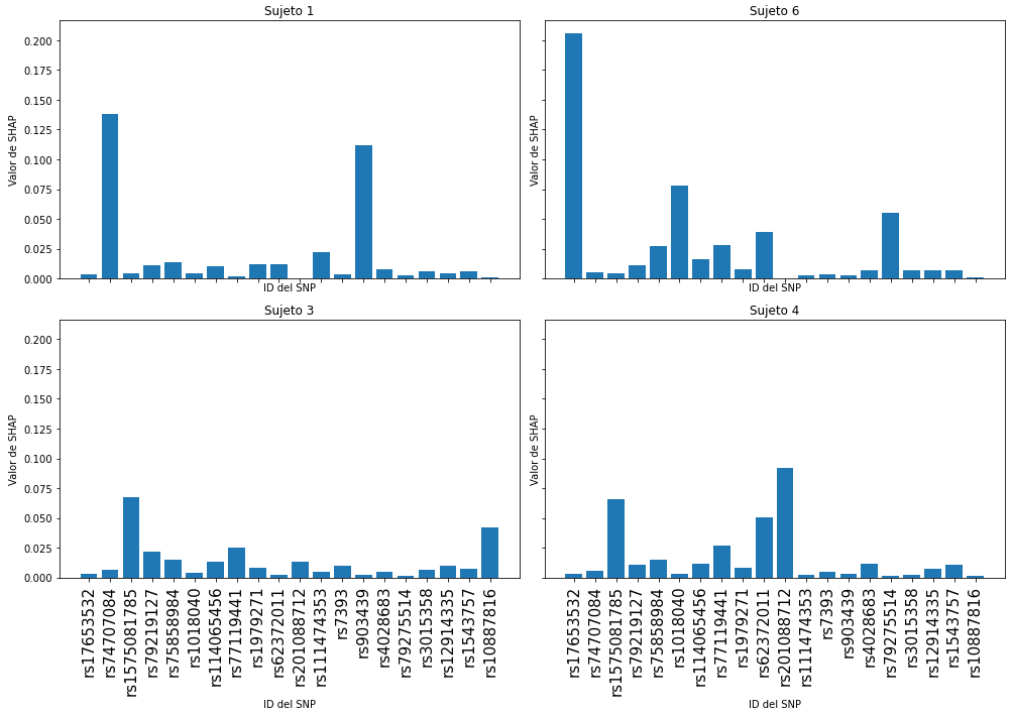

Figura 3.3 *SHAP values de los SNPs en 4 sujetos distintos.*

3.4 *Clustering*

En este TFG se va a realizar un análisis multifacético de dos técnicas de *clustering*, K-means y DBSCAN (Stewart & Al-Khassaweneh, 2022), comparando sus resultados con el objetivo de obtener la mayor fiabilidad y precisión posibles. Los algoritmos de *clustering* se han implementado en Python, y en concreto con las librerías numpy, matplotlib y sklearn. Estar familiarizado con estas librerías y el lenguaje de programación Python ha sido fundamental para una obtención e interpretación correcta de los resultados.

El *clustering* ha partido de la matriz de SHAP *values* obtenida, cuyas dimensiones son el número de sujetos de test con sepsis (n=187) y el número de SNPs (3,761). Se probó con el top 20 pero los resultados no variaban, por lo que se decidió coger todos. Antes de aplicar los algoritmos de *clustering* se ha aplicado PCA (*Principal Component Analysis*) a los datos. PCA es una técnica estadística ampliamente utilizada para la reducción de dimensionalidad de manera no supervisada. PCA transforma los datos originales en un nuevo conjunto de variables, llamadas componentes principales, que son combinaciones lineales de las variables originales. Las componentes principales son ortogonales entre sí, lo que significa que no están correlacionadas y capturan la máxima varianza posible de los datos en una nueva dimensión (Ding chqding, 2004). PCA puede mejorar la efectividad del *clustering* al reducir la dimensionalidad de los datos y, al mismo tiempo, maximizar la varianza explicada, lo que lleva a una mejor representación y separación de los datos en diferentes clústeres (Ding chqding, 2004).

3.4.1 K-means

Partiendo únicamente del número de clústeres (*K*) y los datos de entrada (matriz de SHAP *values*), K-means asigna a cada sujeto a un clúster mediante un procedimiento que consta de los siguientes pasos (Francisco Sanz, 2024):

1. Elección del número de clústeres *K*.
2. Inicialización aleatoria de centroides. Los centroides son puntos que representan el centro de cada clúster.
3. Asignación de los datos a los clústeres cuyo centroide este más cercano. Normalmente se emplea la distancia euclidiana (Lifeder, 2019):

$$d(x_i, c_j) = \sqrt{\sum_{k=1}^{N} (x_{ik} - c_{jk})^2} \quad (1)$$

   siendo x_i el dato, c_j el centroide del clúster *j*, y *n* es el número de dimensiones
4. Una vez que todos los puntos de datos se han asignado a clústeres, se recalculan los centroides de cada clúster. El nuevo centroide de un clúster es el promedio de todos los puntos de datos que pertenecen a ese clúster.
5. Se repiten los pasos 3 y 4 hasta alcanzar la convergencia. Se alcanza cuando tras un cierto número de iteraciones los centroides dejan de cambiar, los datos puntos dejan de cambiar de clúster o se alcanza el límite de iteraciones establecido. Así, todos los datos serán asignados a uno de los *K* clústeres.

En este TFG, el algoritmo K-means ha sido aplicado mediante la liberaría sklearn. Un aspecto importante en la aplicación del algoritmo K-means es determinar el número de clústeres, *K*, de forma que refleje adecuadamente los patrones presentes en el conjunto de datos. Para ello, se han aplicado dos técnicas, codo y silueta:

- Método del codo: Se ha implementado el método del codo para determinar el número óptimo de clústeres en K-means. Primero, se define una función que calcula las distorsiones para diferentes números de clústeres. Esta función toma como entrada el conjunto de datos y el número máximo de clústeres a considerar. Para cada número de clústeres, se crea y ajusta un modelo de K-means, calculando la inercia del modelo ajustado, que es una medida de la suma

de las distancias cuadradas dentro de cada clúster. Estas inercias se almacenan en una lista. Luego, se aplica esta función al conjunto de datos, obteniendo una lista de distorsiones para cada número de clústeres desde 1 hasta el número máximo especificado. Finalmente, se representa en una gráfica la relación entre el número de clústeres y la distorsión correspondiente, permitiendo así visualizar el "codo" en el gráfico, Figura 3.4. El punto donde se produce un cambio notable en la pendiente de la curva (el codo) indica el número óptimo de clústeres para el conjunto de datos. A la hora de implementar este método, así como interpretar los resultados, se ha tomado como referencia la información suministrada en (Cui, 2020). De este modo, el número óptimo de clústeres se ha obtenido como el valor de *K* en el "codo", es decir, el punto después del cual la distorsión/inercia comienza a disminuir de forma lineal. Este método se aplica de la siguiente manera (Cui, 2020):

1. **Ejecución de K-means para diferentes valores de *K*:** Se ejecuta el algoritmo K-means varias veces, cada vez incrementando el número de clústeres (por ejemplo, de 1 a 10).
2. **Cálculo de WCSS:** Para cada valor de *K*, se calcula la suma de las distancias cuadradas dentro del clúster, que mide la compactación de los clústeres.
3. **Gráfico de WCSS contra K:** Se grafica el número de clústeres *K* en el eje x y el valor de WCSS en el eje y.

- Método de la silueta: El análisis se realiza para un rango de posibles números de clústeres (de 2 a 6 en este caso). Para cada número de clústeres, se ajusta un modelo de K-means y se calculan los coeficientes de silueta, que miden la cohesión dentro de los clústeres y la separación entre ellos. El gráfico resultante incluye dos subgráficos para cada número de clústeres: uno que muestra los coeficientes de silueta de los datos y otro que visualiza los clústeres formados. En la Figura 3.5 se puede ver un ejemplo de aplicación del método de la silueta con los dos subgráficos. En el subgráfico de silueta, los coeficientes de silueta para cada punto de datos se agrupan y se colorean por clúster, permitiendo evaluar la calidad de la agrupación. La línea roja discontinua indica la media de los coeficientes de silueta. El otro subgráfico muestra los datos agrupados en un espacio bidimensional, con los puntos coloreados según su clúster asignado y los centroides de los clústeres resaltados. La implementación del método del método y como interpretar los resultados se toma como referencia la información suministrada en la guía de la librería sklearn y en https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html. Este método se aplica de la siguiente manera (scikit-learn, 2024):

  1. **Ejecución de K-means para diferentes valores de *K*:** Se ejecuta el algoritmo K-means varias veces con diferentes valores de *K*.
  2. **Cálculo de la puntuación de la silueta:** Para cada valor de *K*, se calcula la puntuación media de la silueta de todos los puntos del conjunto de datos.
  3. **Gráfico de la puntuación de la silueta contra *K*:** Se grafica la puntuación media de la silueta en el eje y con el número de clústeres *K* en el eje x.

Tras la aplicación del método del codo y el de la silueta, se ha procedido a analizar los resultados de ambos métodos y elegir la opción más adecuada. Esta elección proporcionará un equilibrio adecuado entre la variabilidad explicada por los clústeres y la claridad de la estructura de los clústeres, asegurando una clasificación robusta y significativa de los datos.

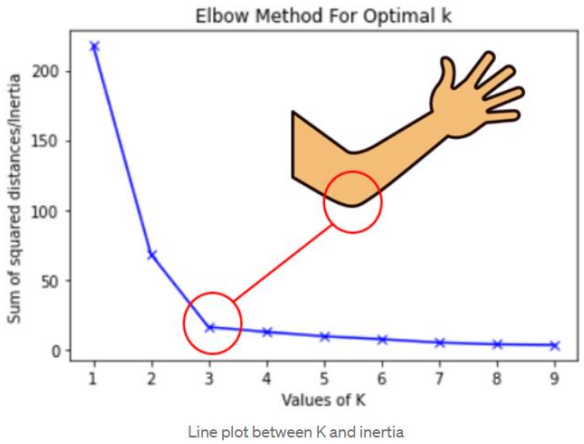

Figura 3.4 *Ejemplo método del codo. Imagen procedente de: https://www.linkedin.com/pulse/k-means-elbow-method-clustering-bogus%C5%82aw-konefa%C5%82/*

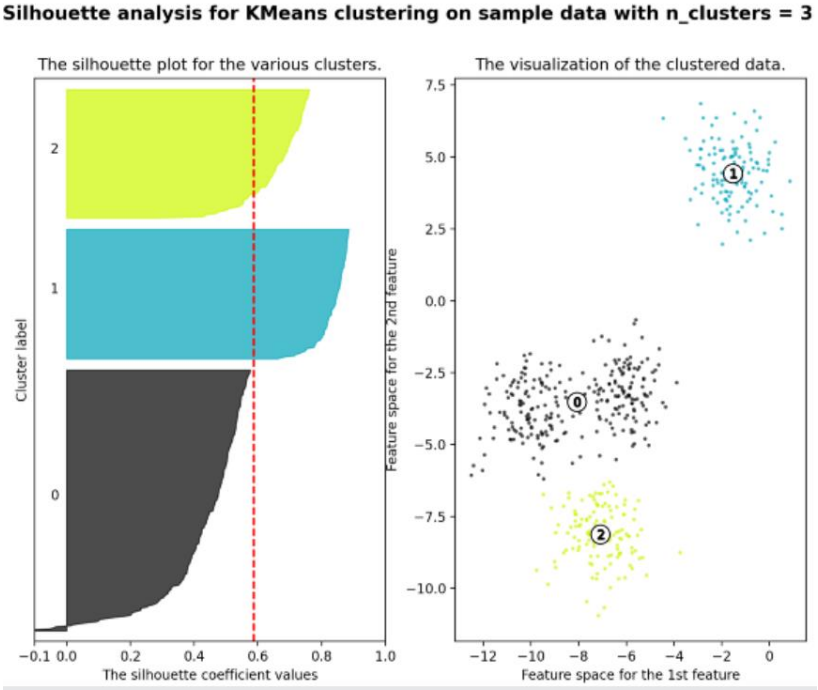

Figura 3.5 *Ejemplo método de la silueta. Imagen procedente de: https://aiplanet.com/learn/unsupervised-learning-es/analisis-y-tecnicas-de-clustering/1623/algoritmos-de-clustering-k-medias.*

3.4.2 DBSCAN

En DBSCAN, al igual que con K-means, hemos empleado la librería de Python sklearn. Sin embargo, en este caso no hay que especificar el número de clústeres, pero sí los hiperparámetros *Eps* (la distancia máxima entre dos puntos para que se consideren en el mismo clúster) y *minPts* (el número mínimo de puntos necesarios para formar un clúster). El algoritmo DBSCAN consta de los siguientes pasos (Deng, 2020):

1. Se define la *Eps*, en cada observación, se considera una vecindad de radio ε (epsilon) y se miran los puntos que están en ella.

2. Se identifican las observaciones centrales. Si una observación tiene al menos un cierto número de vecinos (*MinPts*) dentro de su ε-vecindad, incluida ella misma, se considera una observación central. Esto indica una región de alta densidad.

3. Todas las observaciones en la vecindad de una observación central pertenecen al mismo clúster. Si hay observaciones centrales cercanas entre sí, se conectan formando un único

clúster. De esta manera, se obtiene una secuencia de observaciones centrales conectadas que constituyen un clúster.

4. Cualquier observación que no sea una observación central y que no tenga ninguna observación central en su vecindad se considera una anomalía o ruido. Estos puntos no se asignan a ningún clúster.

3.5 Interpretación clínica y funcional

3.5.1 Análisis clínico

Cada clúster obtenido tiene unos sujetos con sus respectivos SHAP *values* de cada SNP. Teniendo esta información, se ha procedido a realizar un análisis clínico de los datos de cada clúster para identificar características comunes y diferencias estadísticamente significativas (p-valor < 0.05) entre los 3 grupos de pacientes y entre parejas de dichos clústeres (0 *vs*. 1, 0 *vs*. 2 y 1 *vs*. 2) desde un punto de vista clínico. Esto puede revelar patrones clínicos específicos que no serían evidentes sin el agrupamiento previo, validando así la utilidad del *clustering* y sugiriendo ajustes si los clústeres no reflejan diferencias clínicas claras. Además, identificar características clínicas comunes dentro de un clúster puede ayudar a personalizar tratamientos y estrategias de manejo para los pacientes de ese grupo específico.

Para realizar el análisis se han analizado las variables clínicas de los sujetos de la base de datos GenoSEPSIS (ver Tabla 3.2). El análisis es de los sujetos de cada clúster y antes de llevarse a cabo se realiza una revisión de la base de datos y se corrigen errores de esta o se trasforman algunas variables de manera que sea más fácil su interpretación. El análisis clínico se divide en 3 tipos según las variables:

- Variables clínicas. Se ha procedido a analizar diferencias entre clústeres en las siguientes variables: edad, sexo, tipo de sepsis, mortalidad a 90 días, neutrófilos, linfocitos, glóbulos blancos y creatinina. La variable de mortalidad a 90 días va a ser transformada a binario, de manera que si la variable en la base de datos es NAN (Not A Number) o es mayor de 91 días la variable es 0 y si no es 1. De esta manera vamos a diferenciar dos subtipos de variables, categóricas y continuas. Es importante diferenciarlas porque se las va a realizar un análisis estadístico distinto:
  - Categóricas (sexo, tipo de sepsis y mortalidad a 90 días): Se calcula la proporción de un valor especifico, en el caso del sexo, 1, que equivale a ser hombre; en el caso del tipo de sepsis, 3, que equivale a padecer Shock séptico; y en el caso de la mortalidad en 90 días, 1, si el paciente muere antes de alcanzar los 90 días. Además, se incluye en el análisis un p-valor, calculado para los 3 clústeres y para dichos clústeres por parejas, usando la métrica Chi-cuadrado de independencia de variables en una tabla de contingencia. Es elegida debido a que se aplica a variables categóricas de ≥2 grupos. Se calcula con la librería scipy.
  - Continuas (edad, neutrófilos, linfocitos, glóbulos blancos y creatinina): A diferencia de las no binarias, se calcula la mediana y el rango intercuartílico (percentil 25-percentil 75). El análisis del p-valor para los 3 clústeres se calcula con Kruskal-Wallis, que prueba la hipótesis nula de que la mediana poblacional de todos los grupos es igual. Es elegida debido a que se aplica a variables continuas de >2 grupos. Sin embargo, para el cálculo del p-valor de parejas de clústeres necesitamos emplear la prueba U de Mann-Whitney, ya que a diferencia de Chi-cuadrado, Kruskal-Wallis no se puede emplear para 2 clústeres. Mann-Whitney es una prueba no paramétrica que sirve para determinar si existe una diferencia significativa en la distribución de una variable entre los dos clústeres. Se calculan también con la librería scipy.

- Comorbilidades. Analizar las comorbilidades dentro de cada clúster permite identificar patrones de enfermedades coexistentes que son más frecuentes en ciertos grupos de pacientes. Esto puede ayudar a entender mejor el perfil de riesgo y la carga de enfermedad en cada clúster, lo que es crucial para el manejo y prevención de complicaciones de la sepsis. Concretamente, se ha procedido a analizar diferencias entre clústeres en las siguientes comorbilidades (ver Tabla 3.2): Enfermedad cardiovascular crónica, Enfermedad respiratoria crónica, Hipertensión, Fallo renal crónico, Fallo hepático crónico, Diabetes mellitus, Obesidad e Inmunosupresión. En este caso, se va a calcular la proporción de pacientes que presentan la comorbilidad, 1. Como en el caso de variables categóricas del caso anterior, se va a calcular esta proporción y se va a hacer un análisis del *p*-valor usando la métrica Chi-cuadrado de la librería scipy.
- Datos de hospitalización. Se ha procedido a analizar diferencias entre clústeres en las siguientes variables: días en UCI, días de estancia hospitalaria, días de ventilación mecánica, muerte en UCI. La variable de muerte en UCI no pertenece a la base datos, se va a crear. Es una variable binaria en la cual si hay un valor NAN en Días en UCI o Días en UCI ≠ Mortalidad en 90 días (sin binarizar) va a ser 0 y si no es 1. Para realizar el análisis del *p*-valor en días en UCI, días de estancia hospitalaria y días de ventilación mecánica como son continuas se va a emplear Kruskal-Wallis y Mann-Whitney, mientras que para Muerte en UCI se va a emplear Chi-cuadrado.

3.5.2 Análisis funcional

Cada clúster obtenido tiene unos sujetos con sus respectivos SHAP *values* de cada SNP. Teniendo esta información podemos analizar los SNPs más importantes en cada clúster haciendo la media de los SHAP *values* de los sujetos que pertenecen a ese clúster. Así, se han determinado los 5 SNPs en formato rs, más importantes de cada clúster. A partir de estos SNPs, se ha procedido a analizar las bases genéticas y funcionales que pueden estar contribuyendo a las diferencias observadas entre los grupos de pacientes. Esto puede revelar genes y vías biológicas importantes que están asociados con las características específicas de cada clúster.

Este análisis funcional se ha realizado mediante tres enfoques:

- Primero, se realiza un análisis general del rs en la página web del EMBL: Tras seleccionar Human en la búsqueda e introducir el rs, obtenemos la consecuencia funcional del SNP, su localización y la frecuencia del alelo menor en la población ibérica. (http://grch37.ensembl.org/index.html).
- Segundo, se busca información sobre el gen donde se encuentra el rs en la base de datos del NCBI: Obtenemos el gen o los genes más cercanos (si el SNP es intergénico). Esto nos va a proporcionar información sobre que funciones afecta el SNP (https://www.ncbi.nlm.nih.gov/snp/).
- Por último, se investigan las implicaciones biológicas del gen donde se encuentra el rs en la página web del Human Gene Database y Malacards. Esta búsqueda nos va a proporcionar información sobre el gen o genes seleccionados en el apartado anterior, tanto posibles enfermedades asociadas como vías metabólicas específicas en las que participan. (https://www.genecards.org/) y (https://www.malacards.org/). De manera general, se incidirá en el SNP más importante de cada clúster y el gen al que afecta. Por último, se relacionarán y compararán las implicaciones biológicas del gen principal del clúster con las conclusiones clínicas de dicho clúster con el resto.

Capítulo 4. Resultados

4.1 Identificación de clústeres relacionados con la sepsis

Tras haber explicado los materiales y métodos que se ha empleado en el TFG, se presentan los resultados obtenidos en la identificación de clústeres de sujetos de sepsis mediante K-means y DBSCAN.

4.1.1 K-means

El primer algoritmo de *clustering* que se ha aplicado es K-means. Como se mencionó en la metodología previamente, se comienza calculando el número optimo de clústeres *K* del algoritmo, empleando el método del codo y el método de la silueta.

Para determinar el número óptimo de clústeres con el método del codo, la Figura 4.1 muestra la distorsión con respecto al valor de *K*., En esta figura se observa el valor de *K* en el "codo", es decir, el punto después del cual la distorsión/inercia comienza a disminuir de forma lineal. En este caso, concluimos que el número óptimo de grupos estaría entre 3 y 4.

Con respecto al método de la silueta, hay que analizar los resultados para cada valor de *K*. Para ello, se evalúa la media de los coeficientes de la silueta (ver Tabla 4.1), así como la representación gráfica correspondiente de cada *K*.

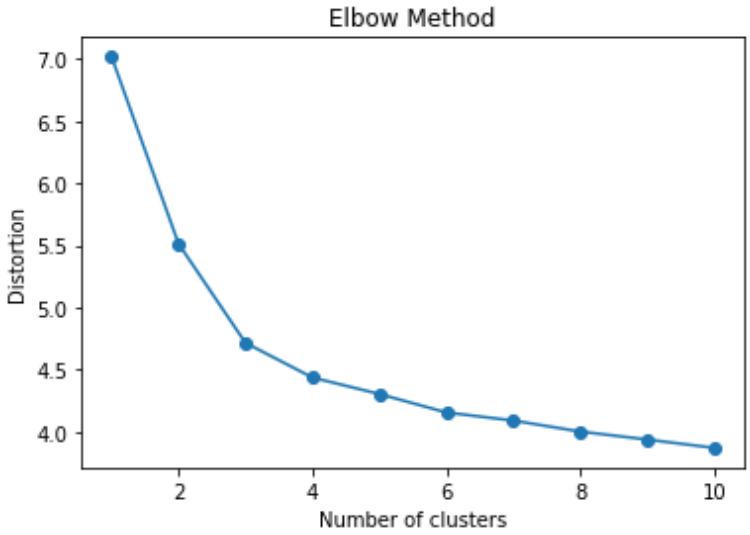

Figura 4.1 *Resultados método del codo en K-means.*

Número de clústeres	**Media de los coeficientes de silueta**
2	0.25147509085156766
3	0.19606539591503333
4	0.1142657637263336
5	0.12594841486976072
6	0.1273444752328521

Tabla 4.1 *Media de los coeficientes de silueta para cada número de clústeres.*

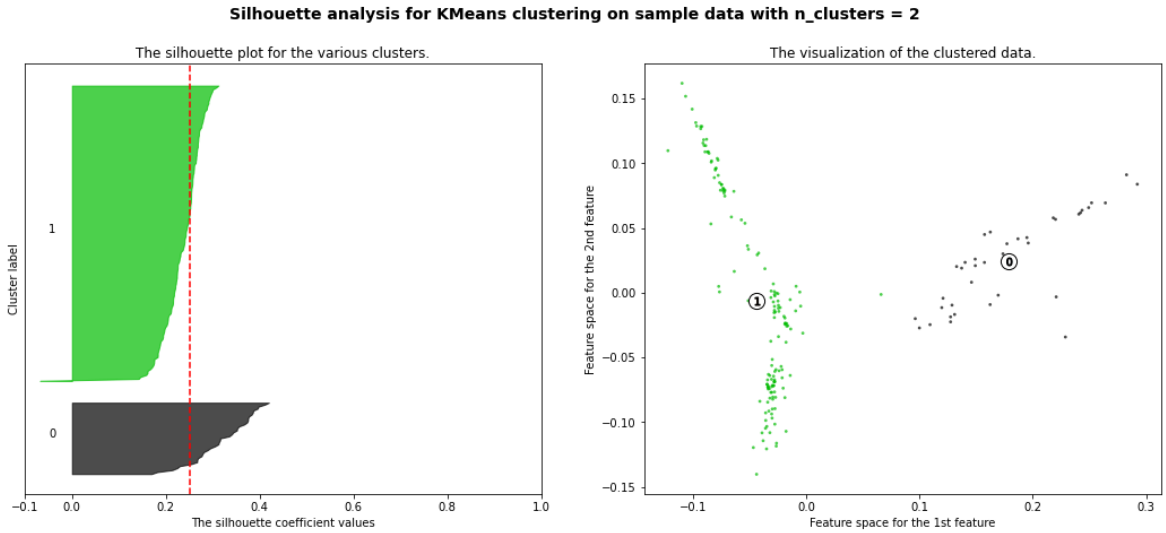

Figura 4.2 *Resultados método de la silueta para K=2 en K-means.*

La Figura 4.2 muestra los resultados obtenidos con el método de la silueta para *K*=2. Se observan las siguientes apreciaciones:

- La media del coeficiente de silueta es 0.251, indicando una decente separación entre los dos clústeres.
- El gráfico de silueta muestra que ambos clústeres alcanzan la línea discontinua, lo que sugiere una buena cohesión interna dentro de cada clúster.
- No hay partes negativas significativas, lo que indica que casi todas las muestras están correctamente asignadas a sus respectivos clústeres.
- La mayor parte del gráfico de silueta es bastante gruesa, lo que indica que los clústeres tienen un tamaño considerablemente grande.

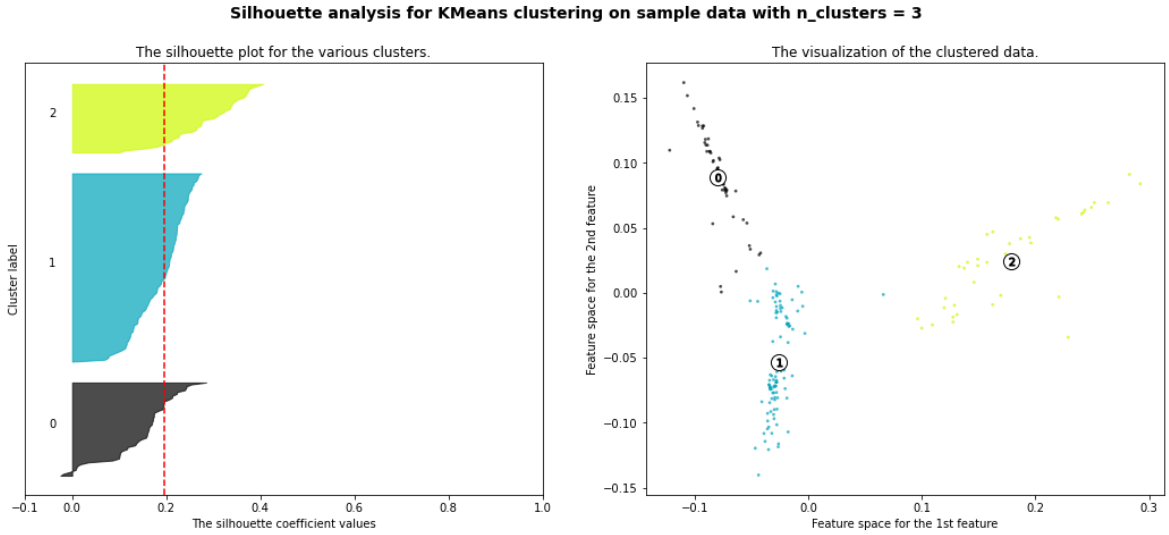

Figura 4.3 *Resultados método de la silueta para K=3 en K-means.*

La Figura 4.3 muestra los resultados obtenidos con el método de la silueta para *K*=3. Se observan las siguientes apreciaciones:

- La media del coeficiente de silueta es 0.196, lo cual es menor que para *K*=2, pero no es tan baja como para los siguientes casos.
- El gráfico de silueta muestra una pequeña parte negativa para uno de los clústeres. Esto podría deberse a la cercanía de algunos puntos al límite de decisión entre dos clústeres.
- La mayoría de los clústeres alcanzan o están muy cerca de la línea discontinua, sugiriendo una buena cohesión interna.
- La fluctuación en el grosor de las parcelas de silueta no es muy grande, indicando que los clústeres tienen tamaños relativamente uniformes.

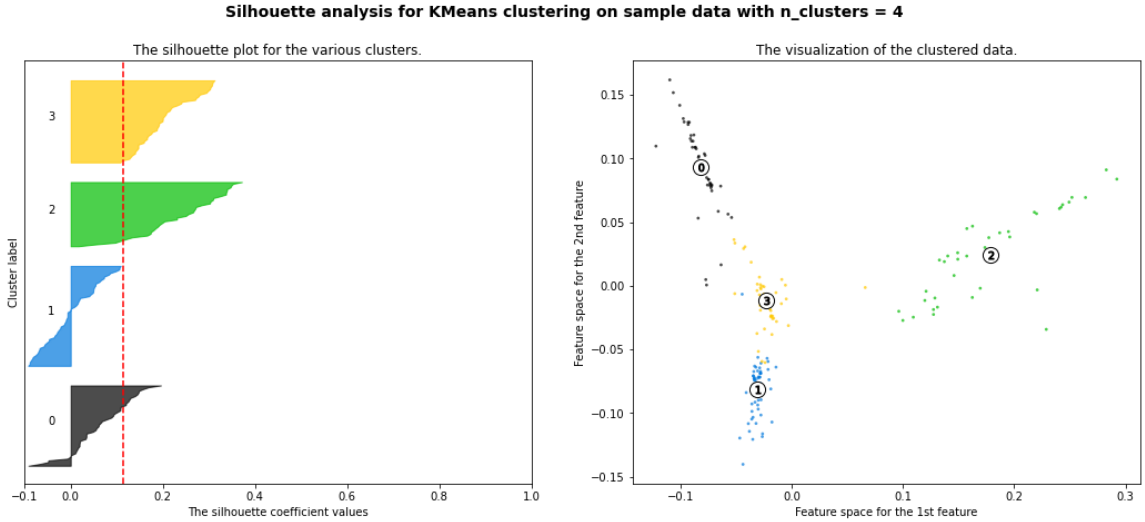

Figura 4.4 *Resultados método de la silueta para K=4 en K-means.*

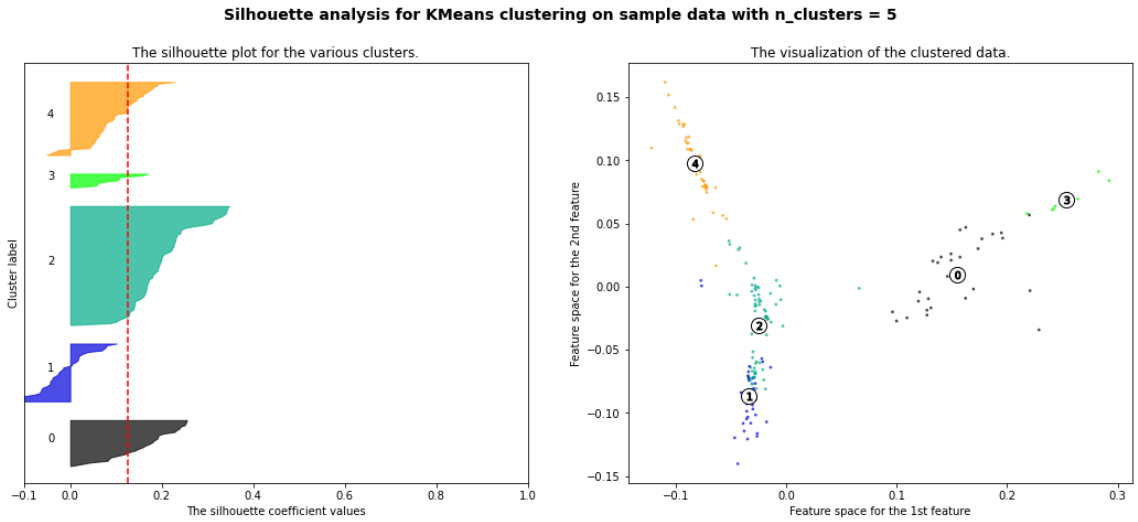

Figura 4.5 *Resultados método de la silueta para K=5 en K-means.*

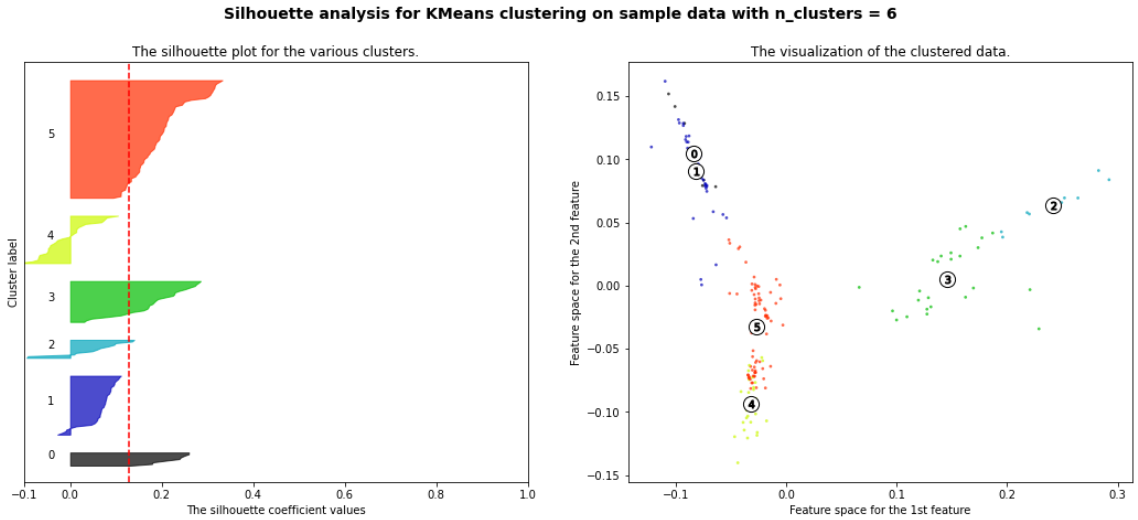

Figura 4.6 *Resultados método de la silueta para K=6 en K-means.*

La Figura 4.4, la Figura 4.5 y la Figura 4.6 muestran los resultados obtenidos con el método de la silueta para *K*=4, *K*=5 y *K*=6, respectivamente. Se observan las siguientes apreciaciones:

- La media de los coeficientes de silueta disminuye aún más, lo que sugiere una menor separación entre los clústeres.
- Los gráficos de silueta muestran varias partes negativas significativas, indicando que muchas muestras podrían estar incorrectamente asignadas a sus clústeres.
- Se observan fluctuaciones notables en el grosor de las parcelas de silueta, indicando clústeres de tamaños desiguales.
- No todos los clústeres no alcanzan la línea discontinua, lo que sugiere que hay una cohesión interna pobre dentro de estos clústeres.

Los resultados del análisis de la silueta indican que los valores de *K*=2 y *K*=3 son las opciones óptimas. Ambos valores muestran una buena cohesión interna y separación entre los clústeres, con la mayoría de los puntos alcanzando la línea discontinua y pocas partes negativas. Aunque *K*=3 tiene una pequeña parte negativa, esto puede atribuirse a la proximidad de algunos puntos al límite de decisión entre dos clústeres, pero en general, la estructura de los clústeres sigue siendo clara y bien definida.

Al realizar el análisis combinado, el método del codo sugiere la elección de 3 o 4 clústeres y el análisis de la silueta también respalda la opción de *K*=3 con una buena cohesión interna y separación entre los clústeres. Por ello, se decide que el número óptimo de clústeres para nuestro conjunto de datos es 3.

Por otra parte, valoramos el uso de PCA. La Figura 4.7 y la Figura 4.8 muestran los clústeres encontrados al emplear K-means con y sin PCA, respectivamente. Durante la fase experimental, se varió el número de componentes principales al aplicar K-means y se observó que la distribución de pacientes en los distintos clústeres era la misma en todos los casos. Por lo tanto, se decidió emplear PCA con todas las componentes principales disponibles para obtener una representación más clara de los clústeres. Los resultados finales del algoritmo K-means se muestran en la Figura 4.7. Se obtiene que el Clúster 0 tiene 50 sujetos; el Clúster 1, 100 y el Clúster 2, 37.

Por último, la Figura 4.7, la Figura 4.9 y la Figura 4.10, representan las componentes principales 0 y 1 (Figura 4.7), 0 y 2 (Figura 4.9) y 1 y 2 (Figura 4.10) para los sujetos con un número de

clústeres *K*=3, para tener una visión más completa y detallada de la estructura de los datos, ayudar a validar la reducción de dimensionalidad y el *clustering*, y facilitar la interpretación de los resultados. En este caso se observa que el Clúster 2 está claramente separado de los demás, como se observa en las figuras 4.7 y 4.9, mientras que el Clúster 0 y el Clúster 1 tienden a estar más juntos en las 3 figuras. Estas características se tendrán en cuenta en la posterior interpretación de los resultados.

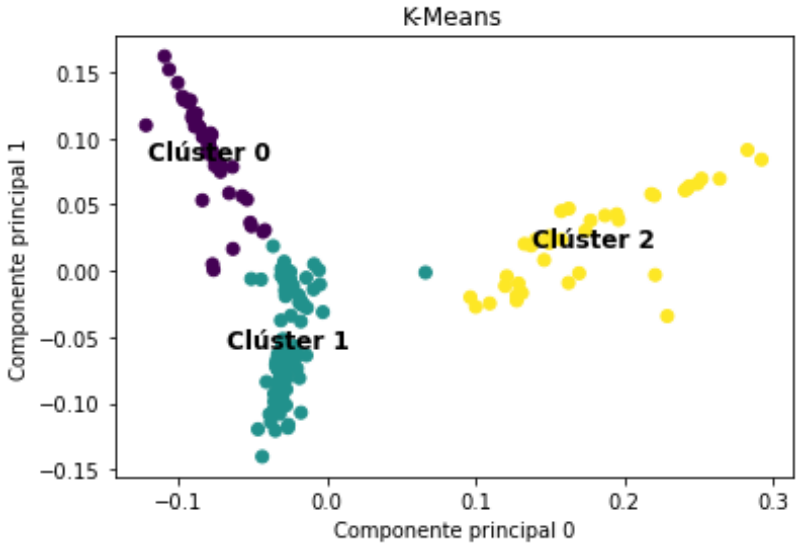

Figura 4.7 *Resultados K-means con PCA.*

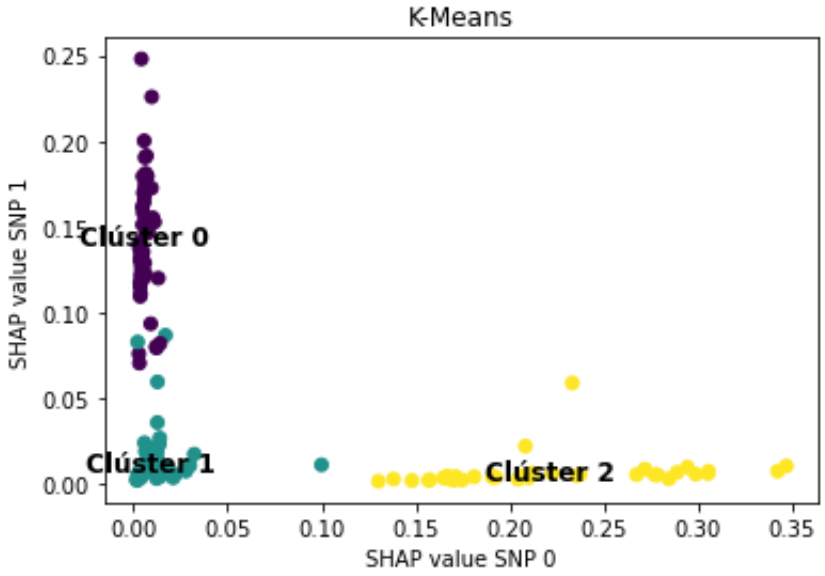

Figura 4.8 *Resultados K-means sin PCA.*

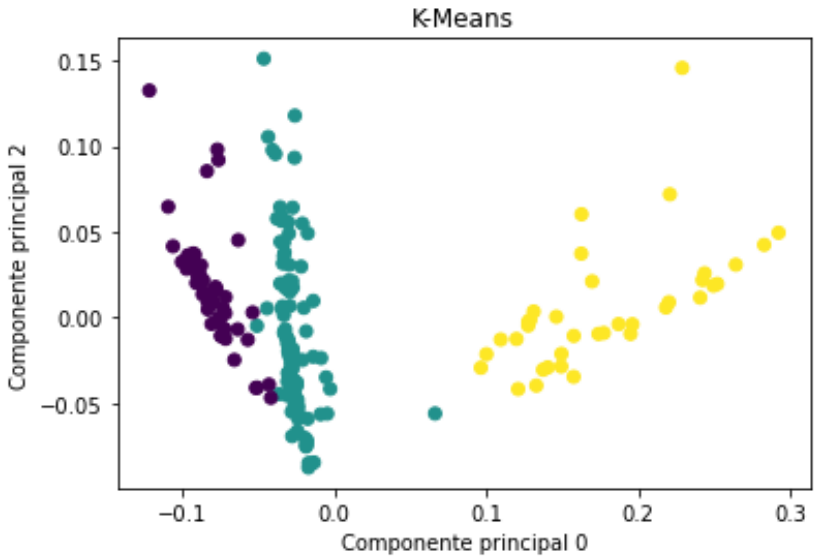

Figura 4.9 *Resultados K-means componente principal 0 y 2.*

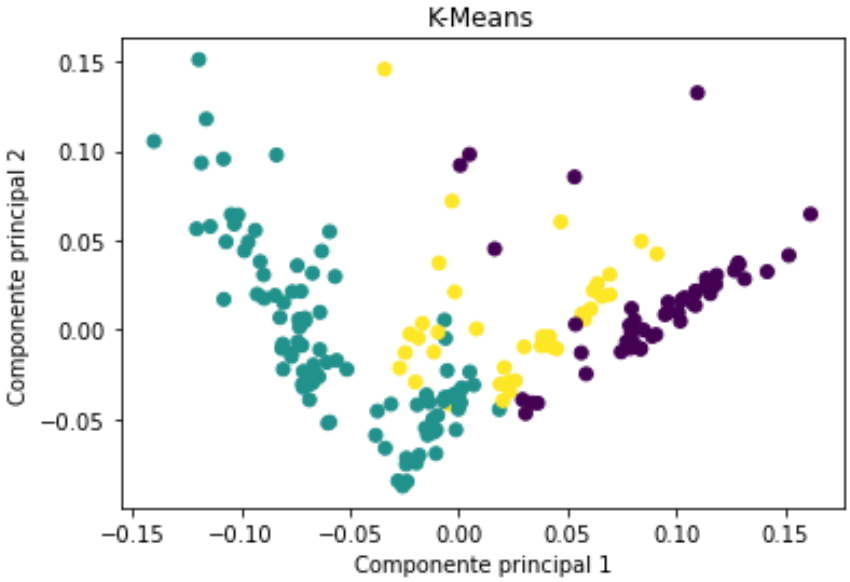

Figura 4.10 *Resultados K-means componente principal 1 y 2.*

4.1.2 DBSCAN

El primer paso es obtener los valores de los hiperparámetros *Eps* y *minPts* que mejor se adaptan a nuestros datos. De manera experimental se han obtenido los siguientes valores óptimos: *Eps* = 0.15 y *minPts* = 5.

La Figura 4.11 muestra los clústeres obtenidos al aplicar DBSCAN con estos hiperparámetros. Se puede observar que el número de clústeres es 3, uno de ellos correspondiente al ruido. Los clústeres se componen de 106, 76 y 5 sujetos, siendo el de 106 el correspondiente al del ruido. Este método confirma que la elección de $K = 3$ en el algoritmo K-means es adecuada. Sin embargo, la diferenciación entre clústeres es más difusa, lo que resulta en una menor calidad de los resultados.

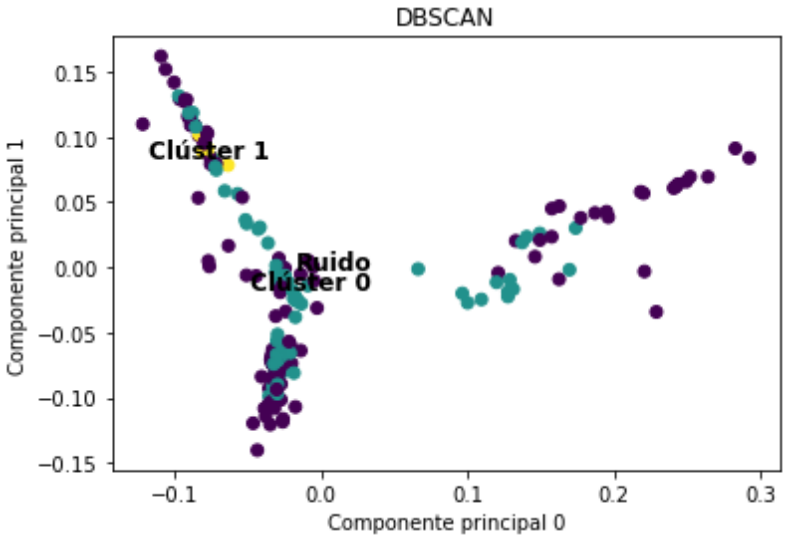

Figura 4.11 *Resultados DBSCAN.*

Analizando la Figura 4.11, se concluye que DBSCAN no proporciona buenos resultados porque los datos no presentan una estructura de densidad clara, es decir, no hay regiones con una alta concentración de puntos separadas por áreas de baja densidad (Khan et al., 2014). Los puntos parecen estar distribuidos de manera relativamente uniforme, sin formar grupos densos bien definidos. Como resultado, DBSCAN clasifica erróneamente muchos puntos como "ruido" y no logra identificar clústeres distintos de manera efectiva.

Por otro lado, hemos observado anteriormente como K-means se ajusta mejor a los datos porque busca dividir el espacio en clústeres de tamaño y forma similares, lo cual es más apropiado para distribuciones de datos donde los clústeres no están definidos por densidad, sino por proximidad en el espacio de características. Por lo tanto, se decide elegir distribución de los sujetos en clústeres de K-means (ver Figuras 4.7, 4.9 y 4.10), para la posterior interpretación clínica y funcional.

4.2 Interpretación clínica de los clústeres

Los resultados del análisis clínico de los clústeres se muestran todos en las Tablas 4.2, 4.3, 4.4 y 4.5. Se observa cómo solo 3 variables clínicas obtienen diferencias estadísticamente significativas (p-valor < 0.05) entre clústeres: Mortalidad en 90 días, Días de Ventilación Mecánica y Muerte en UCI.

En el Clúster 0, se observa que la mortalidad en 90 días (44%) es más alta que la muerte en UCI (32%), lo que sugiere que una parte significativa de los pacientes que sobreviven la UCI fallecen posteriormente. Por otra parte, la duración de la ventilación mecánica (mediana de 9 días) es relativamente prolongada, lo que indica que los pacientes tienen una condición clínica seria que requiere soporte ventilatorio extendido.

El Clúster 1, tiene la menor mortalidad en 90 días (34%) y la menor tasa de muerte en UCI (21%), sugiriendo una condición clínica menos grave en comparación con los otros clústeres. La mediana de días de ventilación mecánica parecida al Clúster 0 (8 días), lo que indica que los pacientes en este clúster también requieren soporte ventilatorio extendido, aunque su recuperación general es más favorable, lo que se refleja en las tasas de mortalidad más bajas.

El Clúster 2 tiene las tasas más altas de mortalidad en 90 días (62.2%) y muerte en UCI (45.9%), indicando que los pacientes tienen la peor condición clínica. La mediana de días de ventilación

mecánica (12 días) es la más alta, reflejando la gravedad de la enfermedad y la necesidad de un soporte más prolongado.

Finalmente, también observamos diferencias estadísticamente significativas entre pares de clústeres, entre el Clúster 1 y el Clúster 2 en las variables clínicas: mortalidad en 90 días, días en UCI, días de ventilación mecánica y muerte en UCI.

Variables	Clúster 0	Clúster 1	Clúster 2	*p*-valor
Sexo (% hombres)	62.0	76.0	59.5	0.082
Shock séptico (%)	86.0	86.0	81.1	0.754
Mortalidad en 90 días (%)	44.0	34.0	62.2	0.012[(a)]

Tabla 4.2 *Análisis estadístico de las variables clínicas categóricas. (a): Diferencias estadísticamente significativas entre el Clúster 1 y el Clúster 2.*

Variables	Clúster 0	Clúster 1	Clúster 2	*p*-valor
Edad (años)	68.0, [60.0, 75.8]	71.0, [60.0, 78.0]	74.0, [67.0, 77.0]	0.394
Neutrófilos (células/mm3)	86.9 [83.4, 90.0]	88.3, [81.8, 91.5]	87.8, [81.0, 90.4]	0.839
Linfocitos (células/mm3)	6.7, [4.20, 10.3]	7.3, [4.3, 11.6]	8.2, [5.50, 12.7]	0.614
Glóbulos Blancos (células/mm3)	11570.0, [6662.5, 16307.5]	12323.5, [6997.5, 16877.5]	11780.0, [7120.0, 15640.0]	0.907
Creatinina (mg/dl)	1.7, [1.1, 2.8]	1.7, [1.1, 3.3]	1.5, [1.1, 2.2]	0.428

Tabla 4.3 *Análisis estadístico de las variables clínicas continuas.*

Comorbilidades	Clúster 0	Clúster 1	Clúster 2	*p*-valor
Fallo renal crónico (%)	12.5	12.5	17.6	0.732
Enfermedad cardiovascular crónica (%)	41.7	40.4	30.6	0.521
Obesidad (%)	12.5	14.1	11.4	0.908
Diabetes mellitus (%)	16.3	25.3	35.1	0.135
Fallo hepático crónico (%)	6.3	7.1	13.9	0.376
Inmunosupresión (%)	8.7	12.2	17.6	0.485
Hipertensión (%)	34.7	44.4	40.5	0.524
Enfermedad respiratoria crónica (%)	14.3	25.3	27.0	0.254

Tabla 4.4 *Análisis estadístico de las comorbilidades.*

Variables	Clúster 0	Clúster 1	Clúster 2	*p*-valor
Días en UCI (n)	14.0, [6.0, 23.0]	13.0, [7.0, 19.0]	17.0, [10.0, 25.0]	0.120[(a)]
Días en Hospital (n)	32.0, [23.0, 48.0]	32.0, [19.0, 49.3]	32.0, [19.0, 47.0]	0.927
Días Ventilación Mecánica (n)	9.0, [2.0, 19.0]	8.0, [2.75, 15.0]	12.0, [9.0, 20.3]	0.024[(a)]
Muerte en UCI (%)	32.0	21.0	45.9	0.015[(a)]

Tabla 4.5 *Análisis estadístico de las variables de estancia en hospital. (a): Diferencias estadísticamente significativas entre el Clúster 1 y el Clúster 2.*

4.3 Interpretación funcional de los clústeres

Para los resultados del análisis funcional se han creado unas tablas que reflejan los aspectos más importantes de los SNPs de cada clúster (Tablas 4.6-4.8). Posteriormente se analizan dichos SNPs, con especial incidencia en el SNP principal de cada clúster, debido a su gran importancia en el clúster en comparación con el resto. Hay SNPs sin información sobre la frecuencia y mutación en la población ibérica, esto se debe a la falta de datos disponibles para este marcador específico.

4.3.1 Clúster 0 (50 sujetos)

La Figura 4.12 muestra la media de los SHAP *values* de los SNP correspondientes a los pacientes con sepsis del clúster 0. El top 5 SNPs del clúster es: rs74707084, rs903439, rs111474353, rs79219127 y rs75858984:

- **rs74707084:** Es el SNP más importante del clúster. Esta localizado en el intrón del gen SYNPR. En la población Ibérica se produce G>A con una frecuencia de un 1%. Este gen codifica la sinaptoporina, una proteína localizada en el sistema nervioso central.
  A pesar de que SYNPR no presenta información detallada en Genecards sobre vías metabólicas específicas asociadas o enfermedades relacionadas con la sepsis, otras bases de datos, como Malacards, sugieren asociaciones con enfermedades como la inmunodeficiencia común variable tipo 10, cáncer de pulmón y hepatitis autoinmune. Estas asociaciones pueden ofrecer pistas sobre su relevancia en la sepsis.
- rs903439: SNP localizado en el intrón del gen PDE10A. Este gen codifica una proteína que pertenece a la familia de las fosfodiesterasas cíclicas de nucleótidos y que desempeña un papel en la transducción de señales al regular la concentración intracelular de nucleótidos cíclicos. En la población Ibérica se produce C>T con una frecuencia de un 7%.
- rs111474353: SNP localizado en la región intergénica, *upstream* (dirección 5' en la cadena de ADN). En la población Ibérica se produce G>A con una frecuencia de un 1%. Como es una región intergénica afecta a varios genes. Uno de ellos es SFTA2, que codifica una proteína que se localiza en el aparato de Golgi, en la región extracelular y en vesículas de transporte. Asimismo, este SNP afecta al gen DPCR1**,** codifica una proteína localizada en el citoplasma y la membrana plasmática como un componente integral de la membrana.
- rs79219127: SNP localizado en el intrón del gen NALF1, involucrado en canales de calcio y transporte de iones. En la población Ibérica se produce G>C con una frecuencia de un 1%.
- rs75858984: SNP localizado en la región intergénica, *upstream*. En la población Ibérica se produce C>T con una frecuencia de un 1%. En este caso solo se considera relevante la afectación al gen ST8SIA3, que está involucrado en varios procesos, como la biosíntesis de gangliósidos, el metabolismo de glicoproteínas y la sialilación de proteínas.

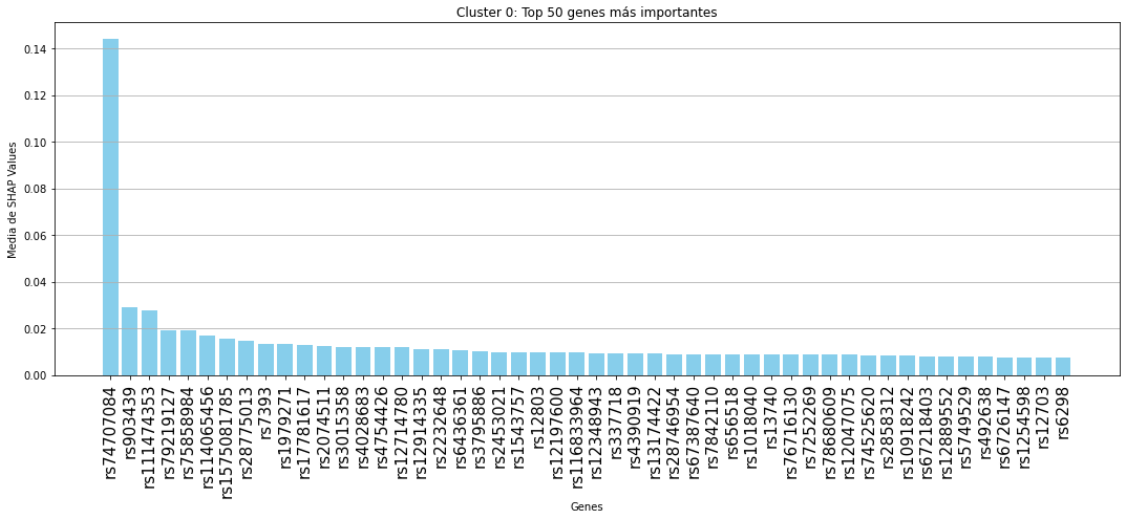

Figura 4.12 *Media de los SHAP values de los SNPs de los sujetos en el Clúster 0.*

Cromosoma	Posición (Hg37)	SNP	Localización	Gen más cercano	*p*-valor^	SHAP *values**
3	63389846	rs74707084	Intrón	SYNPR	1.28E-65	0.1439
6	166147479	rs903439	Intron	PDE10A	1.31E-05	0.0290
6	30901543	rs111474353	Variante intergénica (*upstream*)	SFTA2/DPCR1	6.08E-59	0.0277
13	108050599	rs79219127	Intrón	NALF1	4.09E-33	0.0190
18	55016814	rs75858984	Variante intergénica (*upstream*)	ST8SIA3	1.09E-24	0.0189

Tabla 4.6 *Análisis funcional del Top 5 SNPs del Clúster 0. *Media del SHAP value del SNP en todos los pacientes del clúster, ^ p-valor extraído del análisis GWAS del artículo.*

4.3.2 Clúster 1 (100 sujetos)

La Figura 4.13 muestra la media de los SHAP *values* de los SNP correspondientes a los pacientes con sepsis del clúster 1. El top 5 SNPs del clúster es: rs1575081785, rs77119441, rs201088712, rs79219127 y rs75858984:

- **rs1575081785**: Es el SNP más importante del clúster y se encuentra en el gen RBSN. Es una variante sinónima. Este tipo de variante se define como una mutación en la secuencia del ADN que no cambia la secuencia de aminoácidos de la proteína debido a la redundancia del código genético. Aunque no alteran directamente la secuencia de proteínas, estas variantes pueden afectar la regulación de la expresión génica, la eficiencia de traducción, la estabilidad del ARN mensajero y otros aspectos biológicos importantes. Este gen codifica una proteína de la familia de dedos de zinc FYVE, que está implicada en el tráfico de vesículas.
  Además, es un gen codificante de proteínas que en caso de ser afectado puede causar Neutropenia Congénita Severa, Tipo 5**.** Es un trastorno genético raro caracterizado por una disminución extrema en el número de neutrófilos, un tipo de glóbulo blanco esencial para combatir infecciones bacterianas y fúngicas.
  Por otra parte, RBSN participa en *pathways* relacionados con la respuesta inmune innata y la cascada del receptor Toll-Like 3, un conjunto de interacciones bioquímicas que son relevantes en el contexto de la sepsis. Un *pathway* es una serie de interacciones bioquímicas que ocurren dentro de una célula, donde moléculas específicas actúan en secuencia para llevar a cabo una

función biológica particular, como la señalización celular, el metabolismo o la regulación genética.

- rs77119441: SNP localizado en la región intergénica, *downstream* (dirección 3' en la cadena de ADN). En la población Ibérica se produce A>G con una frecuencia de un 3%. En este caso solo se considera relevante la afectación al gen SORL1, que participa en la endocitosis y clasificación de proteínas, especialmente aquellas relacionadas con la enfermedad de Alzheimer. Este receptor también está involucrado en la señalización lipídica.
- rs201088712: Es una variante *missense* del gen CENPJ. Este tipo de SNP se define como una mutación en la secuencia del ADN que provoca un cambio en un codón, resultando en la incorporación de un aminoácido diferente en la proteína. Este cambio puede alterar la estructura y función de la proteína, con potenciales efectos significativos en la biología del organismo y en la predisposición a enfermedades. El gen afectado es esencial para la integridad del centrosoma y la división celular. También juega un papel en la disociación de microtúbulos y la regulación de la transcripción.
- rs79219127: SNP localizado en el intrón del gen NALF1, que está involucrado en los canales de calcio y el transporte de iones. En la población Ibérica se produce C>G con una frecuencia de un 6%.
- rs75858984: SNP localizado en la región intergénica, *upstream*. En la población Ibérica se produce C>T con una frecuencia de un 1%. En este caso solo se considera relevante la afectación al gen ST8SIA3, que está involucrado en varios procesos, como la biosíntesis de gangliósidos, el metabolismo de glicoproteínas y la sialilación de proteínas.

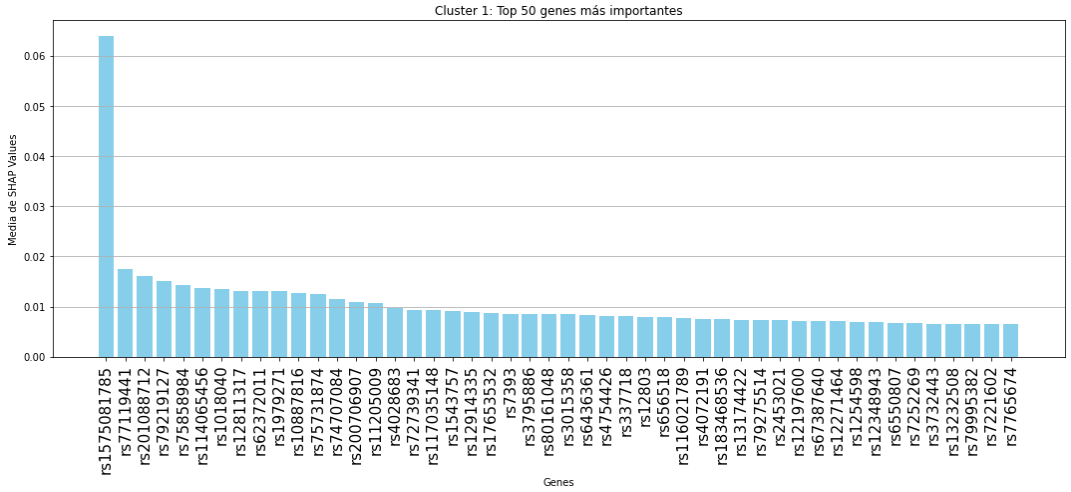

Figura 4.13 *Media de los SHAP values de los SNPs de los sujetos en el Clúster 1.*

Cromosoma	Posición (Hg37)	SNP	Localización	Gen más cercano	*p*-valor^	SHAP *values**
3	151153369	rs1575081785	Variante sinónima	RBSN	4.98E-40	0.0638
11	121787956	rs77119441	Variante intergénica (*downstream*)	SORL1	1.47E-16	0.0174
13	25480913	rs201088712	*Missense*	CENPJ	3.49E-29	0.0161
13	108050599	rs79219127	Intrón	NALF1	4.09E-33	0.0151
18	55016814	rs75858984	Variante intergénica (*upstream*)	ST8SIA3	1.09E-24	0.0143

Tabla 4.7 *Análisis funcional del Top 5 SNPs del Clúster 1. *Media del SHAP value del SNP en todos los pacientes del clúster, ^ p-valor extraído del análisis GWAS del artículo.*

4.3.3 Clúster 2 (37 sujetos)

La Figura 4.14 muestra la media de los SHAP *values* de los SNP correspondientes a los pacientes con sepsis del clúster 2. El top 5 SNPs del clúster es: rs17653532, rs1018040, rs1575081785, rs79275514 y rs62372011:

- **rs17653532:** Es el SNP más importante del clúster. Localizado en el intrón del gen PRIM2, que en la población Ibérica se produce A>C con una frecuencia de un 1%. PRIM2 codifica la subunidad de 58 kilodalton de la ADN primasa, una enzima que desempeña un papel clave en la replicación del ADN. Las mutaciones en PRIM2 se han vinculado a enfermedades como el síndrome de Seckel. Además, PRIM2 está implicado en vías cruciales como la activación del complejo pre-replicativo y la síntesis de la cadena retardada en los telómeros, subrayando su relevancia en procesos de replicación y estabilidad genómica.
- rs1018040: SNP localizado en una región intergénica, downstream. En la población Ibérica se produce T>A con una frecuencia de un 8%. Como es una región intergénica afecta a varios genes. Uno de ellos es TGFB2, codifica un ligando de la superfamilia TGF-beta que regula la expresión génica a través de la activación de factores de transcripción SMAD y participa en la formación de péptidos maduros y latentes que afectan diversas funciones celulares y procesos patológicos. Asimismo, este SNP también afecta a LYPLAL1, que participa en la despalmitoilación de proteínas, un proceso que regula la localización y función de varias proteínas dentro de la célula. Este gen también está involucrado en el transporte de proteínas desde el Golgi a la membrana plasmática.
- rs1575081785: Es una variante sinónima que afecta al gen RBSN, involucrado en el tráfico de membranas y la endocitosis, procesos esenciales para la correcta señalización y homeostasis celular.
- rs79275514: SNP localizado en el intrón del gen PRKN. En la población Ibérica se produce T>C con una frecuencia de un 1%. PRKN juega un papel crucial en la ubiquitinación y degradación de proteínas a través del sistema de proteasomas. También es esencial para la homeostasis celular y la respuesta al estrés.
- rs62372011: SNP localizado en el intrón del gen PDE4D. En la población Ibérica se produce T>C con una frecuencia de un 4%. PDE4D codifica una proteína que degrada el cAMP, una molécula clave en la transducción de señales celulares, y puede generar múltiples isoformas funcionales a través de empalme alternativo.

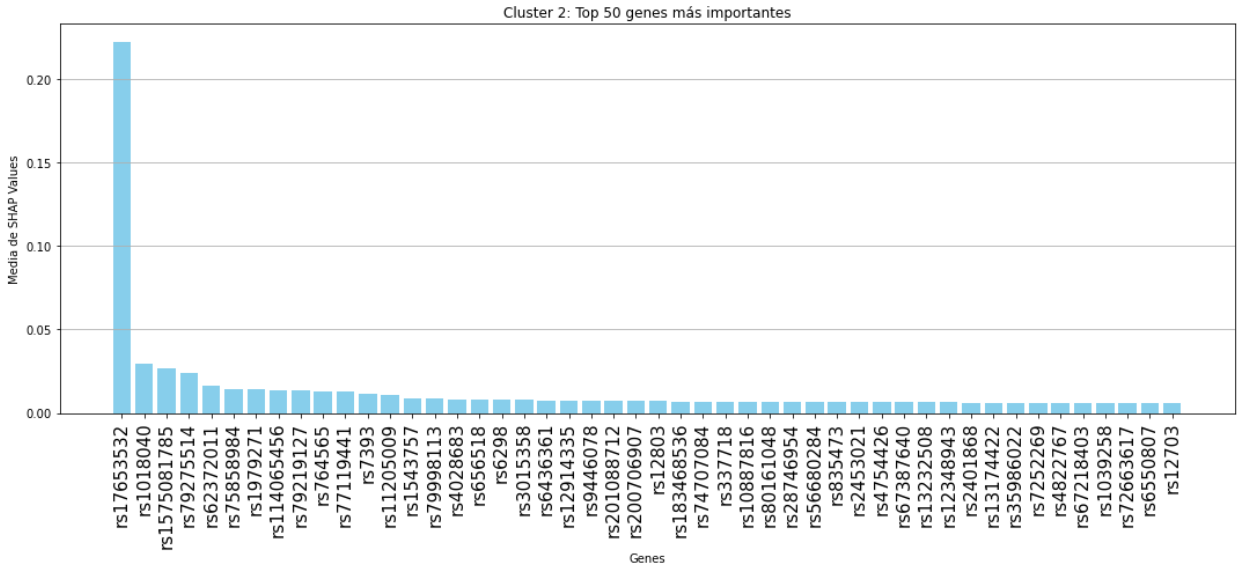

Figura 4.14 *Media de los SHAP values de los SNPs de los sujetos en el Clúster 2.*

Cromosoma	Posición (Hg37)	SNP	Localización	Gen más cercano	*p*-valor^	SHAP values*
6	57189167	rs17653532	Intrón	PRIM2	1.18E-03	0.2217
1	218705814	rs1018040	Variante intergénica (Downstream)	TGFB2/LYPLAL1	1.69E-05	0.0292
3	15115369	rs1575081785	Variante sinónima	RBSN	4.98E-40	0.0265
6	162119549	rs79275514	Intrón	PRKN	8.30E-04	0.0237
5	59757497	rs62372011	Intrón	PDE4D	2.28E-10	0.0161

Tabla 4.8 *Análisis funcional del Top 5 SNPs del Clúster 2. *Media del SHAP value del SNP en todos los pacientes del clúster, ^ p-valor extraído del análisis GWAS del artículo.*

Capítulo 5. Discusión

5.1 Discusión de los resultados obtenidos

Los resultados obtenidos en este TFG mediante el análisis de *clustering* con K-means de los SHAP *values* de los SNPs de los pacientes con sepsis han permitido identificar 3 clústeres distintos. El propósito de esta segmentación es entender mejor las diferencias genéticas y clínicas entre los grupos de pacientes, lo que puede facilitar la personalización del tratamiento y mejorar los resultados clínicos. Para ello se realiza una interpretación funcional y clínica de cada clúster, asociándolo a su vez con la sepsis:

- **Clúster 0.** El SNP principal del clúster es rs74707084, que afecta al gen SYNPR. Este gen codifica sinaptoporina, una proteína localizada en el sistema nervioso central. Esta proteína está implicada en el tráfico de vesículas sinápticas y la liberación de neurotransmisores (Knaus et al., 1990).
  Este clúster presenta una alta mortalidad a 90 días y una relativamente prolongada ventilación mecánica. La alta mortalidad post-UCI sugiere que las complicaciones persistentes podrían estar relacionadas con la disfunción en la comunicación neuronal y su posible impacto en la respuesta inmunitaria, como se observa en su implicación con la inmunodeficiencia común variable tipo 10, sugiere un papel potencial en la disfunción inmune. Esto dificultaría una recuperación completa tras el alta hospitalaria (Pei et al., 2024).
- **Clúster 1.** El SNP principal del clúster es rs1575081785, que afecta al gen RBSN. Este gen codifica una proteína de la familia de dedos de zinc FYVE, que está implicada en el tráfico de vesículas. Las proteínas dedos de zinc son fundamentales en la modulación de la expresión génica, regulando genes clave que intervienen en la inflamación, la función de las células inmunes y la reparación de tejidos. En el contexto de la sepsis, estas proteínas pueden influir en la disfunción celular, la apoptosis y los mecanismos de reparación del ADN, afectando así la supervivencia celular y la integridad de los tejidos (Liu et al., 2013; Rakhra & Rakhra, 2021). Además, participa en *pathways* relacionados con la respuesta inmune innata y la cascada del receptor Toll-Like. Estos *pathways* son cruciales para la detección y respuesta a patógenos, y están directamente relacionados con la sepsis (Wiersinga et al., 2014) y sumado a la asociación con la Neutropenia Congénita Severa refuerza la conexión con una mayor vulnerabilidad a infecciones graves (Guzmán Cotaya et al., 2021).
  Este clúster muestra la menor tasa de mortalidad en UCI y a 90 días, junto con una mediana de ventilación mecánica más baja. A pesar de la afectación del sistema inmune, estos pacientes parecen beneficiarse de un manejo médico intensivo, lo que podría explicar las menores tasas de mortalidad y la duración más corta de la ventilación mecánica en comparación con otros clústeres (Genga & Russell, 2017).
- **Clúster 2.** El SNP principal del clúster es rs17653532, que afecta al gen PRIM2. Este gen está involucrado en la replicación del ADN, un proceso crucial en el contexto de la sepsis debido a la apoptosis, el estrés oxidativo y los cambios metabólicos que ocurren durante esta condición. Además, PRIM2 ha sido relacionado con casos de sepsis consecutiva a trauma, según análisis de perfiles de expresión (Dong et al., 2018), y participa en vías como la activación del complejo pre-replicativo y la síntesis de la cadena retardada en los telómeros. Este clúster presenta las tasas más altas de mortalidad en UCI y a 90 días, así como la mayor duración de ventilación mecánica. Esto refleja la gravedad de la sepsis y las complicaciones severas asociadas con defectos en la reparación del ADN y la respuesta inmune (Mihaljevic et al., 2023).

Comparando brevemente los clústeres, el Clúster 2, caracterizado por la alteración en el gen PRIM2, muestra las peores variables clínicas, con la mayor mortalidad a los 90 días (62.2%) y en UCI (45.9%), así como la duración más prolongada de ventilación mecánica (mediana de 12 días). PRIM2 es crucial para la replicación del ADN, y su disfunción agrava la sepsis al dificultar la reparación celular y la gestión del estrés oxidativo, lo que resulta en mayor gravedad clínica y una elevada necesidad de soporte vital prolongado. En contraste, los Clústeres 0 y 1 presentan variables clínicas más favorables en comparación con el Clúster 2, aunque con diferencias entre ellos. En el Clúster 0, donde el gen afectado es SYNPR, se observa una mortalidad intermedia a los 90 días (44%) y en UCI (32%), con una mediana de 9 días de ventilación mecánica. Aunque la disfunción del gen SYNPR, que está relacionado con la liberación de neurotransmisores, no está directamente implicada en la respuesta inmune, podría influir indirectamente en la capacidad de recuperación del sistema nervioso tras el alta hospitalaria, lo que explicaría la alta mortalidad post-UCI observada. Por otro lado, el Clúster 1, con alteración en el gen RBSN, muestra las mejores variables clínicas, con la menor mortalidad a los 90 días (34%) y en UCI (21%), junto con una duración de ventilación mecánica similar a la del Clúster 0 (mediana de 8 días). El gen RBSN, implicado en el tráfico de vesículas y en la regulación de genes clave para la inflamación y la función inmune, parece estar asociado con una menor gravedad clínica, probablemente porque, aunque los pacientes en este clúster tienen un sistema inmune afectado, su respuesta clínica se beneficia más de un manejo médico intensivo, lo que les permite una recuperación más favorable.

Además, hay que tener en cuenta el análisis estadístico por pares de clústeres. Los resultados mostraron diferencias estadísticamente significativas entre los Clústeres 1 y 2 en cuatro variables clínicas clave: mortalidad en 90 días, días en UCI, días de ventilación mecánica y mortalidad en UCI. Estos resultados subrayan la gravedad clínica del Clúster 2 en comparación con el Clúster 1, evidenciando una mayor necesidad de intervención médica y peores resultados clínicos en los pacientes del Clúster 2. En contraste, no se encontraron diferencias estadísticamente significativas cuando se compararon los clústeres con el Clúster 0, lo que sugiere que, aunque pueda haber diferencias clínicas, estas no alcanzan un nivel de significancia estadística bajo el umbral utilizado por el reducido tamaño muestral del estudio. Este hallazgo respalda la interpretación de que el Clúster 0 representa un grupo intermedio en términos de severidad clínica y resultados, con complicaciones más manejables en un entorno clínico adecuado.

También es fundamental comparar los resultados obtenidos en este TFG con los obtenidos en el estudio de referencia previo realizado por López Herrero et al. (2024). En dicho estudio, se identificaron los SNPs más importantes para la predicción de sepsis, destacando 3 SNPs principales: rs17653532, rs1575081785 y rs74707084. Los resultados de este TFG muestran que los SNPs principales de los 3 clústeres detectados coinciden con los SNPs más importantes identificados en el estudio de López Herrero et al. (2024). Esta coincidencia resalta la validez de nuestra metodología de *clustering* y su capacidad para identificar SNPs relevantes en la predisposición genética a la sepsis. Además, los SNPs que conforman el top 5 de cada clúster se encuentran dentro del top 20 de SNPs identificados en el estudio previo, lo que refuerza aún más la consistencia y relevancia de nuestros hallazgos.

Además, la identificación de estos tres clústeres mediante técnicas de *clustering* aplicadas a los SHAP *values* de los SNPs ha demostrado ser de gran utilidad por varias razones:

- **Personalización del tratamiento**. Permite personalizar las estrategias de tratamiento basadas en las características genéticas y las complicaciones asociadas de cada clúster. Por ejemplo, los pacientes en el clúster 1 podrían beneficiarse de un seguimiento más intensivo y profilaxis para prevenir infecciones, mientras que los pacientes en el clúster 2 podrían requerir tratamientos más agresivos y monitoreo constante en la UCI.

- **Predicción de resultados clínicos**. Facilita la predicción de los resultados clínicos, como la mortalidad y la necesidad de ventilación mecánica, lo que permite una mejor planificación y asignación de recursos en un entorno clínico.
- **Investigación genética y biomédica**. Proporciona perspectivas valiosas para la investigación genética y biomédica, identificando SNPs clave y sus asociaciones con enfermedades específicas y complicaciones en sepsis. Es crucial destacar que un clúster no se define por un solo SNP, sino por la combinación de varios SNPs que interactúan conjuntamente en un paciente. Esta integración es esencial para el enfoque de IA en genética aplicada a la sepsis, y guía futuras investigaciones y el desarrollo de nuevas terapias.

5.2 Comparación de resultados

Como se ha observado en la revisión bibliográfica, estudios como Papin et al. (2021) y Jang et al. (2022) se centraron en identificar clústeres de pacientes con sepsis utilizando características clínicas y hematológicas. Papin et al. (2021) emplearon análisis de correspondencias múltiples para extraer las primeras 52 dimensiones de 63 variables clínicas, con el objetivo de identificar seis clústeres de pacientes basados en estas características clínicas y biológicas disponibles al ingreso en la UCI. Jang et al. (2022), por otro lado, utilizaron K-means y silueta para identificar tres y cuatro clústeres basados en características como edad, recuento de leucocitos relación neutrófilo a linfocito, hemoglobina, plaquetas, entre otros, para identificar factores de riesgo robustos para la sepsis.

A diferencia de estos estudios, en este TFG se aplican técnicas de *clustering* a datos genéticos. En lugar de enfocarse únicamente en datos clínicos, se utiliza información genética a nivel de SNPs para realizar la agrupación de los pacientes. Este enfoque genético es crucial porque permite una comprensión más profunda de las bases moleculares de la sepsis, que los datos clínicos por sí solos no pueden proporcionar. La integración de datos genéticos en el análisis permite identificar variaciones individuales en el genoma que pueden predisponer a la sepsis, proporcionando una perspectiva novedosa y potencialmente más robusta sobre las diferencias entre los pacientes con sepsis (Dahmer et al., 2005). Esta capacidad de detección genética avanzada no solo mejora la personalización del tratamiento, sino que también puede ayudar a desarrollar modelos predictivos y estrategias terapéuticas más precisas, mejorando así los resultados clínicos (Stanski & Wong, 2019).

Por otra parte, hay otros estudios que sí que utilizan datos genéticos, los estudios de Maslove et al. (2012) y Zhang et al. (2020). Sin embargo, observamos diferencias en la metodología y los objetivos. Maslove et al. (2012) se enfocaron en identificar subtipos moleculares de sepsis basados en patrones de expresión génica, utilizando el algoritmo PAM con silueta y HC para identificar dos subtipos. Zhang et al. (2020) buscaron identificar subgrupos de pacientes con sepsis y desarrollar un modelo predictivo basado en características genéticas, utilizando K-means con codo y silueta para identificar dos clústeres basados en 50 características seleccionadas por su relación con la mortalidad.

Este TFG, en cambio, se distingue por el uso de SHAP *values* para realizar el *clustering*. Los SHAP *values* permiten una interpretación detallada y precisa de la importancia de cada SNP en la predicción de los resultados clínicos. Esta técnica ofrece ventajas significativas en términos de interpretabilidad y precisión, facilitando una comprensión más clara de cómo cada variante genética contribuye al riesgo y al pronóstico de la sepsis (Johnsen et al., 2021). El uso de SHAP *values* no solo mejora la capacidad de identificar subgrupos genéticamente distintos, sino que también proporciona una base sólida para la personalización del tratamiento. Además, al final de nuestro análisis, también se obtienen las características clínicas de cada clúster, lo que permite

una integración completa de los datos genéticos y clínicos para una mejor comprensión y manejo de la sepsis. Esta combinación de datos genéticos con SHAP *values* y análisis clínicos nos permite ofrecer una metodología más robusta y exhaustiva que los estudios previos, subrayando la importancia de un enfoque multidimensional para el tratamiento y la investigación de la sepsis.

5.3 Limitaciones

Los resultados obtenidos en este TFG pueden estar sujetos a limitaciones que deben ser consideradas:

- **Tamaño muestral**. Aunque se han analizado un número considerable de sujetos, un tamaño de muestra mayor podría proporcionar resultados más robustos y generalizables. Un mayor número de participantes permitiría una mejor evaluación de la estructura genética y clínica de los pacientes, aumentando la precisión y solidez de los hallazgos.
- **Representatividad de la muestra.** Este estudio se basa en una base de datos de pacientes caucásicos postoperatorios con sepsis de dos hospitales en España, lo que podría limitar la generalización de los resultados a otras poblaciones. Los hallazgos podrían no ser completamente aplicables a conjuntos de datos independientes y geográficamente distintos, especialmente en pacientes postoperatorios con diversas ascendencias genéticas y factores ambientales. La inclusión de datos de otras regiones y poblaciones podría mejorar la validez externa de los resultados y proporcionar una representación más amplia de la sepsis en diferentes contextos.
- **Selección de SNPs**. La elección de SNPs y su interpretación funcional puede estar limitada por la base de datos y el conocimiento actual. La inclusión de más SNPs y un análisis funcional más exhaustivo podrían mejorar la comprensión de los mecanismos genéticos subyacentes.
- **Métodos de *clustering***. Aunque se utilizaron K-means y DBSCAN, otros métodos de *clustering* podrían proporcionar diferentes perspectivas sobre la estructura de los datos. La exploración de técnicas adicionales, como *clustering* jerárquico o modelos de mezcla gaussiana, podría complementar los resultados obtenidos. Además, la selección de los hiperparámetros de los algoritmos, como el número de clústeres en el caso de K-means, es arbitraria y puede influir en los resultados obtenidos.
- **Mayor número de variables clínicas:** No se han analizado otras variables clínicas relevantes, como la glucosa, bilirrubina, sodio, potasio, fracción de oxígeno inspirado, y presión parcial de oxígeno, entre otras, ya que estas solo se recogieron en un único hospital, el Hospital Clínico Universitario de Valladolid, y no estaban disponibles para todos los sujetos de la cohorte. La falta de estas mediciones en todos los sujetos limita la posibilidad de realizar un análisis más completo, integrando dichas variables adicionales para explorar su relación potencial con los clústeres genéticos identificados.
- **Interpretación de resultados**. La interpretación funcional de los resultados se basa en el análisis de SNPs individuales, utilizando métodos que permiten evaluar cada SNP por separado. Sin embargo, estos métodos no consideran la posible interacción o combinación de múltiples SNPs, lo que podría limitar la comprensión completa de sus efectos en la biología y la genética de la sepsis. La falta de un análisis integrado de los SNPs puede restringir la identificación de patrones complejos y mecanismos subyacentes que podrían ser relevantes para la sepsis. Futuras investigaciones que integren análisis de interacción entre SNPs podrían ofrecer una visión más completa y refinada de los factores genéticos implicados.

Capítulo 6. Conclusiones y líneas futuras

6.1 Contribuciones

Este TFG ha realizado varias contribuciones significativas al campo del estudio de la sepsis y su relación con la genética:

- En primer lugar, se ha desarrollado una metodología innovadora al aplicar técnicas de *clustering* a los SHAP *values* de genes asociados con la sepsis. Esta aplicación es inédita en el campo, ya que no se había utilizado previamente el *clustering* en SHAP *values* para analizar perfiles genéticos en sepsis. Este enfoque ha permitido interpretar de manera más clara y detallada la importancia de los SNPs seleccionados en el modelo predictivo, ofreciendo una nueva perspectiva para entender los mecanismos genéticos subyacentes a la sepsis y marcando una contribución significativa al estudio de la enfermedad.
- También se ha realizado una comparación exhaustiva entre los métodos de *clustering* K-means y DBSCAN. Los resultados indicaron que K-means es más adecuado para los datos analizados, ya que se ajusta mejor a la estructura de los datos en comparación con DBSCAN, que no identificó clústeres bien definidos. Esta comparación proporciona una comprensión más profunda sobre la selección de métodos de *clustering* para datos genéticos y clínicos en estudios de sepsis.
- Finalmente, se ha llevado a cabo una interpretación clínica y funcional de cada clúster identificado. Este análisis ha facilitado la comprensión de cómo los diferentes perfiles genéticos pueden influir en la severidad y el pronóstico de la sepsis, proporcionando una visión más profunda de los mecanismos genéticos y su impacto en la evolución de la enfermedad.

6.2 Conclusiones

Este TFG ha cumplido con éxito el objetivo principal planteado: identificar, mediante técnicas de IA, clústeres de pacientes con sepsis con distinto perfil clínico y genético. Las conclusiones generales obtenidas son las siguientes:

- Se han identificado tres clústeres genéticamente distintos de pacientes con sepsis mediante K-means, cada uno asociado con un SNP principal y una enfermedad genética específica:
  - **Clúster 0 (50 sujetos).** Asociado al SNP rs74707084 del gen SYNPR, afectando la comunicación neuronal. Este grupo presenta alta mortalidad y relativamente prolongada ventilación mecánica, sugiriendo complicaciones en la recuperación post-UCI.
  - **Clúster 1 (100 sujetos).** Asociado al SNP rs1575081785 del gen RBSN, que impacta el tráfico de vesículas y la regulación inmune. Muestra la menor mortalidad y ventilación mecánica, reflejando una mejor respuesta clínica con manejo intensivo.
  - **Clúster 2 (37 sujetos).** Asociado con el SNP rs17653532 en el gen PRIM2, crucial para la replicación del ADN. Exhibe las peores variables clínicas, con alta mortalidad y ventilación mecánica más prolongada, debido a defectos en la reparación del ADN y respuesta inmune.
- La utilización de los SHAP *values* para identificar los clústeres ha permitido interpretar de manera más clara y detallada la importancia de los SNPs seleccionados en el modelo predictivo, lo cual es una aportación valiosa para entender los mecanismos genéticos subyacentes a la sepsis.
- La integración de datos genéticos con variables clínicas ha demostrado ser una estrategia efectiva para identificar subgrupos de pacientes con diferentes pronósticos y necesidades

terapéuticas. Este enfoque ha facilitado la comprensión de cómo los diferentes perfiles genéticos pueden influir en la severidad y el pronóstico de la sepsis.
- En comparación con estudios previos que se han centrado principalmente en datos clínicos, nuestro estudio ha resaltado la ventaja de utilizar datos genéticos para la identificación de clústeres, proporcionando una base más sólida para la personalización del tratamiento y el pronóstico de la sepsis.
- El análisis comparativo entre métodos de *clustering* mostró que K-means se comporta mejor que DBSCAN en el contexto de los datos estudiados. K-means logró identificar clústeres bien definidos, mientras que DBSCAN clasificó muchos pacientes como "ruido" debido a la falta de una estructura de densidad clara en los datos empleados.
- Aunque el estudio se ha basado en una muestra de pacientes españoles, los resultados obtenidos sientan las bases para futuras investigaciones que incluyan poblaciones más diversas, permitiendo validar y generalizar los hallazgos.

6.3 Líneas futuras

Durante el desarrollo de este TFG, se han identificado varias y prometedoras líneas de investigación a explorar en el futuro:

- **Ampliación del tamaño muestral.** Incluir un mayor número de pacientes de diferentes regiones y poblaciones para validar y generalizar los resultados obtenidos. Ayudando así a confirmar la robustez de los clústeres identificados y a descubrir posibles nuevas asociaciones genéticas.
- **Análisis de más variables clínicas.** Incluir un mayor número de variables clínicas en el análisis para mejorar la caracterización de los clústeres y proporcionar una visión más completa de los factores que influyen en la sepsis.
- **Desarrollo de modelos predictivos personalizados.** Utilizar los clústeres identificados para desarrollar modelos predictivos que puedan ser aplicados clínicamente para personalizar el tratamiento de los pacientes con sepsis, mejorando así los resultados clínicos.
- **Evaluación de intervenciones terapéuticas.** Realizar estudios clínicos para evaluar la eficacia de intervenciones terapéuticas específicas en los diferentes clústeres de pacientes, adaptando los tratamientos según el perfil genético de cada paciente.
- **Desarrollo de herramientas de diagnóstico.** Crear herramientas software de diagnóstico basadas en los hallazgos genéticos y clínicos para identificar de manera temprana a los pacientes con mayor riesgo de desarrollar sepsis grave y personalizar las intervenciones de manera proactiva.

En resumen, este TFG ha sentado las bases para futuras investigaciones que podrían transformar el tratamiento de la sepsis, acercándonos a un enfoque más personalizado y efectivo en el manejo de esta compleja y grave condición médica.

Referencias

Anna C. Edens Hurst. (2022). *Genética*. https://medlineplus.gov/spanish/ency/article/002048.htm

Arora, J., Mendelson, A. A., & Fox-Robichaud, A. (2023). Sepsis: network pathophysiology and implications for early diagnosis. *American Journal of Physiology - Regulatory Integrative and Comparative Physiology*, *324*(5), R613–R624. https://doi.org/10.1152/AJPREGU.00003.2023

Bouguettaya, A., Yu, Q., Liu, X., Zhou, X., & Song, A. (2015). Efficient agglomerative hierarchical clustering. *Expert Systems with Applications*, *42*(5), 2785–2797. https://doi.org/10.1016/J.ESWA.2014.09.054

Christoph Molnar. (2024). *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*.

Cooper, A., Doyle, O., & Bourke, A. (2021). Supervised Clustering for Subgroup Discovery: An Application to COVID-19 Symptomatology. *Communications in Computer and Information Science*, *1525 CCIS*, 408–422. https://doi.org/10.1007/978-3-030-93733-1_29/TABLES/2

Cui, M. (2020). Introduction to the K-Means Clustering Algorithm Based on the Elbow Method. *Accounting, Auditing and Finance*, *1*(1), 5–8. https://doi.org/10.23977/ACCAF.2020.010102

D'Urso, S., Rajbhandari, D., Peach, E., De Guzman, E., Li, Q., Medland, S. E., Gordon, S. D., Martin, N. G., Ligthart, S., Brown, M. A., Powell, J., McArthur, C., Rhodes, A., Meyer, J., Finfer, S., Myburgh, J., Blumenthal, A., Cohen, J., Venkatesh, B., … Evans, D. M. (2020). Septic Shock: A Genomewide Association Study and Polygenic Risk Score Analysis. *Twin Research and Human Genetics*, *23*(4), 204–213. https://doi.org/10.1017/THG.2020.60

Dahmer, M. K., Randolph, A., Vitali, S., & Quasney, M. W. (2005). Genetic polymorphisms in sepsis. *Pediatric Critical Care Medicine*, *6*(3 SUPPL.). https://doi.org/10.1097/01.PCC.0000161970.44470.C7

DataScientest. (2022). *Machine Learning & Clustering: el algoritmo DBSCAN*. https://datascientest.com/es/machine-learning-clustering-dbscan

Deng, D. (2020). DBSCAN Clustering Algorithm Based on Density. *Proceedings - 2020 7th International Forum on Electrical Engineering and Automation, IFEEA 2020*, 949–953. https://doi.org/10.1109/IFEEA51475.2020.00199

Ding chqding, C. (2004). *K-means Clustering via Principal Component Analysis*.

Dong, L., Li, H., Zhang, S., & Su, L. (2018). Identification of genes related to consecutive trauma-induced sepsis via gene expression profiling analysis. *Medicine*, *97*(15). https://doi.org/10.1097/MD.0000000000010362

Dwivedi, R., Dave, D., Naik, H., Singhal, S., Omer, R., Patel, P., Qian, B., Wen, Z., Shah, T., Morgan, G., & Ranjan, R. (2023). Explainable AI (XAI): Core Ideas, Techniques, and Solutions. *ACM Computing Surveys*, *55*(9). https://doi.org/10.1145/3561048/ASSET/160EFD77-21FC-4899-B447-110C1806F0DB/ASSETS/GRAPHIC/CSUR-2021-0681-F21.JPG

Edwards, D., Forster, J. W., Chagné, D., & Batley, J. (2007). What Are SNPs? *Association Mapping in Plants*, 41–52. https://doi.org/10.1007/978-0-387-36011-9_3

El Naqa, I., Murphy, M. J., El Naqa, I., & Murphy, M. J. (2015). What Is Machine Learning? *Machine Learning in Radiation Oncology*, 3–11. https://doi.org/10.1007/978-3-319-18305-3_1

Engoren, M., Jewell, E. S., Douville, N., Moser, S., Maile, M. D., & Bauer, M. E. (2022). Genetic variants associated with sepsis. *PLOS ONE*, *17*(3), e0265052. https://doi.org/10.1371/JOURNAL.PONE.0265052

Evans, L., Rhodes, A., Alhazzani, W., Antonelli, M., Coopersmith, C. M., French, C., MacHado, F. R., McIntyre, L., Ostermann, M., Prescott, H. C., Schorr, C., Simpson, S., Wiersinga, W. J., Alshamsi, F., Angus, D. C., Arabi, Y., Azevedo, L., Beale, R., Beilman, G., … Levy, M. (2021). Surviving

Sepsis Campaign: International Guidelines for Management of Sepsis and Septic Shock 2021. *Critical Care Medicine*, *49*(11), E1063–E1143. https://doi.org/10.1097/CCM.0000000000005337

Ezugwu, A. E., Ikotun, A. M., Oyelade, O. O., Abualigah, L., Agushaka, J. O., Eke, C. I., & Akinyelu, A. A. (2022). A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. *Engineering Applications of Artificial Intelligence*, *110*, 104743. https://doi.org/10.1016/J.ENGAPPAI.2022.104743

Fatih Karabiber. (2024). *Hierarchical Clustering*. https://www.learndatasci.com/glossary/hierarchical-clustering/

Fleischmann-Struzek, C., Mellhammar, L., Rose, N., Cassini, A., Rudd, K. E., Schlattmann, P., Allegranzi, B., & Reinhart, K. (2020). Incidence and mortality of hospital- and ICU-treated sepsis: results from an updated and expanded systematic review and meta-analysis. *Intensive Care Medicine*, *46*, 1552–1562. https://doi.org/10.1007/s00134-020-06151-x

Francisco Sanz. (2024). *Algoritmo K-Means Clustering – aplicaciones y desventajas*. https://www.themachinelearners.com/k-means/

Genga, K. R., & Russell, J. A. (2017). Update of Sepsis in the Intensive Care Unit. *Journal of Innate Immunity*, *9*(5), 441–455. https://doi.org/10.1159/000477419

Ghosal, A., Nandy, A., Das, A. K., Goswami, S., & Panday, M. (2020). A Short Review on Different Clustering Techniques and Their Applications. *Advances in Intelligent Systems and Computing*, *937*, 69–83. https://doi.org/10.1007/978-981-13-7403-6_9/TABLES/2

Guzmán Cotaya, R., Baeza Bastarrachea, R., & Espinosa Padilla, S. E. (2021). Neutropenia congénita. *Alergia, Asma e Inmunología Pediátricas*, *30*(1), 24–27. https://doi.org/10.35366/100114

Hernandez-Beeftink, T., Guillen-Guio, B., Lorenzo-Salazar, J. M., Corrales, A., Suarez-Pajes, E., Feng, R., Rubio-Rodríguez, L. A., Paynton, M. L., Cruz, R., García-Laorden, M. I., Prieto-González, M., Rodríguez-Pérez, A., Carriedo, D., Blanco, J., Ambrós, A., González-Higueras, E., Espinosa, E., Muriel, A., Tamayo, E., … Flores, C. (2022). A genome-wide association study of survival in patients with sepsis. *Critical Care*, *26*(1), 1–10. https://doi.org/10.1186/S13054-022-04208-5/TABLES/2

IBM. (2024). *¿Qué es la inteligencia artificial (IA)?* https://www.ibm.com/mx-es/topics/artificial-intelligence

Jang, J. Y., Yoo, G., Lee, T., Uh, Y., & Kim, J. (2022). Identification of the robust predictor for sepsis based on clustering analysis. *Scientific Reports 2022 12:1*, *12*(1), 1–8. https://doi.org/10.1038/s41598-022-06310-8

Jarczak, D., Kluge, S., & Nierhaus, A. (2021). Sepsis—Pathophysiology and Therapeutic Concepts. *Frontiers in Medicine*, *8*, 628302. https://doi.org/10.3389/FMED.2021.628302/BIBTEX

Johnsen, P. V., Riemer-Sørensen, S., DeWan, A. T., Cahill, M. E., & Langaas, M. (2021). A new method for exploring gene–gene and gene–environment interactions in GWAS with tree ensemble methods and SHAP values. *BMC Bioinformatics*, *22*(1), 1–29. https://doi.org/10.1186/S12859-021-04041-7/FIGURES/12

Khan, K., Rehman, S. U., Aziz, K., Fong, S., Sarasvady, S., & Vishwa, A. (2014). DBSCAN: Past, present and future. *5th International Conference on the Applications of Digital Information and Web Technologies, ICADIWT 2014*, 232–238. https://doi.org/10.1109/ICADIWT.2014.6814687

Knaus, P., Marquèze-Pouey, B., Scherer, H., & Betzt, H. (1990). Synaptoporin, a novel putative channel protein of synaptic vesicles. *Neuron*, *5*(4), 453–462. https://doi.org/10.1016/0896-6273(90)90084-S

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature 2015 521:7553*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Lifeder. (2019). *Distancia euclidiana: concepto, fórmula, cálculo, ejemplo.* https://www.lifeder.com/distancia-euclidiana/

Liu, X., Zhang, P., Bao, Y., Han, Y., Wang, Y., Zhang, Q., Zhan, Z., Meng, J., Li, Y., Li, N., Zhang, W. J., & Cao, X. (2013). Zinc finger protein ZBTB20 promotes toll-like receptor-triggered innate immune responses by repressing IκBα gene transcription. *Proceedings of the National Academy of Sciences*

of the United States of America, *110*(27), 11097–11102. https://doi.org/10.1073/PNAS.1301257110/SUPPL_FILE/PNAS.201301257SI.PDF

López Herrero, R., Vaquerizo Villar, F., Hernández Beeftink, T., Bardají Carrillo, M., Gómez Sánchez, E., & Tamayo, E. (2024). Identificación de genes de susceptibilidad a la sepsis mediante un enfoque de inteligencia artificial. *XXXVII Congreso de La Sociedad Española de Anestesiología, Reanimación y Terapeútica Del Dolor (SEDAR)*.

Lundberg, S. M., & Lee, S. I. (2017). A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems*, *2017-December*, 4766–4775. https://arxiv.org/abs/1705.07874v2

Maslove, D. M., Tang, B. M., & McLean, A. S. (2012). Identification of sepsis subtypes in critically ill adults using gene expression profiling. *Critical Care*, *16*(5), 1–12. https://doi.org/10.1186/CC11667/TABLES/3

Mihaljevic, O., Zivancevic-Simonovic, S., Jovanovic, D., Drakulic, S. M., Vukajlovic, J. T., Markovic, A., Pirkovic, M. S., Srejovic, I., Jakovljevic, V., & Milosevic-Djordjevic, O. (2023). Oxidative stress and DNA damage in critically ill patients with sepsis. *Mutation Research/Genetic Toxicology and Environmental Mutagenesis*, *889*, 503655. https://doi.org/10.1016/J.MRGENTOX.2023.503655

NIH. (2019). *ADN*. https://www.genome.gov/es/about-genomics/fact-sheets/acido-desoxirribonucleico

NIH. (2024). *Gen*. https://www.genome.gov/es/genetics-glossary/Gen

Papin, G., Bailly, S., Dupuis, C., Ruckly, S., Gainnier, M., Argaud, L., Azoulay, E., Adrie, C., Souweine, B., Goldgran-Toledano, D., Marcotte, G., Gros, A., Reignier, J., Mourvillier, B., Forel, J. M., Sonneville, R., Dumenil, A. S., Darmon, M., Garrouste-Orgeas, M., … Letrou, S. (2021). Clinical and biological clusters of sepsis patients using hierarchical clustering. *PLOS ONE*, *16*(8), e0252793. https://doi.org/10.1371/JOURNAL.PONE.0252793

Pei, F., Gu, B., Miao, S. M., Guan, X. D., & Wu, J. F. (2024). Clinical practice of sepsis-induced immunosuppression: Current immunotherapy and future options. *Chinese Journal of Traumatology*, *27*(2), 63–70. https://doi.org/10.1016/J.CJTEE.2023.11.001

Qiao, W., Akhter, N., Fang, X., Maximova, T., Plaku, E., & Shehu, A. (2018). From mutations to mechanisms and dysfunction via computation and mining of protein energy landscapes 06 Biological Sciences 0601 Biochemistry and Cell Biology. *BMC Genomics*, *19*(7), 1–13. https://doi.org/10.1186/S12864-018-5024-Z/TABLES/1

Rakhra, G., & Rakhra, G. (2021). Zinc finger proteins: insights into the transcriptional and post transcriptional regulation of immune response. *Molecular Biology Reports*, *48*(7), 5735–5743. https://doi.org/10.1007/S11033-021-06556-X

Reynolds, A. P., Richards, G., & Rayward-Smith, V. J. (2004). The Application of K-Medoids and PAM to the Clustering of Rules. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 3177, pp. 173–178). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-28651-6_25

RODRIGO QUEIRÓS CONCEIÇÃO. (2023). *SUPERVISED CLUSTERING WITH SHAP VALUES*.

Rosier, F., Brisebarre, A., Dupuis, C., Baaklini, S., Puthier, D., Brun, C., Pradel, L. C., Rihet, P., & Payen, D. (2021). Genetic Predisposition to the Mortality in Septic Shock Patients: From GWAS to the Identification of a Regulatory Variant Modulating the Activity of a CISH Enhancer. *International Journal of Molecular Sciences 2021, Vol. 22, Page 5852*, *22*(11), 5852. https://doi.org/10.3390/IJMS22115852

Sanabria-Navarro, J. R., Silveira-Pérez, Y., Pérez-Bravo, D. D., & de-Jesús-Cortina-Núñez, M. (2023). Incidencias de la inteligencia artificial en la educación contemporánea. *Oxbridge Publishing House*, *31*(77), 97–107. https://doi.org/10.3916/C77-2023-08

Sarma, A., Calfee, C. S., & Ware, L. B. (2020). Biomarkers and Precision Medicine: State of the Art. *Critical Care Clinics*, *36*(1), 155–165. https://doi.org/10.1016/J.CCC.2019.08.012

Saxena, A., Prasad, M., Gupta, A., Bharill, N., Patel, O. P., Tiwari, A., Er, M. J., Ding, W., & Lin, C. T.

(2017). A review of clustering techniques and developments. *Neurocomputing*, *267*, 664–681. https://doi.org/10.1016/J.NEUCOM.2017.06.053

scikit-learn. (2024). *Selecting the number of clusters with silhouette analysis on KMeans clustering*. https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

Seymour, C. W., Kennedy, J. N., Wang, S., Chang, C. C. H., Elliott, C. F., Xu, Z., Berry, S., Clermont, G., Cooper, G., Gomez, H., Huang, D. T., Kellum, J. A., Mi, Q., Opal, S. M., Talisa, V., Van Der Poll, T., Visweswaran, S., Vodovotz, Y., Weiss, J. C., … Angus, D. C. (2019). Derivation, Validation, and Potential Treatment Implications of Novel Clinical Phenotypes for Sepsis. *JAMA*, *321*(20), 2003–2017. https://doi.org/10.1001/JAMA.2019.5791

Sinaga, K. P., & Yang, M. S. (2020). Unsupervised K-means clustering algorithm. *IEEE Access*, *8*, 80716–80727. https://doi.org/10.1109/ACCESS.2020.2988796

Singer, M., Deutschman, C. S., Seymour, C. W., Shankar-Hari, M., Annane, D., Bauer, M., Bellomo, R., Bernard, G. R., Chiche, J.-D., Coopersmith, C. M., Hotchkiss, R. S., Levy, M. M., Marshall, J. C., Martin, G. S., Opal, S. M., Rubenfeld, G. D., van der Poll, T., Vincent, J.-L., & Angus, D. C. (2016). The Third International Consensus Definitions for Sepsis and Septic Shock (Sepsis-3). *JAMA*, *315*(8), 801. https://doi.org/10.1001/jama.2016.0287

Skibsted, S., Bhasin, M. K., Aird, W. C., & Shapiro, N. I. (2013). Bench-to-bedside review: Future novel diagnostics for sepsis - a systems biology approach. *Critical Care*, *17*(5), 1–15. https://doi.org/10.1186/CC12693/FIGURES/4

Srzić, I., Adam, V. N., & Pejak, D. T. (2022). SEPSIS DEFINITION: WHAT'S NEW IN THE TREATMENT GUIDELINES. *Acta Clin Croat (Suppl. 1)*, *61*, 2022. https://doi.org/10.20471/acc.2022.61.s1.11

Stanski, N. L., & Wong, H. R. (2019). Prognostic and predictive enrichment in sepsis. *Nature Reviews Nephrology 2019 16:1*, *16*(1), 20–31. https://doi.org/10.1038/s41581-019-0199-3

Stewart, G., & Al-Khassaweneh, M. (2022). An Implementation of the HDBSCAN* Clustering Algorithm. *Applied Sciences 2022, Vol. 12, Page 2405*, *12*(5), 2405. https://doi.org/10.3390/APP12052405

van den Berg, M., van Beuningen, F. E., ter Maaten, J. C., & Bouma, H. R. (2022). Hospital-related costs of sepsis around the world: A systematic review exploring the economic burden of sepsis. *Journal of Critical Care*, *71*. https://doi.org/10.1016/J.JCRC.2022.154096

van der Poll, T., & Opal, S. M. (2008). Host–pathogen interactions in sepsis. *The Lancet Infectious Diseases*, *8*(1), 32–43. https://doi.org/10.1016/S1473-3099(07)70265-7

Wiersinga, W. J., Leopold, S. J., Cranendonk, D. R., & van der Poll, T. (2014). Host innate immune responses to sepsis. *Virulence*, *5*(1), 36–44. https://doi.org/10.4161/VIRU.25436

World Health Organization. (2024). *Sepsis*. https://www.who.int/es/news-room/fact-sheets/detail/sepsis

Zhang, Z., Pan, Q., Ge, H., Xing, L., Hong, Y., & Chen, P. (2020). Deep learning-based clustering robustly identified two classes of sepsis with both prognostic and predictive values. *EBioMedicine*, *62*, 103081. https://doi.org/10.1016/j.ebiom.2020.103081

Anexo A. Glosario de siglas

A	Adenina
ADN	Ácido desoxirribonucleico
BNADN	Banco Nacional de ADN
C	Citosina
CNN	*Convolutional Neural Network*
DL	*Deep Learning*
DBSCAN	*Density-Based Spatial Clustering of Applications with Noise*
Eps	Distancia máxima entre dos puntos para que se consideren en el mismo clúster
G	Guanina
GMM	*Gaussian Mixture Models*
GWAS	*Genome Wide Association Studies*
HDBSCAN	*Hierarchical Density-Based Spatial Clustering of Applications with Noise*
HC	*Hierarchical Clustering*
IA	Inteligencia Artificial
K	Número de clústeres
minpts	Número mínimo de puntos necesarios para formar un clúster
ML	*Machine Learning*
NAN	*Not A Number*
PAM	*Partitioning Around Medoids*
PCA	*Principal Component Analysis*
qSOFA	*Quick Sequential Organ Failure Assessment*
RIC	Rango Intercuartílico
SNP	*Single Nucleotide Polymorphism*
SHAP	*SHapley Additive exPlanations*
SOFA	*Sequential Organ Failure Assessment*
SSC	*Surviving Sepsis Campaign*
T	Timina
TFG	Trabajo de Fin de Grado
UCI	Unidad de Cuidados Intensivos
WCSS	*Within-Cluster Sum of Squares*
WHO	*World Health Organization*
XAI	*eXplainable Artificial Intelligence*

Anexo B. Código desarrollado

1. Importe de librerías y definición de funciones

```
import scipy.io
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans, DBSCAN
from sklearn.metrics import silhouette_score
from sklearn.decomposition import PCA
import pandas as pd
import re
from scipy.stats import kruskal, chi2_contingency
import os


# Método del codo (Elbow Method)
def elbow_method(X, max_clusters):
    distortions = []
    for i in range(1, max_clusters + 1):
        kmeans = KMeans(n_clusters=i)
        kmeans.fit(X)
        distortions.append(kmeans.inertia_)
    return distortions

# Método de la silueta (Silhouette Method)
def silhouette_method(X, max_clusters):
    silhouette_scores = []
    for i in range(2, max_clusters + 1):
        kmeans = KMeans(n_clusters=i)
        cluster_labels = kmeans.fit_predict(X)
        silhouette_avg = silhouette_score(X, cluster_labels)
        silhouette_scores.append(silhouette_avg)
    return silhouette_scores


# Algoritmo K-Means
def kmeans_clustering(X, n_clusters):
    kmeans = KMeans(n_clusters=n_clusters)
    kmeans.fit(X)
    return kmeans.labels_

# Algoritmo DBSCAN
def dbscan_clustering(X, eps, min_samples):
    dbscan = DBSCAN(eps=eps, min_samples=min_samples)
    return dbscan.fit_predict(X)

# Visualización
def visualize_clusters(X, labels, title):
    plt.scatter(X[:,0], X[:,1], c=labels, cmap='viridis')
    plt.title(title)
    plt.xlabel('Componente principal 0')
    plt.ylabel('Componente principal 1')
    plt.show()

def visualize_clusters_comparacion(X, labels, title,id1,id2):
```

```
    plt.scatter(X[:, id1], X[:, id2], c=labels, cmap='viridis')
    plt.title(title)
    plt.xlabel(f'Componente principal {id1}')
    plt.ylabel(f'Componente principal {id2}')
    plt.show()

def visualize_clusters_n(X, labels, cluster_names, title):
    plt.scatter(X[:, 0], X[:, 1], c=labels, cmap='viridis')
    plt.title(title)
    plt.xlabel('Componente principal 0')
    plt.ylabel('Componente principal 1')

    # Mostrar nombres de los clusters en el gráfico
    for cluster_label, cluster_name in cluster_names.items():
        cluster_indices = np.where(labels == cluster_label)[0]
        cluster_center = np.mean(X[cluster_indices], axis=0)
        plt.text(cluster_center[0], cluster_center[1], cluster_name,
fontsize=12, ha='center', va='center', color='black',
fontweight='bold')

    plt.show()
# Función para extraer los IDs de las rutas de archivo
def extract_ids(file_paths):
    ids = []
    for file_path in file_paths:
        match = re.search(patron_id, file_path)
        if match:
            ids.append(match.group(1))  # Agregar el ID encontrado
    return ids
# Definir la función para convertir 'time_outc_mortdays' en una
variable binaria

def convertir_time_outc_mortdays_a_binario(valor):
    if np.isnan(valor) or valor > 91:
        return 0
    else:
        return 1

def muerte_uci(valor_mortdays, valor_icudays):
    if np.isnan(valor_icudays):
        return 0
    else:
        return 1 if valor_mortdays == valor_icudays else 0
# Función para calcular mediana y IQR para variables clínicas
def calcular_mediana_iqr_variables_clinicas(datos, variable):
    valores = datos[variable]
    mediana = valores.median()
    quartil_25 = valores.quantile(0.25)
    quartil_75 = valores.quantile(0.75)
    iqr = [quartil_25, quartil_75]
    return mediana, iqr

def calcular_proporcion(datos, variable, valor_deseado):
    proporcion = datos[variable].value_counts(normalize=True)
    proporcion_seleccionada = proporcion[valor_deseado] if
valor_deseado in proporcion.index else 0
    return proporcion_seleccionada
```

```
def validar_comorbilidad(valor):
    if valor in [0, 1]:
        return valor
    else:
        return np.nan
```

2. Se cargan las bases de datos y se definen las variables adecuadamente

```
# Cargar datos .mat
mat = scipy.io.loadmat('data_TFG_David_Segovia_all_subjects.mat')
matriz_completa=mat['matrix_shap_values']
matriz=mat['matrix_shap_values_1']
shap_values = mat['global_shap_values_1']
SNPs = mat['SNPS ID']
#Me quedo con los 20 genes mas importantes
shap_ordenados=np.argsort(shap_values)
shap_ordenados_girados=np.flip(shap_ordenados).squeeze()


datos=matriz[:,shap_ordenados_girados]

# Obtener los índices de los 20 genes seleccionados
SNPS_genes_seleccionados = SNPs[shap_ordenados_girados]

name_path_snp_genes='Axiom_SpainBA.na35.annot.csv'
unique_genes=SNPS_genes_seleccionados.tolist()
file_snp_genes = open(name_path_snp_genes, 'r')
unique_genes2=[x.strip(' ') for x in unique_genes]

ax_genes = []
rs_genes = []
chr_genes = []
pos_genes = []
gene_genes = []
i=0
while True:
    # Get next line from file
    line = file_snp_genes.readline()
    i = i + 1
    # if line is empty
    # end of file is reached
    if not line:
        break
    else:
        if i > 20:
            line_arr = np.array(line.split(","))
            if line_arr[0][1:] in unique_genes2:
                ax_genes.append(line_arr[0].replace('"', ''))
                rs_genes.append(line_arr[2].replace('"', ''))
                chr_genes.append(line_arr[4].replace('"', ''))
                pos_genes.append(int(line_arr[5].replace('"', '')))
file_snp_genes.close()

# Crear un diccionario para mapear los genes de ax_genes a sus valores
en rs_genes, chr_genes y pos_genes
gen_to_rs = dict(zip(ax_genes, rs_genes))
```

```
gen_to_chr = dict(zip(ax_genes, chr_genes))
gen_to_pos = dict(zip(ax_genes, pos_genes))

# Crear una lista ordenada de rs_genes según el orden de
SNPS_genes_seleccionados
rs_genes_ordenado = []
for gen in unique_genes2:
    rs_value = gen_to_rs[gen]
    if rs_value == "---":
        # chr_value = gen_to_chr[gen]
        # pos_value = gen_to_pos[gen]
        rs_value = "rs1575081785"
    rs_genes_ordenado.append(rs_value)
identificadores_genes_seleccionados=rs_genes_ordenado

# Mostrar los SNPs de los genes seleccionados
print("rs de los genes seleccionados:")
print(identificadores_genes_seleccionados)


# Calcular la media de los SHAP values para cada SNP
mean_shap_values = datos.mean(axis=0)

# Convertir la lista de IDs de SNPs en un DataFrame para facilitar el
manejo
snp_ids = pd.Series(identificadores_genes_seleccionados)

# Crear un DataFrame para facilitar la visualización
shap_summary_df = pd.DataFrame({
    'SNP_ID': snp_ids,
    'Mean_SHAP_Value': mean_shap_values
})

# Seleccionar solo los primeros 20 SNPs
top_20_shap_summary_df = shap_summary_df.head(20)

# Crear el histograma
plt.figure(figsize=(12, 6))
plt.bar(top_20_shap_summary_df['SNP_ID'],
top_20_shap_summary_df['Mean_SHAP_Value'])
plt.xlabel('ID del SNP')
plt.ylabel('Media de los SHAP values')
plt.title('Media de los SHAP values para los primeros 20 SNPs')
plt.xticks(rotation=90, fontsize=15)  # Rotar etiquetas del eje x y
cambiar el tamaño de la fuente
plt.tight_layout()
plt.show()



#Mostrar los valores de 20 SNPs de 4 sujetos
# Seleccionar los primeros 20 SNPs
num_snps_to_plot = 20
top_snps = identificadores_genes_seleccionados[:num_snps_to_plot]

# Crear una figura con 4 subgráficas (una para cada sujeto)
```

```
fig, axes = plt.subplots(2, 2, figsize=(14, 10), sharex=True,
sharey=True)
axes = axes.flatten()  # Para facilitar la indexación

# Especificar los sujetos a plotear
sujetos_a_plotear = [0, 5, 2, 3]  # Reemplazar el sujeto 2 (índice 1)
por el sujeto 5 (índice 4)

# Plotear los valores de SHAP para los sujetos seleccionados
for i, sujeto in enumerate(sujetos_a_plotear):
    # Obtener los valores de SHAP para el sujeto y los primeros 20
SNPs
    subject_shap_values = datos[sujeto, :num_snps_to_plot]

    # Crear el gráfico de barras para el sujeto
    axes[i].bar(top_snps, subject_shap_values)
    axes[i].set_title(f'Sujeto {sujeto + 1}')
    axes[i].set_xlabel('ID del SNP')
    axes[i].set_ylabel('Valor de SHAP')
    axes[i].tick_params(axis='x', rotation=90, labelsize=15)  # Rotar
y cambiar el tamaño de las etiquetas del eje x

# Ajustar el diseño para que no se sobrepongan
plt.tight_layout()
plt.show()


names_testx=mat['names_testx']
pca=PCA()
pca.fit(datos)
pca.explained_variance_ratio_
explained_variance_ratio = pca.explained_variance_ratio_

plt.figure(figsize=(10,8))
plt.plot(range(1, len(explained_variance_ratio) +
1),explained_variance_ratio.cumsum(), marker='o', linestyle='--')
plt.title('Varianza de los componentes explicada')
plt.xlabel('Numero de componentes')
plt.ylabel('Varianza acumulada')

#pca=PCA(n_components=20)
pca.fit(datos)

pca.transform(datos)
scores_pca=pca.transform(datos)



# Patrón de expresión regular para extraer los IDs de las rutas de
archivo
patron_id =
r'D:/Fernando/Colaboraciones/EduardoTamayo/AssocTamara_.+?/(.+)\.mat'
```

```
names_txt =
["D:/Fernando/Colaboraciones/EduardoTamayo/AssocTamara_p_values_5e-
3/BNADN_10448850_A10_530.mat", ...]

# Extraer los IDs de las rutas de archivo
ids = extract_ids(names_testx)

# Cargar los datos de control y de no control
datos_control =
pd.read_excel('SEPSIS_data_id_age_sex_phenotype_category_sorted_contro
ls_v4.xlsx')
datos_no_control =
pd.read_excel('SEPSIS_data_id_age_sex_phenotype_category_sorted_cases_
v4.xlsx')

# Filtrar los datos de control
datos_control_filtrados = datos_control[datos_control['ID'].isin(ids)]
# Filtrar los datos de no control (sujetos de sepsis)
datos_sepsis_filtrados =
datos_no_control[datos_no_control['ID'].isin(ids)]
variables_clinicas = {
    'Edad': 'Age',
    'Sexo': 'Sex_imputed',
    'Tipo de Sepsis': 'Sepsis_Category',
    'Mortalidad en 90 días': 'time_outc_mortdays',
    'Neutrófilos': 'meas_neut',
    'Linfocitos': 'meas_linf',
    'Glóbulos Blancos': 'meas_wbc',
    'Creatinina': 'meas_crea'
}
variables_mortalidad = {
    'Dias en UCI': 'time_outc_icudays',
    'Dias en Hospital': 'time_outc_hospdays',
    'Dias Ventilacion Mecanica': 'time_outc_ventdays',
}

# Iterar sobre las variables clínicas y aplicar la función a
'time_outc_mortdays' si está presente
for nombre_variable, variable_columna in variables_clinicas.items():
    if variable_columna == 'time_outc_mortdays':
        # Aplicar la función a la columna correspondiente
        datos_sepsis_filtrados[variable_columna + '_binario'] =
datos_sepsis_filtrados[variable_columna].apply(convertir_time_outc_mor
tdays_a_binario)



# Aplicar la función a las columnas correspondientes
datos_sepsis_filtrados['muerte_en_uci'] =
datos_sepsis_filtrados.apply(lambda x:
muerte_uci(x['time_outc_mortdays'], x['time_outc_icudays']), axis=1)

sujetos_sepsis = datos_sepsis_filtrados['ID'].tolist()

# Ahora tienes los datos filtrados para control y sujetos de sepsis
utilizando los IDs extraídos de las rutas de archivo
```

3. Implementación de los algoritmos K-means y DBSCAN

```
# Ejemplo de uso
max_clusters = 10

# Método del codo
distortions = elbow_method(scores_pca, max_clusters)
plt.figure()
plt.plot(range(1, max_clusters + 1), distortions, marker='o')
plt.xlabel('Number of clusters')
plt.ylabel('Distortion')
plt.title('Elbow Method')
plt.show()

# Método de la silueta
plt.figure()
silhouette_scores = silhouette_method(scores_pca, max_clusters)
plt.plot(range(2, max_clusters + 1), silhouette_scores, marker='o')
plt.xlabel('Number of clusters')
plt.ylabel('Silhouette Score')
plt.title('Silhouette Method')
plt.show()
# Obtener número óptimo de clusters
optimal_clusters_silhouette =
3#optimal_clusters_silhouette(scores_pca, max_clusters)
optimal_clusters_elbow = 3 #optimal_clusters_elbow(scores_pca,
max_clusters)

print("Número óptimo de clusters (Silueta):",
optimal_clusters_silhouette)
print("Número óptimo de clusters (Codo):", optimal_clusters_elbow)

# Clustering usando K-Means
labels_kmeans_silhouette = kmeans_clustering(scores_pca,
optimal_clusters_silhouette)
labels_kmeans_elbow = kmeans_clustering(scores_pca,
optimal_clusters_elbow)
labels_kmeans_elbow_todos=labels_kmeans_elbow
# Clustering usando DBSCAN
eps = 0.15 # Ajusta este valor 0.1 para50 componentes, 0.03 para 20
min_samples = 5  # Ajusta este valor
labels_dbscan = dbscan_clustering(scores_pca, eps, min_samples)
np.unique(labels_dbscan)

# Visualización de resultados
visualize_clusters(scores_pca, labels_kmeans_silhouette, 'K-Means')
visualize_clusters_comparacion(scores_pca, labels_kmeans_elbow, 'K-
Means',0,1)
visualize_clusters_comparacion(scores_pca, labels_kmeans_elbow, 'K-
Means',0,2)
visualize_clusters_comparacion(scores_pca, labels_kmeans_elbow, 'K-
Means',1,2)
visualize_clusters(scores_pca, labels_dbscan, 'DBSCAN')

# Asociar los sujetos a los clústeres
sujetos_por_cluster_s = {}
for i, cluster in enumerate(labels_kmeans_silhouette):
```

```
    id_sujeto = sujetos_sepsis[i]  # Obtener el ID del sujeto
    if cluster not in sujetos_por_cluster_s:
        sujetos_por_cluster_s[cluster] = []
    sujetos_por_cluster_s[cluster].append(id_sujeto)
sujetos_por_cluster_c = {}
for i, cluster in enumerate(labels_kmeans_elbow):
    id_sujeto = sujetos_sepsis[i]  # Obtener el ID del sujeto
    if cluster not in sujetos_por_cluster_c:
        sujetos_por_cluster_c[cluster] = []
    sujetos_por_cluster_c[cluster].append(id_sujeto)
sujetos_por_cluster_d = {}
for i, cluster in enumerate(labels_dbscan):
    id_sujeto = sujetos_sepsis[i]  # Obtener el ID del sujeto
    if cluster not in sujetos_por_cluster_d:
        sujetos_por_cluster_d[cluster] = []
    sujetos_por_cluster_d[cluster].append(id_sujeto)


# Asignar nombres a los clústeres manualmente
nombres_clusters_s = {   0: "Clúster 0", 1: "Clúster 1",2: "Clúster
2"}
nombres_clusters_c = {   0: "Clúster 0", 1: "Clúster 1", 2: "Clúster
2"}
nombres_clusters_d = {   0: "Clúster 0", 1: "Clúster 1", 2: "Clúster
2", 3: "Clúster 3",-1: "Ruido"}

visualize_clusters_n(scores_pca, labels_kmeans_silhouette,
nombres_clusters_s, 'K-Means')
visualize_clusters_n(scores_pca, labels_kmeans_elbow,
nombres_clusters_c, 'K-Means')
visualize_clusters_n(scores_pca, labels_dbscan, nombres_clusters_d,
'DBSCAN')




clusters =sujetos_por_cluster_c

# Mostrar los IDs por cluster
for cluster, ids in sujetos_por_cluster_c.items():
    print(f"Cluster {cluster}: {ids}")
```

4. Cálculo de variables estadísticas y representación del top 50 SNPs por clúster

```
# Variables clínicas a considerar
variables_clinicas = {
    'Edad': 'Age',
    'Sexo': 'Sex_imputed',
    'Tipo de Sepsis': 'Sepsis_Category',
    'Mortalidad en 90 días': 'time_outc_mortdays_binario',
    'Neutrófilos': 'meas_neut',
    'Linfocitos': 'meas_linf',
    'Glóbulos Blancos': 'meas_wbc',
    'Creatinina': 'meas_crea'
}

# Valores deseados para cada variable
```

```
valores_deseados = {
    'Sexo': 1,  # Por ejemplo, se desea la proporción de valor 1 para
la variable Sexo
    'Tipo de Sepsis': 3,  # Por ejemplo, se desea la proporción de
valor 'A' para la variable Tipo de Sepsis
    'Mortalidad en 90 días': 1,  # Por ejemplo, se desea la proporción
de valor 0 para la variable Mortalidad en 90 días
}

# Directorio donde se almacenarán los resultados
directorio_resultados = "resultados_clusters"

# Crear el directorio si no existe
if not os.path.exists(directorio_resultados):
    os.makedirs(directorio_resultados)

# Inicializar diccionarios para almacenar los resultados
resultados_importantes = {}
resultados_resto = {}

# Mostrar los resultados por cluster y guardarlos en un archivo Excel
for cluster, sujetos in clusters.items():
    data_importantes = {}
    data_resto = {}
    for nombre_variable, variable in variables_clinicas.items():
        if variable in ['Sex_imputed', 'Sepsis_Category',
'time_outc_mortdays_binario']:
            proporcion_seleccionada =
calcular_proporcion(datos_sepsis_filtrados.loc[datos_sepsis_filtrados[
'ID'].isin(sujetos), :], variable,
valores_deseados.get(nombre_variable, 0))
            data_importantes[nombre_variable] =
f"{proporcion_seleccionada}"
        else:
            mediana, iqr =
calcular_mediana_iqr_variables_clinicas(datos_sepsis_filtrados.loc[dat
os_sepsis_filtrados['ID'].isin(sujetos), :], variable)
            data_resto[nombre_variable] = f"{mediana},{iqr}"

    # Almacenar los resultados en los diccionarios por cluster
    resultados_importantes[cluster] = data_importantes
    resultados_resto[cluster] = data_resto

# Crear los DataFrames con los resultados
df_importantes = pd.DataFrame(resultados_importantes)
df_resto = pd.DataFrame(resultados_resto)

# Guardar los DataFrames en un archivo Excel
archivo_excel_importantes = os.path.join(directorio_resultados,
"resultados_importantes.xlsx")
archivo_excel_resto = os.path.join(directorio_resultados,
"resultados_resto.xlsx")
df_importantes.to_excel(archivo_excel_importantes,
float_format="%.3f")
df_resto.to_excel(archivo_excel_resto, float_format="%.3f")
```

```
variables_mortalidad = {
    'Dias en UCI': 'time_outc_icudays',
    'Dias en Hospital': 'time_outc_hospdays',
    'Dias Ventilacion Mecanica': 'time_outc_ventdays',
    'Muerte en UCI': 'muerte_en_uci'
}

# Proporción deseada para la variable "Muerte en UCI" (1)
proporcion_deseada_muerte_uci = 1

resultados_dias = {}

for cluster, sujetos in clusters.items():
    data_dias = {}
    for nombre_variable, variable in variables_mortalidad.items():
        if variable == 'muerte_en_uci':
            proporcion_seleccionada =
calcular_proporcion(datos_sepsis_filtrados.loc[datos_sepsis_filtrados[
'ID'].isin(sujetos), :], variable, proporcion_deseada_muerte_uci)
            data_dias[nombre_variable] = f"{proporcion_seleccionada}"
        else:
            mediana, iqr =
calcular_mediana_iqr_variables_clinicas(datos_sepsis_filtrados.loc[dat
os_sepsis_filtrados['ID'].isin(sujetos), :], variable)
            data_dias[nombre_variable] = f"{mediana},{iqr}"
    resultados_dias[cluster] = data_dias

df_dias = pd.DataFrame(resultados_dias)
archivo_excel_dias = os.path.join(directorio_resultados,
"resultados_dias.xlsx")
df_dias.to_excel(archivo_excel_dias, float_format="%.3f")
#Calcular SNPs mas importantes para cada cluster
# Paso 1: Calcular la media de los SHAP values para cada gen en cada
cluster
media_shap_por_cluster = {}
for cluster in set(labels_kmeans_elbow):
    indices_cluster = np.where(labels_kmeans_elbow == cluster)[0]  #
Índices de sujetos en el cluster
    shap_values_cluster = datos[indices_cluster]  # SHAP values de los
sujetos en el cluster
    media_shap_por_gen = np.mean(shap_values_cluster, axis=0)  # Media
de SHAP values por gen
    media_shap_por_cluster[cluster] = media_shap_por_gen

# Paso 2: Encontrar los genes más importantes para cada cluster
genes_importantes_por_cluster = {}
indices_genes_importantes_cluster={}
for cluster, media_shap in media_shap_por_cluster.items():
    indices_genes_importantes = np.argsort(media_shap)[::-1]  #
Índices de los 5 genes más importantes
    indices_genes_importantes_cluster[cluster]=
indices_genes_importantes
    nombres_genes_importantes =
[identificadores_genes_seleccionados[i] for i in
indices_genes_importantes]  # Nombres de los genes más importantes
    genes_importantes_por_cluster[cluster] = nombres_genes_importantes
```

```
# Paso 3: Mostrar los resultados
for cluster, nombres_genes in genes_importantes_por_cluster.items():
    print(f"Cluster {cluster}: Genes más importantes -
{nombres_genes[:5]}")



# Definir el número de genes más importantes
num_genes_importantes = 3761


# Directorio para almacenar los shap values de cada cluster
shap_values_por_cluster = {}

# Iterar sobre los clusters
for cluster in set(labels_kmeans_elbow):
    # Obtener los índices de los sujetos en este cluster
    indices_cluster = np.where(labels_kmeans_elbow == cluster)[0]

    # Obtener los shap values de los sujetos en este cluster
    shap_values_cluster_2 = datos[indices_cluster]

    # Almacenar los shap values en el directorio
    shap_values_por_cluster[cluster] = shap_values_cluster_2
shap_values_medias_ordenadas = {}

# Iterar sobre los clusters
for cluster, shap_values_cluster in shap_values_por_cluster.items():
    # Calcular la media de cada gen en este cluster
    media_por_gen = np.mean(shap_values_cluster, axis=0)

    # Obtener una lista de tuplas (gen, media) y ordenarlas de mayor a
menor
    medias_ordenadas = sorted(zip(range(len(media_por_gen)),
media_por_gen), key=lambda x: x[1], reverse=True)

    # Almacenar las medias ordenadas en este cluster
    shap_values_medias_ordenadas[cluster] = medias_ordenadas



# Iterar sobre cada cluster
for cluster, nombres_genes_importantes in
genes_importantes_por_cluster.items():
    # Crear una nueva figura para cada cluster
    plt.figure(figsize=(15, 7))

    # Obtener los 50 primeros valores de SHAP de cada cluster
    primeros_50_shap_values =
shap_values_medias_ordenadas[cluster][:50]

    # Obtener los nombres de los genes correspondientes
    nombres_genes_50_importantes =
[identificadores_genes_seleccionados[indice] for indice, _ in
primeros_50_shap_values]
```

```
    # Obtener los valores de SHAP correspondientes
    valores_shap_50_importantes = [valor for _, valor in
primeros_50_shap_values]

    # Plotear un histograma para los valores de SHAP
    plt.bar(nombres_genes_50_importantes, valores_shap_50_importantes,
color='skyblue')

    plt.title(f'Cluster {cluster}: Top 50 genes más importantes')
    plt.xlabel('Genes')
    plt.ylabel('Media de SHAP Values')
    plt.xticks(rotation=90, fontsize=15)  # Cambia el tamaño de la
fuente aquí
    plt.grid(axis='y')
    plt.tight_layout()


    plt.show()



pvalores_gen = []
    # Iterar sobre los genes
for i in range(num_genes_importantes):
    # Obtener los shap values de este gen para cada cluster
# Realizar el test de Kruskal-Wallis para este gen en todos los
clusters
    pvalor_gen = kruskal(shap_values_por_cluster[0][:,
i],shap_values_por_cluster[1][:, i],shap_values_por_cluster[2][:,
i]).pvalue

    # Almacenar el p-valor en la lista de resultados
    pvalores_gen.append(pvalor_gen)
# Crear un DataFrame con los p-valores y los nombres de los genes
df_pvalores_gen = pd.DataFrame({
    'Gen': identificadores_genes_seleccionados,
    'P-valor': pvalores_gen
})

df_pvalores_gen_ordenado = df_pvalores_gen.sort_values(by='P-valor',
ascending=True)

# Restablecer los índices del DataFrame ordenado
df_pvalores_gen_ordenado =
df_pvalores_gen_ordenado.reset_index(drop=True)

# Mostrar el DataFrame ordenado
print(df_pvalores_gen_ordenado)
# Especifica el nombre del archivo Excel
nombre_archivo_excel = "pvalores_genes_todos.xlsx"

# Guarda el DataFrame en un archivo Excel
df_pvalores_gen.to_excel(nombre_archivo_excel, index=False)

# Crear un diccionario para almacenar los datos filtrados por cluster
para cada variable clínica
datos_por_variable_por_cluster = {}
```

```
# Iterar sobre las variables clínicas
for nombre_variable, variable_columna in variables_clinicas.items():
    # Crear un diccionario para almacenar los datos filtrados por
cluster para esta variable
    datos_por_cluster = {}

    # Filtrar los datos para cada cluster y almacenarlos en el
diccionario
    for cluster, sujetos in sujetos_por_cluster_c.items():
        datos_por_cluster[cluster] =
datos_sepsis_filtrados[datos_sepsis_filtrados['ID'].isin(sujetos)][var
iable_columna]

    # Almacenar los datos por variable y por cluster en el diccionario
principal
    datos_por_variable_por_cluster[nombre_variable] =
datos_por_cluster



# Crear un diccionario para almacenar los resultados de las pruebas
estadísticas por variable clínica
resultados_estadisticos_por_variable = {}
resultados_comparacion_pares = {}

# Variables para aplicar chi-cuadrado
variables_chi_cuadrado = ['Tipo de Sepsis', 'Sexo', 'Mortalidad en 90
días']

# Iterar sobre las variables clínicas
for nombre_variable, variable_columna in variables_clinicas.items():
    resultados_estadisticos_por_variable[nombre_variable] = {}

    # Obtener los datos por cluster para esta variable
    datos_por_cluster =
datos_por_variable_por_cluster[nombre_variable]

    # Si la variable está en variables_chi_cuadrado, aplicar la prueba
de chi-cuadrado
    if nombre_variable in variables_chi_cuadrado:
        # Calcular los recuentos de datos por valor para cada cluster
        recuentos_por_cluster = [datos.value_counts() for datos in
datos_por_cluster.values()]

        # Crear un DataFrame con los recuentos por valor en cada
cluster
        datos_cluster_concatenados = pd.concat(recuentos_por_cluster,
axis=1, keys=datos_por_cluster.keys())

        # Aplicar la prueba de chi-cuadrado
        _, p_valor_general, _, _ =
chi2_contingency(datos_cluster_concatenados)

        # Almacenar el p-valor en el diccionario de resultados
        resultados_estadisticos_por_variable[nombre_variable] =
{'tipo': 'Chi-cuadrado', 'p-valor': p_valor_general}
```

```
        # Comparaciones por pares para Chi-cuadrado
        resultados_comparacion_pares[nombre_variable] = {}
        num_clusters = len(datos_por_cluster)

        for i in range(num_clusters):
            for j in range(i + 1, num_clusters):
                # Crear una tabla de contingencia para los clusters i
y j
                tabla_contingencia =
pd.concat([recuentos_por_cluster[i], recuentos_por_cluster[j]],
axis=1, keys=[i, j])
                tabla_contingencia = tabla_contingencia.fillna(0)  #
Rellenar valores NaN con 0

                # Aplicar la prueba de Chi-cuadrado
                _, p_valor_pares, _, _ =
chi2_contingency(tabla_contingencia)

                # Almacenar el p-valor para esta comparación de pares
                comparacion = f'Cluster {i} vs Cluster {j}'

resultados_comparacion_pares[nombre_variable][comparacion] =
p_valor_pares

    else:
        # Almacenar los datos en una lista de listas para cada cluster
        datos_por_cluster_list = [datos.tolist() for datos in
datos_por_cluster.values()]

        # Aplicar la prueba de Kruskal-Wallis
        resultado_kruskal = kruskal(*datos_por_cluster_list)

        # Almacenar el resultado en el diccionario de resultados
        resultados_estadisticos_por_variable[nombre_variable] =
{'tipo': 'Kruskal-Wallis', 'estadístico': resultado_kruskal.statistic,
'p-valor': resultado_kruskal.pvalue}

        # Comparaciones por pares para Kruskal-Wallis
        resultados_comparacion_pares[nombre_variable] = {}
        num_clusters = len(datos_por_cluster_list)

        for i in range(num_clusters):
            for j in range(i + 1, num_clusters):
                # Comparar los datos entre los clusters i y j
utilizando Mann-Whitney U test
                p_valor_pares =
mannwhitneyu(datos_por_cluster_list[i], datos_por_cluster_list[j],
alternative='two-sided').pvalue

                # Almacenar el p-valor para esta comparación de pares
                comparacion = f'Cluster {i} vs Cluster {j}'

resultados_comparacion_pares[nombre_variable][comparacion] =
p_valor_pares

# Mostrar los resultados generales
```

```
print("Resultados generales de las pruebas estadísticas:")
for nombre_variable, resultado_estadistico in
resultados_estadisticos_por_variable.items():
    print(f"Variable: {nombre_variable}")
    print(f"Tipo de prueba estadística:
{resultado_estadistico['tipo']}")

    if resultado_estadistico['tipo'] == 'Chi-cuadrado':
        print(f"P-valor Chi-cuadrado general:
{resultado_estadistico['p-valor']}")
    else:
        print(f"Estadístico de prueba de Kruskal-Wallis:
{resultado_estadistico['estadístico']}")
        print(f"P-valor: {resultado_estadistico['p-valor']}")

    print()

# Mostrar los resultados estadísticamente significativos (p-valor <
0.05) para las comparaciones por pares entre clusters
print("Resultados estadísticamente significativos (p-valor < 0.05)
para comparaciones por pares entre clusters:")
for nombre_variable, comparaciones in
resultados_comparacion_pares.items():
    for comparacion, p_valor_pares in comparaciones.items():
        if p_valor_pares < 0.05:
            print(f"Variable: {nombre_variable}")
            print(f"Comparación: {comparacion}")
            print(f"P-valor: {p_valor_pares}")
            print()

# Añadir los p-valores generales al DataFrame df_resto
df_resto['p-valor'] = df_resto.index.map(lambda var:
resultados_estadisticos_por_variable.get(var, {}).get('p-valor',
None))

# Añadir columna 'p-valor' al DataFrame df_importantes (solo para las
variables importantes)
variables_importantes = ['Sexo', 'Tipo de Sepsis', 'Mortalidad en 90
días']
df_importantes['p-valor'] =
[resultados_estadisticos_por_variable.get(var, {}).get('p-valor',
None) for var in variables_importantes]

# Guardar los DataFrames en archivos Excel
archivo_excel_importantes = os.path.join(directorio_resultados,
"resultados_importantes.xlsx")
archivo_excel_resto = os.path.join(directorio_resultados,
"resultados_resto.xlsx")

df_importantes.to_excel(archivo_excel_importantes,
float_format="%.3f")
df_resto.to_excel(archivo_excel_resto, float_format="%.3f")


#DIAS
# Crear un diccionario para almacenar los datos filtrados por cluster
para cada variable clínica
```

```
datos_por_variable_por_cluster_d = {}

# Iterar sobre las variables clínicas
for nombre_variable, variable_columna in variables_mortalidad.items():
    # Crear un diccionario para almacenar los datos filtrados por
cluster para esta variable
    datos_por_cluster_dias = {}

    # Filtrar los datos para cada cluster y almacenarlos en el
diccionario
    for cluster, sujetos in sujetos_por_cluster_c.items():
        datos_por_cluster_dias[cluster] =
datos_sepsis_filtrados[datos_sepsis_filtrados['ID'].isin(sujetos)][var
iable_columna]

    # Almacenar los datos por variable y por cluster en el diccionario
principal
    datos_por_variable_por_cluster_d[nombre_variable] =
datos_por_cluster_dias




# Crear un diccionario para almacenar los resultados de las pruebas
estadísticas por variable clínica
resultados_estadisticos_por_variable_d = {}
resultados_comparacion_pares_d = {}

# Variables para aplicar chi-cuadrado
variables_chi_cuadrado = ['muerte_en_uci']

# Iterar sobre las variables clínicas
for nombre_variable, variable_columna in variables_mortalidad.items():
    resultados_estadisticos_por_variable_d[nombre_variable] = {}

    # Obtener los datos por cluster para esta variable
    datos_por_cluster =
datos_por_variable_por_cluster_d[nombre_variable]

    # Si la variable está en variables_chi_cuadrado, aplicar la prueba
de chi-cuadrado
    if nombre_variable in variables_chi_cuadrado:
        # Calcular los recuentos de datos por valor para cada cluster
        recuentos_por_cluster = [datos.value_counts() for datos in
datos_por_cluster.values()]

        # Crear un DataFrame con los recuentos por valor en cada
cluster
        datos_cluster_concatenados = pd.concat(recuentos_por_cluster,
axis=1, keys=datos_por_cluster.keys())

        # Aplicar la prueba de chi-cuadrado
        _, p_valor_general, _, _ =
chi2_contingency(datos_cluster_concatenados)

        # Almacenar el p-valor en el diccionario de resultados
```

```
        resultados_estadisticos_por_variable_d[nombre_variable] =
{'tipo': 'Chi-cuadrado', 'p-valor': p_valor_general}

        # Comparaciones por pares para Chi-cuadrado
        resultados_comparacion_pares_d[nombre_variable] = {}
        num_clusters = len(datos_por_cluster)

        for i in range(num_clusters):
            for j in range(i + 1, num_clusters):
                # Crear una tabla de contingencia para los clusters i
y j
                tabla_contingencia =
pd.concat([recuentos_por_cluster[i], recuentos_por_cluster[j]],
axis=1, keys=[i, j])
                tabla_contingencia = tabla_contingencia.fillna(0)  #
Rellenar valores NaN con 0

                # Aplicar la prueba de Chi-cuadrado
                _, p_valor_pares, _, _ =
chi2_contingency(tabla_contingencia)

                # Almacenar el p-valor para esta comparación de pares
                comparacion = f'Cluster {i} vs Cluster {j}'

resultados_comparacion_pares_d[nombre_variable][comparacion] =
p_valor_pares

    else:
        # Almacenar los datos en una lista de listas para cada cluster
        datos_por_cluster_list = [datos.tolist() for datos in
datos_por_cluster.values()]

        # Aplicar la prueba de Kruskal-Wallis
        resultado_kruskal = kruskal(*datos_por_cluster_list,
nan_policy='omit')

        # Almacenar el resultado en el diccionario de resultados
        resultados_estadisticos_por_variable_d[nombre_variable] =
{'tipo': 'Kruskal-Wallis', 'estadístico': resultado_kruskal.statistic,
'p-valor': resultado_kruskal.pvalue}

        # Comparaciones por pares para Kruskal-Wallis
        resultados_comparacion_pares_d[nombre_variable] = {}
        num_clusters = len(datos_por_cluster_list)

        for i in range(num_clusters):
            for j in range(i + 1, num_clusters):
                # Comparar los datos entre los clusters i y j
utilizando Mann-Whitney U test
                p_valor_pares =
mannwhitneyu(datos_por_cluster_list[i], datos_por_cluster_list[j],
alternative='two-sided').pvalue

                # Almacenar el p-valor para esta comparación de pares
                comparacion = f'Cluster {i} vs Cluster {j}'
```

```
resultados_comparacion_pares_d[nombre_variable][comparacion] =
p_valor_pares

# Mostrar los resultados generales
print("Resultados generales de las pruebas estadísticas:")
for nombre_variable, resultado_estadistico in
resultados_estadisticos_por_variable_d.items():
    print(f"Variable: {nombre_variable}")
    print(f"Tipo de prueba estadística:
{resultado_estadistico['tipo']}")

    if resultado_estadistico['tipo'] == 'Chi-cuadrado':
        print(f"P-valor Chi-cuadrado general:
{resultado_estadistico['p-valor']}")
    else:
        print(f"Estadístico de prueba de Kruskal-Wallis:
{resultado_estadistico['estadístico']}")
        print(f"P-valor: {resultado_estadistico['p-valor']}")

    print()

# Mostrar los resultados estadísticamente significativos (p-valor <
0.05) para las comparaciones por pares entre clusters
print("Resultados estadísticamente significativos (p-valor < 0.05)
para comparaciones por pares entre clusters:")
for nombre_variable, comparaciones in
resultados_comparacion_pares_d.items():
    for comparacion, p_valor_pares in comparaciones.items():
        if p_valor_pares < 0.05:
            print(f"Variable: {nombre_variable}")
            print(f"Comparación: {comparacion}")
            print(f"P-valor: {p_valor_pares}")
            print()

# Añadir los p-valores generales al DataFrame df_dias
df_dias['p-valor'] = df_dias.index.map(lambda var:
resultados_estadisticos_por_variable_d.get(var, {}).get('p-valor',
None))

# Guardar el DataFrame actualizado en un archivo Excel
archivo_excel_dias = os.path.join(directorio_resultados,
"resultados_dias.xlsx")
df_dias.to_excel(archivo_excel_dias, float_format="%.3f")


# Comorbilidades
comorbilidades = {
    'Chronic renal failure': 'comorb_crf',
    'Chronic cardiovascular disease': 'comorb_card',
    'Obesity': 'comorb_obes',
    'Diabetes mellitus': 'comorb_diab',
    'Chronic hepatic failure': 'comorb_chf',
    'Immunosuppression': 'comorb_inm',
    'High blood pressure': 'comorb_hbp',
    'Chronic respiratory disease': 'comorb_crd'
}
```

```
# Directorio donde se almacenarán los resultados
directorio_resultados_comorbilidades = "resultados_comorbilidades"

# Crear el directorio si no existe
if not os.path.exists(directorio_resultados_comorbilidades):
    os.makedirs(directorio_resultados_comorbilidades)




# Validar todas las comorbilidades en el DataFrame
for variable_comorbilidad in comorbilidades.values():
    datos_sepsis_filtrados[variable_comorbilidad] =
datos_sepsis_filtrados[variable_comorbilidad].apply(validar_comorbilid
ad)

resultados_comorbilidades_importantes = {}

# Valor deseado para todas las comorbilidades
valor_deseado_comorbilidades = 1

# Mostrar los resultados por cluster y guardarlos en un archivo Excel
for cluster, sujetos in clusters.items():
    data_importantes_comorbilidades = {}

    for nombre_comorbilidad, variable_comorbilidad in
comorbilidades.items():
        proporcion_seleccionada = calcular_proporcion(

datos_sepsis_filtrados.loc[datos_sepsis_filtrados['ID'].isin(sujetos),
:],
            variable_comorbilidad,
            valor_deseado_comorbilidades
        )
        data_importantes_comorbilidades[nombre_comorbilidad] =
f"{proporcion_seleccionada}"

    # Almacenar los resultados en los diccionarios por cluster
    resultados_comorbilidades_importantes[cluster] =
data_importantes_comorbilidades

# Crear los DataFrames con los resultados de las comorbilidades
df_comorbilidades_importantes =
pd.DataFrame(resultados_comorbilidades_importantes)

# Guardar los DataFrames en un archivo Excel
archivo_excel_comorbilidades_importantes =
os.path.join(directorio_resultados_comorbilidades,
"resultados_comorbilidades.xlsx")
df_comorbilidades_importantes.to_excel(archivo_excel_comorbilidades_im
portantes, float_format="%.3f")

# Crear un diccionario para almacenar los datos filtrados por cluster
para cada variable clínica
datos_comorb_por_cluster = {}
```

```
# Iterar sobre las variables clínicas
for nombre_variable, variable_columna in comorbilidades.items():
    # Crear un diccionario para almacenar los datos filtrados por
cluster para esta variable
    datos_comorb_cluster = {}

    # Filtrar los datos para cada cluster y almacenarlos en el
diccionario
    for cluster, sujetos in sujetos_por_cluster_c.items():
        datos_comorb_cluster[cluster] =
datos_sepsis_filtrados[datos_sepsis_filtrados['ID'].isin(sujetos)][var
iable_columna]

    # Almacenar los datos por variable y por cluster en el diccionario
principal
    datos_comorb_por_cluster[nombre_variable] = datos_comorb_cluster


# Crear un diccionario para almacenar los resultados de la prueba de
chi-cuadrado por variable clínica
resultados_chi_cuadrado_por_variable = {}
resultados_comparacion_pares_comorb = {}

# Iterar sobre las variables clínicas y los clusters
for nombre_variable, variable_columna in comorbilidades.items():
    resultados_chi_cuadrado_por_variable[nombre_variable] = {}

    # Obtener los datos por cluster para esta variable
    datos_comorb_por_cluster_variable =
datos_comorb_por_cluster[nombre_variable]

    # Calcular los recuentos de datos por valor para cada cluster
    recuentos_por_cluster = [datos.value_counts() for datos in
datos_comorb_por_cluster_variable.values()]

    # Crear un DataFrame con los recuentos por valor en cada cluster
    datos_cluster_concatenados = pd.concat(recuentos_por_cluster,
axis=1, keys=datos_comorb_por_cluster_variable.keys())

    # Aplicar la prueba de chi-cuadrado para todos los clusters
    estadistico_chi_cuadrado, p_valor_general, _, _ =
chi2_contingency(datos_cluster_concatenados)

    # Almacenar el p-valor general en el diccionario de resultados
    resultados_chi_cuadrado_por_variable[nombre_variable] = {'p-valor
general': p_valor_general}

    # Comparaciones por pares entre clusters
    resultados_comparacion_pares_comorb[nombre_variable] = {}
    num_clusters = len(datos_comorb_por_cluster_variable)

    for i in range(num_clusters):
        for j in range(i + 1, num_clusters):
            # Crear una tabla de contingencia para los clusters i y j
            tabla_contingencia = pd.concat([recuentos_por_cluster[i],
recuentos_por_cluster[j]], axis=1, keys=[i, j])
```

```
            tabla_contingencia = tabla_contingencia.fillna(0)  #
Rellenar valores NaN con 0

            # Aplicar la prueba de Chi-cuadrado
            _, p_valor_pares, _, _ =
chi2_contingency(tabla_contingencia)

            # Almacenar el p-valor para esta comparación de pares
            comparacion = f'Cluster {i} vs Cluster {j}'

resultados_comparacion_pares_comorb[nombre_variable][comparacion] =
p_valor_pares

# Mostrar los resultados generales
print("Resultados generales de la prueba de Chi-cuadrado:")
for variable, resultados in
resultados_chi_cuadrado_por_variable.items():
    print(f"Variable: {variable}")
    print(f"P-valor Chi-cuadrado general: {resultados['p-valor
general']}")
    print()

# Mostrar los resultados estadísticamente significativos (p-valor <
0.05) para las comparaciones por pares entre clusters
print("Resultados estadísticamente significativos (p-valor < 0.05)
para comparaciones por pares entre clusters:")
for nombre_variable, comparaciones in
resultados_comparacion_pares_comorb.items():
    for comparacion, p_valor_pares in comparaciones.items():
        if p_valor_pares < 0.05:
            print(f"Variable: {nombre_variable}")
            print(f"Comparación: {comparacion}")
            print(f"P-valor: {p_valor_pares}")
            print()

# Añadir los p-valores generales al DataFrame
df_comorbilidades_importantes
df_comorbilidades_importantes['P-valor'] =
df_comorbilidades_importantes.index.map(lambda var:
resultados_chi_cuadrado_por_variable.get(var, {}).get('p-valor
general', None))

# Guardar el DataFrame actualizado en un archivo Excel
archivo_excel_comorbilidades_importantes =
os.path.join(directorio_resultados_comorbilidades,
"resultados_comorbilidades.xlsx")
df_comorbilidades_importantes.to_excel(archivo_excel_comorbilidades_im
portantes, float_format="%.3f")

# Ordenar el DataFrame por el valor de P-valor de menor a mayor
df_ordenado = df_pvalores_gen.sort_values(by='P-valor')

# Guardar el DataFrame ordenado en un archivo Excel
archivo_excel = "Resultados_Pvalores_todos.xlsx"
df_ordenado.to_excel(archivo_excel, index=False)
```