

# **DIAGNÓSTICO MÉDICO PRECOZ MEDIANTE IA**

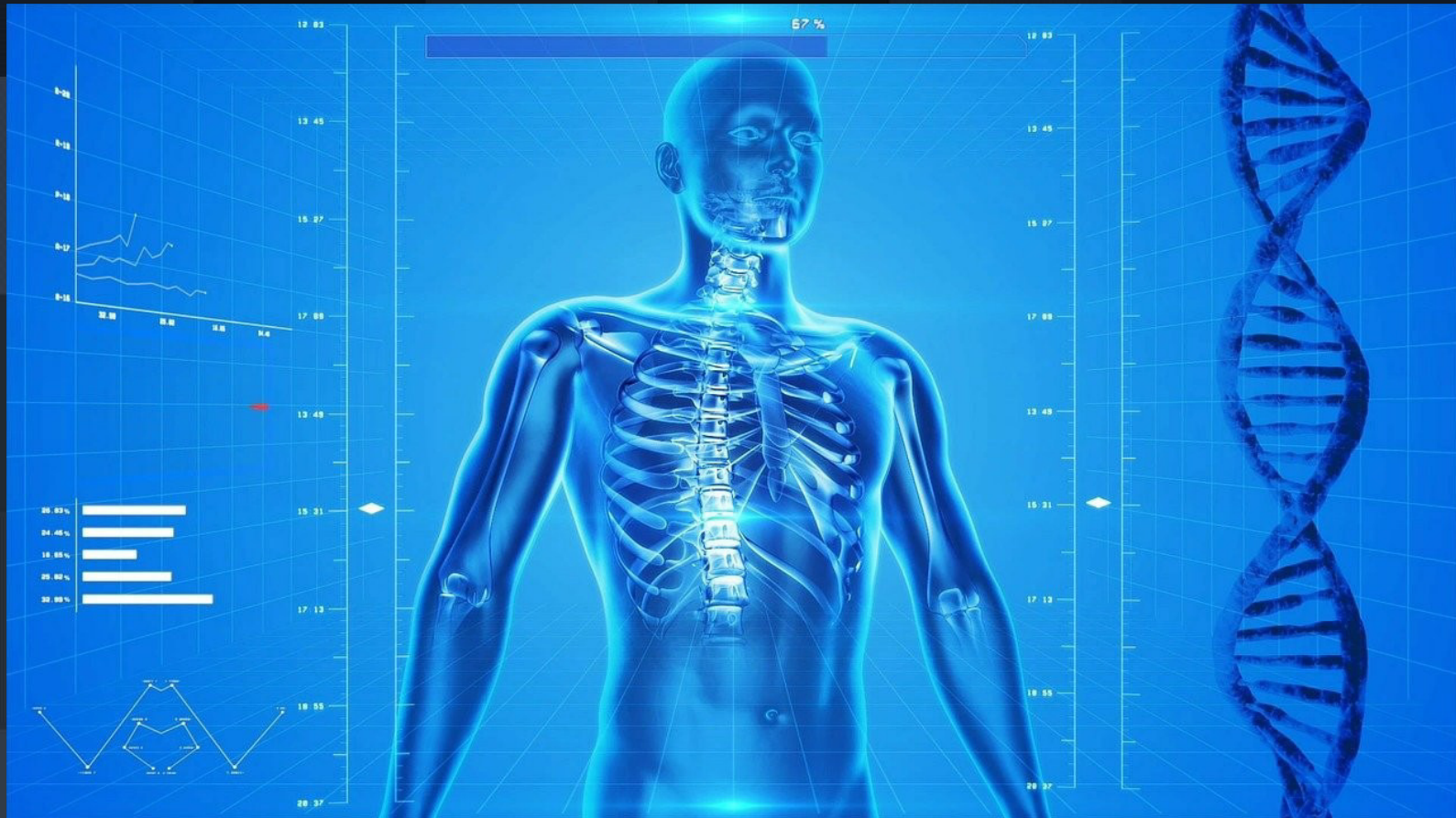
## **ANEXOS DE LOS RECURSOS DE SECUENCIACIÓN**

# **DIAGNÓSTICO MÉDICO PRECOZ MEDIANTE IA**

## **ANEXO 1**

**Presentación descriptiva del problema a resolver**

# Diagnóstico médico precoz mediante IA



Haz click donde veas este símbolo:



# Diagnóstico médico precoz mediante IA



**SALUD Y BIENESTAR:  
POR QUÉ ES  
IMPORTANTE**



**3 SALUD  
Y BIENESTAR**



El gasto de  
1.000 millones

# Diagnóstico médico precoz mediante IA



[partici.fi/90974282](https://partici.fi/90974282)



Entra en esta dirección web para responder de forma interactiva algunas preguntas que nos ayudarán a centrar el tema.

# Diagnóstico médico precoz mediante IA

Organización Mundial de la Salud

Temas de salud ▼ Países ▼ Centro de prensa ▼ Emergencias ▼

## Datos recientes revelan los altos niveles de resistencia a los antibióticos en todo el mundo



29 de enero de 2018 | Comunicado de prensa | BANGKOK

Organización Mundial de la Salud

Temas de salud ▼ Países ▼ Centro de prensa ▼ Emergencias ▼

## Un informe de la OMS confirma que el mundo se está quedando sin antibióticos





# Diagnóstico médico precoz mediante IA



# Diagnóstico médico precoz mediante IA





# Diagnóstico médico precoz mediante IA

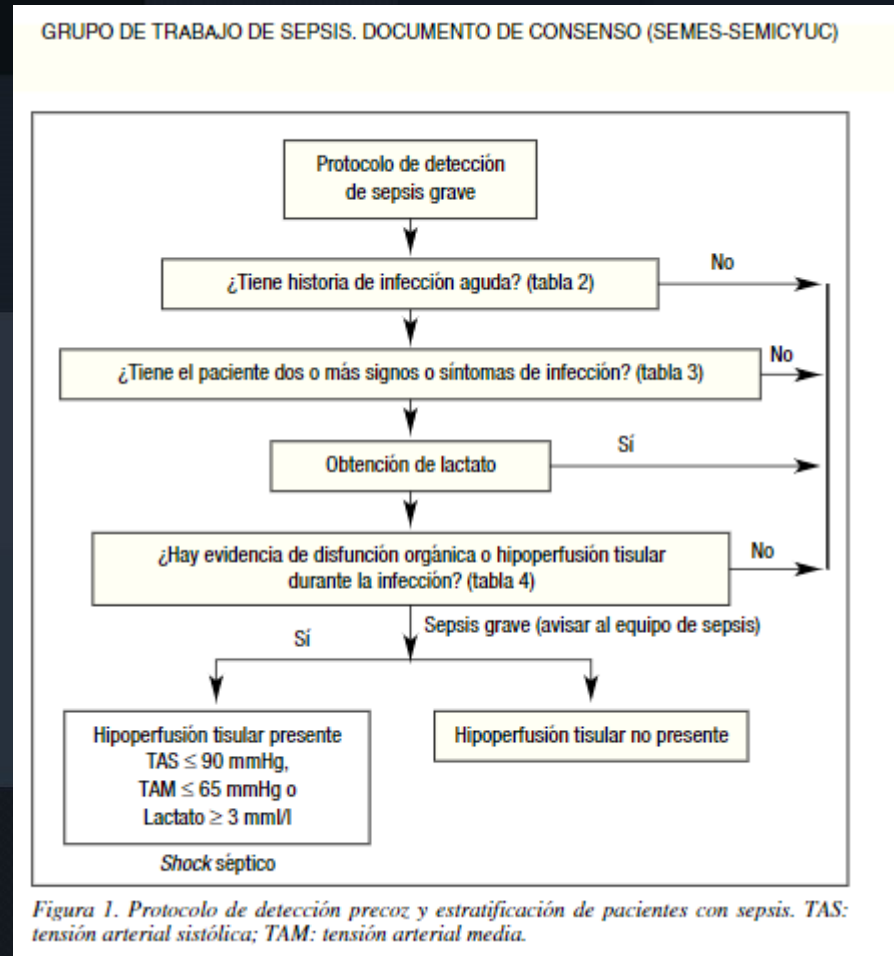
¿Qué ocurre cuando una  
infección no es combatida  
de manera eficaz?



**SEPSIS**

# Diagnóstico médico precoz mediante IA

¿Cuál es el protocolo médico frente a una posible sepsis en Urgencias hospitalarias?



# Diagnóstico médico precoz mediante IA

## ¿Qué podría hacer la IA al respecto?

UCI



Machine Learning Repository

[Center for Machine Learning and Intelligent Systems](#)

### Sepsis survival minimal clinical records Data Set

Download: [Data Folder](#), [Data Set Description](#)

**Abstract:** This dataset collection contains minimal health records of 110,204 admissions (primary cohort), 19,051 admissions

<b>Data Set Characteristics:</b>	Multivariate	<b>Number of Instances:</b>	110341	<b>Area:</b>	Life
<b>Attribute Characteristics:</b>	Integer	<b>Number of Attributes:</b>	4	<b>Date Donated</b>	2020-10-23
<b>Associated Tasks:</b>	Classification	<b>Missing Values?</b>	N/A	<b>Number of Web Hits:</b>	3135



# Diagnóstico médico precoz mediante IA

Diseño de un agente inteligente que realice una predicción de la gravedad de un caso de sepsis en urgencias hospitalarias, para permitir un tratamiento óptimo del paciente.

# **DIAGNÓSTICO MÉDICO PRECOZ MEDIANTE IA**

## **ANEXO 2**

**Presentación de conexión con conocimientos  
previos**

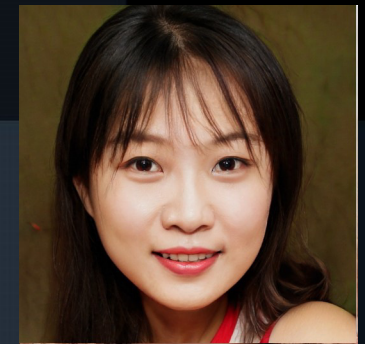
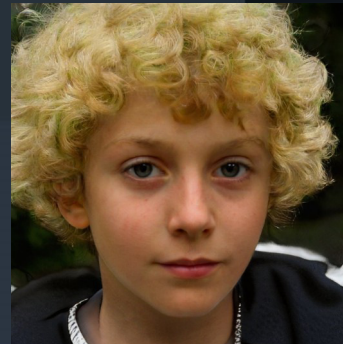
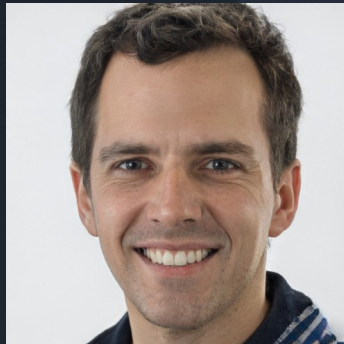


# Resolviendo un problema de clasificación mediante el juego

Haz click donde veas este símbolo:



# Resolviendo un problema de clasificación mediante el juego



Adivina en quié'n estoy pensando...

Las imágenes han sido generadas por una IA en la web <https://thispersondoesnotexist.com/>  
Programado por Phil Wang



# Resolviendo un problema de clasificación mediante el juego

**Datos de entrada:  
Características de  
las muestras y valor  
del objetivo**



**CLASIFICADOR**













**Datos de salida:  
Clase a la que  
pertenece el  
objetivo**

# Resolviendo un problema de clasificación mediante el juego

La construcción del clasificador consiste en encontrar el camino más eficiente (el algoritmo de clasificación más simple, como el mínimo número de preguntas necesarias) que permite asociar una determinada muestra a su clase correspondiente.

# Resolviendo un problema de clasificación mediante el juego

Ejemplo: distinguir tantas clases como muestras

N.º muestra	Género	Color de pelo	Gafas	Sombrero	Longitud del pelo	¿Pelo rizado?	Muestra dientes	¿Conoce la IA?	Imagen
1	Masculino	Castaño	Sí	No	Corto	No	No	No	
2	Femenino	Moreno	No	No	Corto	No	Sí	Sí	
3	Masculino	Rubio	No	No	Corto	No	No	No	
4	Femenino	Castaño	No	No	Largo	No	No	Sí	
5	Masculino	Moreno	No	Sí	Corto	No	No	No	
6	Femenino	Moreno	No	Sí	Corto	Sí	Sí	Sí	
7	Masculino	Moreno	No	No	Corto	No	Sí	Sí	
8	Femenino	Rubio	Sí	No	Corto	No	Sí	Sí	
9	Masculino	Rubio	No	No	Corto	Sí	No	Sí	
10	Femenino	Moreno	No	No	Largo	No	Sí	No	

El clasificador en este caso se limita a describir el detalle de todas y cada una de las muestras, pues se ofrecen la totalidad de características que determina unívocamente cada muestra. De este modo puede servir para establecer un juego de dos jugadores en el que uno elige una muestra y el otro debe adivinar cuál es a través de la formulación de preguntas sobre sus características.













# Resolviendo un problema de clasificación mediante el juego

Ejemplo: distinguir tantas clases como muestras



# Resolviendo un problema de clasificación mediante el juego

Ejemplo: distinguir menos clases que muestras

N.º muestra	Género	Color de pelo	Gafas	Sombrero	Longitud del pelo	¿Pelo rizado?	Muestra dientes	¿Conoce la IA?	Imagen
1	Masculino	Castaño	Sí	No	Corto	No	No	No	
2	Femenino	Moreno	No	No	Corto	No	Sí	Sí	
3	Masculino	Rubio	No	No	Corto	No	No	No	
4	Femenino	Castaño	No	No	Largo	No	No	Sí	
5	Masculino	Moreno	No	Sí	Corto	No	No	No	
6	Femenino	Moreno	No	Sí	Corto	Sí	Sí	Sí	
7	Masculino	Moreno	No	No	Corto	No	Sí	Sí	
8	Femenino	Rubio	Sí	No	Corto	No	Sí	Sí	
9	Masculino	Rubio	No	No	Corto	Sí	No	Sí	
10	Femenino	Moreno	No	No	Largo	No	Sí	No	

Dada una característica que es independiente de todas las demás (como la penúltima columna que dice si conoce la IA) que utilizamos como objetivo de la clasificación, ¿cómo construir el clasificador? Es decir, ¿cuál es el mínimo número de preguntas que nos permite conocer a qué clase corresponde una determinada muestra (Sí conoce la IA o NO conoce la IA)? ¿La clasificación sugerida es cierta o probabilística?

# Resolviendo un problema de clasificación mediante el juego

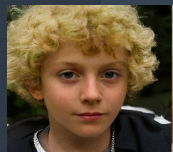
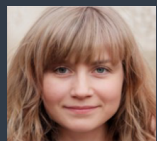
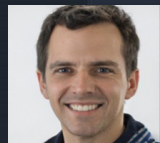
Ejemplo: distinguir menos clases que muestras



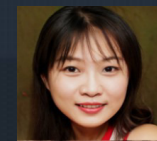
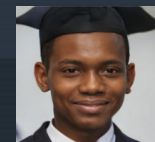
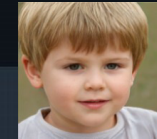
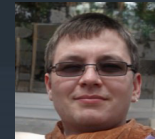
# Resolviendo un problema de clasificación mediante el juego

Resultado: el anterior árbol de clasificación consigue clasificar el conjunto de qué personas NO conocen la IA (por omisión se consigue el grupo complementario). Ese método no es 100% preciso pues se está dejando una muestra fuera, pero, en su conjunto, clasifica bien 9 de las 10 muestras.

Sí conocen la IA



NO conocen la IA



Mal clasificada

# **DIAGNÓSTICO MÉDICO PRECOZ MEDIANTE IA**

## **ANEXO 3**

**Actividad conectar, ampliar, desafiar**



# Conectar, ampliar, desafiar

Tras la experiencia de juego que acaba de suceder en clase, resultaría interesante que reflexionáramos sobre algunos aspectos. Intenta expresar con claridad la respuesta que ofrezcas a cada pregunta. Puedes utilizar ejemplos y experiencias personales si eso te ayuda a expresar la idea que quieres transmitir.

1. ¿Cómo se **conectan** los árboles de decisión que acaban de presentarse con algo que ya conoces?

2. ¿Qué ideas o impresiones tienes que **amplían** tu conocimiento previo en nuevas direcciones? ¿Crees que tiene alguna utilidad?

3. ¿Qué te ha resultado o te resulta **desafiante** o confuso? ¿Hay aspectos que todavía no entiendas? Describe tus dudas con pocas palabras o en frases o preguntas simples.

# **DIAGNÓSTICO MÉDICO PRECOZ MEDIANTE IA**

## **ANEXO 4**

**Instrucciones del juego de clasificación**

## DINÁMICA DE JUEGO DE CLASIFICACIÓN

A continuación vais a intentar resolver un problema de clasificación real: crear un árbol de decisión que permita predecir si una persona que entra en urgencias hospitalarias con un cuadro de infección generalizada (sepsis) tiene probabilidades de sobrevivir o no. No está permitido que utilicéis ningún dispositivo digital para resolver este problema. Tanto el problema como los datos que se os suministran son reales.

Para saber cómo abordar el trabajo que se os propone, aquí se os apuntan algunas indicaciones:

- el [ la docente forma grupos de cuatro personas.
- a cada grupo se le entrega, en una bolsa de plástico cerrada, un conjunto de fichas. Cada una representa los datos de una muestra real, del caso de una persona que acudió a urgencias con un caso de infección generalizada, con un conjunto de características que identifican la muestra (edad, género y número de episodio), junto con el resultado real sobre si sobrevivió o no sobrevivió.
- El total de fichas a repartir entre todos los grupos es 500 y ninguna se repite. Cada grupo recibe un número aleatorio y distinto de fichas (entre 5 y 20). Estas condiciones provocan que el problema que resuelve cada grupo sea distinto y, por tanto, cada grupo ofrecerá una solución diferente.
- Es importante que el alumnado se distribuya tres roles distintos: la secretaría, que toma todas las notas necesarias, la portavoz, que hablará en representación de todo su grupo cuando sea necesario, y los/las investigadores/as, quienes debatirán sobre la manera de resolver el problema. Todo el alumnado de cada grupo asume el rol de investigador, por lo que la portavoz y la secretaría serán cargos que se sumen al de investigador/a.
- Se entrega a cada grupo un pequeño dossier con orientaciones que puedan servir de guía para la realización del trabajo. Este dossier también contiene un conjunto de preguntas que deben ser respondidas por el grupo.
- Al final del trabajo en equipo, se expondrán públicamente las soluciones y los problemas encontrados, a través de los/as portavoces de cada grupo.
- El dossier será entregado al profesor/a, debidamente identificado con los datos del grupo y rellenado con las respuestas a todas y cada una de las preguntas planteadas.

# **DIAGNÓSTICO MÉDICO PRECOZ MEDIANTE IA**

## **ANEXO 5**

**Fichas del juego de clasificación  
(se muestra solo una de las quinientas  
disponibles, a título de ejemplo)**

EDAD: 21

GÉNERO: Femenino

EPISODIO N.º: 1

¿SOBREVIVE?



Si



# **DIAGNÓSTICO MÉDICO PRECOZ MEDIANTE IA**

## **ANEXO 6**

**Cuestionario abierto proceso 2**

# Pensar, trabajar en pareja, compartir

Por favor, lee atentamente este dossier antes de rellenar cualquier campo o de responder a cualquier pregunta. ¡Ánimo con el trabajo!

**Identificación de los miembros de la pareja:** (nombre y apellidos, uno por línea)

- 
- 

Leed las siguientes orientaciones e id respondiendo a las correspondientes preguntas. El objetivo principal de esta dinámica es clasificar las muestras en dos clases distintas: S sobrevive y NO sobrevive. Podéis apoyaros en folios aparte para hacer apuntes antes de responder a las preguntas en este dossier.

En la bolsa de plástico se os entregan unas cuantas fichas, cada una una muestra independiente. Separadlas en dos grupos: uno que utilizaréis para construir el método de clasificación (que podéis llamar 'entrenamiento') y otra que utilizaréis para medir cómo de bueno es vuestro método de clasificación (que podéis llamar 'test'). El número de muestras para entrenamiento debe ser, al menos, triple que la cantidad empleada para test

P1.- ¿Qué característica de las muestras creéis que aporta más información para poder predecir si sobrevive o no cada caso?

Orientaciones:

- ¿Cómo se os ocurre medir la medida de cuánto se parecen dos variables? Quizás dibujar cómo varía cada variable y ponerlas juntas las gráficas os ayude a compararlas. ¿Se os ocurre alguna manera cuantitativa para poder calcularla? ¿Y cualitativa? Ofreced un análisis de lo que se parece cada característica (edad, género y N.º de episodios) a la variable objetivo, la supervivencia. Tras ese análisis, responded a la pregunta P1, incluyendo un razonamiento sobre por qué creéis que tal o cual variable tiene mayor predictivo que las demás.

P2.- En base al análisis anterior, redacta varias frases (más de una) que resuman las condiciones que deben cumplirse para que la persona sobreviva o no sobreviva, en valor medio, de forma aproximada. Deben ser del formato siguiente:

' Si la variable [nombre\_de\_variable] es [condición que cumple con un valor umbral] entonces S /NO sobrevive'

P3.- Tomando como referencia las frases anteriores, formula preguntas que tenga sentido hacerse para pretender clasificar (decir si sobrevive o no) una muestra cualquiera.

P4.- Utilizando una a una las preguntas anteriores sobre las muestras que utilizaremos para testear el método de clasificación, ¿como de correctas son las predicciones que se realizan respondiendo a ellas? ¿Úué relación de aciertos y fallos se consigue con cada secuencia de preguntas?

Ahora que haà.ã concluido el trabajo en parejasÈllega la hora de compartir vuestros pensamientos. Recortad cada rectángulo y poner cada uno (cada pregunta) en el sitio correspondiente del tablón de la clase.

En el espacio rectangular azul que tenéis a continuación, apuntad la información que creáis interesante de lo que han colgado el resto de grupos en el tablón. ¡Ánimo!

# **DIAGNÓSTICO MÉDICO PRECOZ MEDIANTE IA**

## **ANEXO 7**

**Guía didáctica proceso 3**

## Guía didáctica – Construcción de un clasificador mediante IA

Esta guía didáctica pretende ayudarte a seguir las explicaciones que se dan en el aula y servir como referencia sobre el uso adecuado de las herramientas digitales que se hace necesario emplear. Acuda primero a ella siempre que tengas alguna duda.

1.- Entra en la web <https://machinelearningforkids.co.uk/?lang=es> y haz click en Empezar.

**Enséñale a una computadora a jugar un juego**

1. Agrega ejemplos de cosas que quieres que la computadora reconozca/identifique automáticamente

2. Usa los ejemplos para entrenar una computadora y así aprende a reconocer este tipo de objetos (texto/imágenes/números) automáticamente

3. Crea un juego en Scratch que use la habilidad de la computadora para reconocerlos

Empezar Aprender más

2.- Haz click en el botón 'Pruébalo ahora', de manera que no sea necesario registrarse. Recuerda que, de este modo, lo que sea que hagas se perderá cuando apagues el equipo, así que asegúrate de extraer toda la información que se te solicite a través de capturas de pantalla o de cualquier otro medio.

Empezar con el aprendizaje automático.

¿Tu primera vez aquí? **Regístrate** [¿Por qué registrarse?](#)

¿Ya registrado? **Inicia sesión** [¿Olvidaste tus datos?](#)

Prueba sin registrarte **Pruébalo ahora**

3.- Haz click en el botón 'Añadir un nuevo proyecto'.

## Tus proyectos de aprendizaje automático

Haz clic en el botón 'más' a la izquierda para crear tu primer proyecto. →



+ Añadir un nuevo proyecto

Copiar plantilla

4.- En la página que se te abrirá entonces debes hacer tres cosas:

- Primero, darle un nombre al proyecto. Intenta que resuma el objetivo del problema de clasificación
- Segundo, debes indicarle al sistema si la IA que vas a crear trabajará con texto, números, imágenes o sonidos. Nosotros deberemos elegir 'números'.
- Tercero, deberás hacer click en el botón 'Add a value' ('Añadir un valor') para indicarle que variables van a utilizarse para identificar cada muestra. En nuestro caso habrá que introducir tres variables: 'Edad', en formato numérico, 'Género', en formato de respuesta múltiple, y 'Episodio N.º', en formato numérico también.
- Por último, haz click en el botón 'Crear'. Te debe quedar algo como lo que se muestra en la siguiente figura.

Value 1 * Edad	Tipo de valor * número	✕
Value 2 * Género	Tipo de valor * opción múltiple	✕
Opciones: Masculino <input checked="" type="radio"/> Femenino <input checked="" type="radio"/> Agregar una opción		
Value 3 * Episodio N°	Tipo de valor * número	✕
If Episodio N° can be described as numbers, choose "number". Si se puede describir cómo elegir entre algunas opciones, elige "opción múltiple".		

ADD ANOTHER VALUE



CREAR

CANCELAR

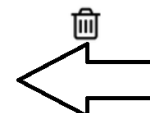
5.- ¡Felicidades! ¡Ya tienes tu primer proyecto creado! Ahora debes hacer click en el botón que muestra el nombre de tu proyecto.

+ Añadir un nuevo proyecto

Copiar plantilla

### Predicción de supervivencia

Reconociendo **números**



6.- En la ventana que te aparecerá entonces se te darán varias opciones posibles. En este momento debes hacer click en la primera, pinchando en botón 'Entrenar'.



## "Predicción de supervivencia"

### Entrenar

Introduce ejemplos de lo que quieres que la computadora identifique

Entrenar



### Aprender & Probar

Usa los ejemplos para entrenar la computadora a reconocer numbers

Aprender & Probar

### Crea

Usa el modelo de aprendizaje automático que has entrenado para crear un juego o una aplicación, en Scratch o en Python

Crea

7.- Aquí es donde vas a definir en cuantas clases se define tu objetivo. En nuestro caso, nuestro objetivo es predecir la variable que nos dice si el/la paciente sobrevive o no, por lo que hemos de definir dos clases: 'Sobrevive' y 'No sobrevive'. Para añadir cada clase deberás hacer click en el botón 'Añadir etiqueta'.

## Reconociendo números

[Volver al proyecto](#)

Haz click sobre el botón 'plus' a la derecha para añadir tu primer contenedor →



+ Añadir etiqueta

Cuando hagas click en ese botón te pedirá el nombre de la clase. Aquí hemos utilizado los nombres 'Sobrevive' y 'No sobrevive' (el espacio lo codifica automáticamente como un guión bajo), obteniendo lo que se ve en la siguiente figura:

## Reconociendo números como Sobrevive or No\_sobrevive

[Volver al proyecto](#)

+ Añadir etiqueta

Sobrevive

No\_sobrevive

+ Añade un ejemplo.

+ Añade un ejemplo.

8.- Ahora, debemos introducir todas y cada una de las muestras de las que disponemos información de manera ordenada: aquellas en las que la variable objetivo tiene por valor que 'Sí sobrevive' las incluiremos en la clase de la izquierda y las que tenga por valor objetivo 'No sobrevive' se introducirán en la clase de la derecha.

Aquí tienes en formato de tabla el conjunto de muestras que se te da para entrenar a la IA, que son los datos que vas a introducir en este paso. Recuerda que para cada muestra se te está pidiendo que introduzcas los valores de las tres primeras columnas, mientras

que el valor de la última columna te ayuda a discriminar a qué clase pertenece cada muestra.

Edad	Género	N.º Episodio	Supervivencia
21	Femenino	1	Sí
72	Masculino	1	No
45	Masculino	1	Sí
63	Masculino	1	No
40	Masculino	1	Sí
89	Femenino	2	No
70	Masculino	1	Sí
80	Masculino	3	No
62	Femenino	1	Sí
62	Femenino	3	No

Puedes introducir cada muestra en la clase correspondiente haciendo click en el botón ‘Añade un ejemplo’. Tras introducir todos los valores, debe quedarte una pantalla como la que se muestra en la siguiente figura.

Reconociendo **números** como **Sobrevive or No\_sobrevive**

[: Volver al proyecto](#)

+ Añadir etiqueta

**Sobrevive**

Edad 21  
Género Femenino  
Episodio N° 1

Edad 45  
Género Masculino  
Episodio N° 1

Edad 40  
Género Masculino  
Episodio N° 1

Edad 70  
Género Masculino  
Episodio N° 1

Edad 62  
Género Femenino  
Episodio N° 1

+ Añade un ejemplo.

5

**No\_sobrevive**

Edad 72  
Género Masculino  
Episodio N° 1

Edad 63  
Género Masculino  
Episodio N° 1

Edad 89  
Género Femenino  
Episodio N° 2

Edad 80  
Género Masculino  
Episodio N° 3

Edad 62  
Género Femenino  
Episodio N° 3

+ Añade un ejemplo.

5

9.- A continuación, haz click en el texto que aparece resaltado en azul en la figura anterior que dice ‘Volver al proyecto’. Te aparecerá una pantalla como la siguiente, en la que deberás hacer click en el botón ‘Aprender & Probar’.

"Predicción de supervivencia"

**Entrenar**

Introduce ejemplos de lo que quieres que la computadora identifique

Entrenar

**Aprender & Probar**

Usa los ejemplos para entrenar la computadora a reconocer numbers

Aprender & Probar

**Crea**

Usa el modelo de aprendizaje automático que has entrenado para crear un juego o una aplicación, en Scratch o en Python

Crea

10.- Una vez allí, se te mostrará una pantalla como la que aparece en la siguiente figura, donde deberás hacer click sobre el botón ‘Entrena un nuevo modelo’.

Modelos de aprendizaje automático

al proyecto

### ¿Qué has hecho hasta ahora?

Has añadido ejemplos de números para que la computadora aprenda a reconocer cuando un número es Sobrevive or No\_sobrevive.

Has agregado:

- 5 examples of Sobrevive,
- 5 examples of No\_sobrevive

### ¿Y ahora qué hay que hacer?

¿Lista para empezar con el entrenamiento ?

Haz click en el botón de abajo para empezar a entrenar el modelo de aprendizaje automático con los ejemplos que has añadido hasta ahora.

(O vuelve a la página [Entrenar](#) si quieres añadir antes más ejemplos.)

Información del entrenamiento:

Entrena un nuevo modelo



Una vez hayas hecho click, acabarás de haber entrenado tu primer modelo de clasificación basado en IA. ¡Enhorabuena! Te aparecerá en la pantalla un mensaje del siguiente estilo:

Añade un número para ver cómo lo identifica tras el entrenamiento.

Edad	<input type="text"/>
Género	<input type="text" value="Femenino"/>
Episodio Nº	<input type="text"/>

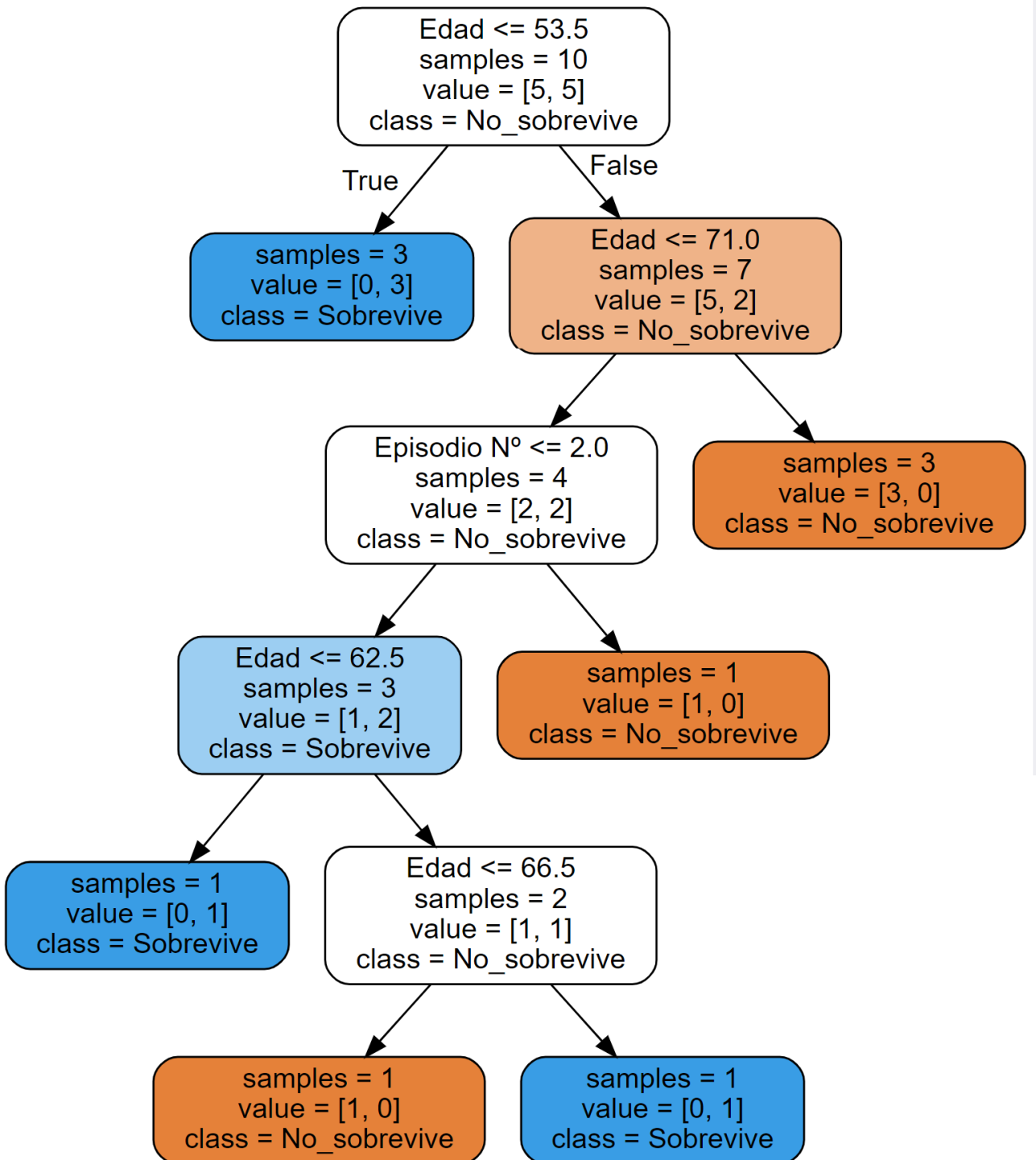
Información del entrenamiento:

**Modelo se inició started training at:** Friday, May 13, 2022 4:44 PM  
**Estatus actual:** Available

Borra este modelo

Entrena un nuevo modelo

En la caja superior puedes introducir los valores que se te ocurran para probar qué predicción arroja tu modelo. Es muy interesante que hagas click en el botón blanco que aparece al lado del botón ‘Probar’, y que pone ‘Describe your model!’. Al hacerlo, se te mostrará, entre otras cosas, el árbol de decisión específico al que ha conducido el entrenamiento de este modelo de IA. Lo ponemos a continuación, para que compruebes qué deberías ver.



# **DIAGNÓSTICO MÉDICO PRECOZ MEDIANTE IA**

## **ANEXO 8**

**Cuestionario abierto proceso 3**

# Hipotetiza, experimenta, analiza y concluye

Ahora que ya tienes operativo tu primer modelo, será interesante que pruebes algunas cosas. Aquí te proponemos algunas:

- aumentar o disminuir el número de muestras utilizadas para el entrenamiento de la IA.
- Eliminar una o más características de todas las muestras.
- Añadir una característica de valor aleatorio a todas las muestras.
- Eliminar una de las dos clases, de manera que el sistema solo aprenda a reconocer una de ellas (la otra se percibirá por omisión).

De la actividad anterior, tenéis cada grupo un conjunto de fichas, representando cada una un caso real con los datos que caracterizan cada muestra y el resultado de si sobrevive o no sobrevive cada uno de los pacientes. Utiliza esta información para el desarrollo de tus experimentos.

**Hipotetiza:** Los experimentos que puedes plantear deben partir de una hipótesis, una afirmación que refleje un hecho que es susceptible de poder probarse o refutarse. Es habitual en este contexto que se formulen hipótesis acerca del efecto sobre la eficacia del agente inteligente a la hora de predecir cada una de las clases (sobrevive / no sobrevive).

**Experimenta:** Explica qué experimentos vas a desarrollar para probar tu hipótesis (qué cosas vas a probar, con qué modelo, etc.).

**Analiza:** Presenta los resultados de los experimentos y los cálculos o análisis realizados sobre aquellos datos que te permiten luego concluir una u otra cosa.

**Concluye:** Resume las conclusiones de tu análisis, corroborando o refutando tu hipótesis.

Finalmente, se os presentan algunas preguntas (indicadas con P1, P2, etc.) que deberéis responder:

P1. ¿Cómo estáis midiendo lo bueno que es vuestro modelo? ¿Podemos resumir esa información en un parámetro? ¿Cuál? Explica cómo puede calcularse ese parámetro y por qué es una buena medida de lo bueno que es el modelo.

P2. ¿De qué maneras puede vuestro modelo fallar en su predicción? ¿Solo falla al predecir la clase 'Sobrevive'? ¿O lo hace únicamente al predecir la clase 'No sobrevive'? ¿Falla en ambas? Si falla en ambas, ¿donde falla más? ¿cómo cuantificaríais este comportamiento? ¿Creéis que es igual de grave fallar al intentar predecir que sobrevive o fallar al intentar predecir que no sobrevive? ¿Por qué?

# **DIAGNÓSTICO MÉDICO PRECOZ MEDIANTE IA**

## **ANEXO 9**

**Cuestionario abierto proceso 4**



# Desarrollo de modelos de IA para predicción de supervivencia por sepsis

A continuación vamos a desarrollar, más en profundidad, modelos de IA que utilicen conjuntos muy grandes de datos reales que permitan a nuestro modelo aprender de manera más eficiente.

Este Jupyter notebook te va a guiar por el proceso de construcción de un modelo de IA. Posteriormente deberás emplear el modelo para analizar sus resultados, su comportamiento y hacer propuestas de mejora sobre los distintos hiperparámetros que lo definen.

¡Ánimo y adelante!

---

**Nota:** Los datos empleados en este notebook pueden descargarse libremente de la web <https://archive.ics.uci.edu/ml/datasets/Sepsis+survival+minimal+clinical+records>, gracias al repositorio sobre aprendizaje automático del Center for Machine Learning and Intelligent Systems [1]

[1] Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [<http://archive.ics.uci.edu/ml>]. Irvine, CA: University of California, School of Information and Computer Science

---

1. Lo primero que vamos a hacer es importar los paquetes necesarios para hacer todas las operaciones que tenemos pensado hacer. Para ello será recomendable que revises la sintaxis de la declaración `import` en el manual de ayuda de Python, especialmente [los ejemplos del apartado 7.11](#). Siguiendo aquella sintaxis deberás importar los siguientes paquetes: `pandas`, `numpy`, `sklearn`, `matplotlib.pyplot` y `graphviz`. Tienes linkados en los nombres de las librerías anteriores guías para el uso elemental de las distintas funciones de cada paquete. Aunque no es algo en lo que necesitemos entrar ahora en detalle, sí es interesante que acudas de manera habitual a estas referencias para entender mejor la sintaxis de cada función y poder ver ejemplos y sus resultados. Resultan fuentes de información muy didácticas que te acompañarán en el proceso de aprendizaje de las distintas funciones y su uso.

In [1]:

```
# TAREA 1
# ----- Aquí debe empezar tu código. Extensión aproximada: cinco líneas de código
import numpy as np
import pandas as pd
from sklearn import tree
import graphviz
import matplotlib.pyplot as plt
# ----- Aquí debe terminar tu código
```

1. El segundo paso será cargar los datos. Tienes el dataset completo en [formato csv](#) (Comma Separated Value) en el propio repositorio que estás ejecutando, por lo que solo hace falta introducir el comando de carga. Lo haremos a través de la librería `pandas` para cargar el



dataset en un DataFrame, en particular, a través de la función `read_csv` de `pandas`. Puedes consultar su sintaxis [aquí](#), donde veréis todo tipo de parámetros para la función, si bien os aconsejo acudir directamente al ejemplo del final, donde solo se hace necesario que le paséis a la función el nombre del archivo csv que queréis cargar.

**Nota:** Para conocer los distintos atributos y métodos aplicables a un dataframe, os aconsejo que os dirijáis a [lo que pone al respecto la documentación de la librería pandas](#).

In [2]:

```
# TAREA 2
# ----- Aquí debe empezar tu código. Extensión aproximada: una línea de código
data_train = pd.read_csv('s41598-020-73558-3_sepsis_survival_primary_cohort.csv')
# ----- Aquí debe terminar tu código
```

1. Una vez cargados los datos, conviene realizar un **análisis exploratorio** de los mismos. En la celda que tienes a continuación, aplica el método `info()` al dataframe donde hemos guardado los datos para averiguar qué campos tiene y qué significan, así como para saber cuántos elementos tiene y en qué formato están guardados. A continuación, usa el método `describe()` al mismo dataframe para entender mejor cómo se comportan cada una de las variables contenidas, qué rango de variación tienen, así como valores característicos de su posición y dispersión.

In [3]:

```
# TAREA 3
# Escribe el comando para representar información acerca del contenido del dataframe
# utilizando los métodos info() y describe()

# ----- Aquí debe empezar tu código. Extensión aproximada: dos líneas de código
data_train.info()
data_train.describe()
# ----- Aquí debe terminar tu código
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 110204 entries, 0 to 110203
Data columns (total 4 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   age_years                             110204 non-null int64
1   sex_0male_1female                    110204 non-null int64
2   episode_number                       110204 non-null int64
3   hospital_outcome_1alive_0dead       110204 non-null int64
dtypes: int64(4)
memory usage: 3.4 MB
```

Out[3]:

	age_years	sex_0male_1female	episode_number	hospital_outcome_1alive_0dead
<b>count</b>	110204.000000	110204.000000	110204.000000	110204.000000
<b>mean</b>	62.735255	0.473948	1.349379	0.926455
<b>std</b>	24.126806	0.499323	0.751799	0.261031
<b>min</b>	0.000000	0.000000	1.000000	0.000000
<b>25%</b>	51.000000	0.000000	1.000000	1.000000
<b>50%</b>	68.000000	0.000000	1.000000	1.000000
<b>75%</b>	81.000000	1.000000	1.000000	1.000000

	age_years	sex_0male_1female	episode_number	hospital_outcome_1alive_0dead
max	100.000000	1.000000	5.000000	1.000000

Sería también interesante que aplicaras el método `unique()` a cada columna del dataframe, para obtener una lista de todos los valores distintos que puede tomar cada variable.

In [4]:

```
# TAREA 4
# ----- Aquí debe empezar tu código. Extensión aproximada: entre una y cuatro líneas
[print(data_train[str(a)].unique()) for a in data_train.columns]
# ----- Aquí debe terminar tu código
```

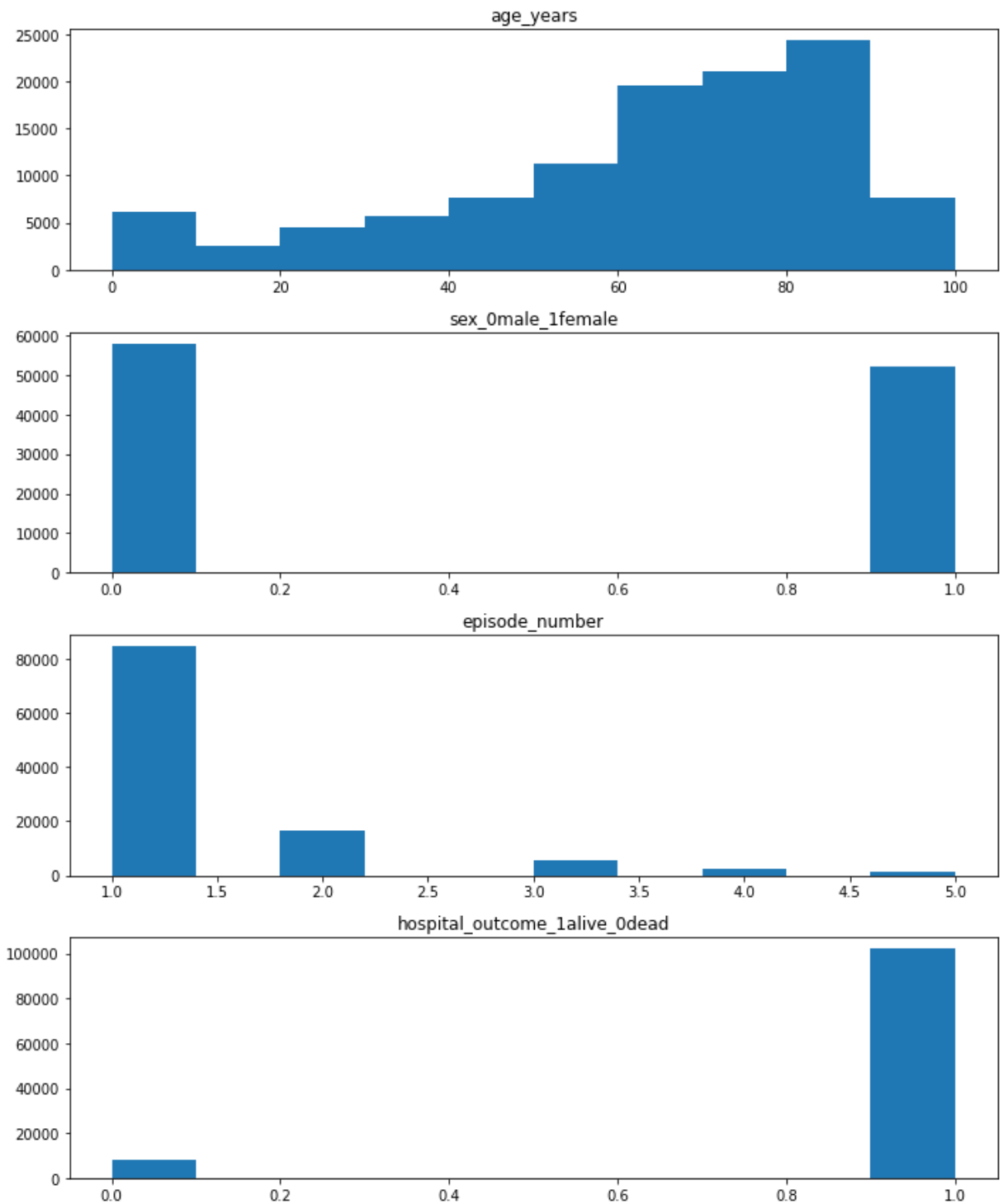
```
[ 21  20  77  72  83  74  69  53  82  75  45  56  46  48  40  39  70  47
 27  11  91   7  79  84  16  73  17  18  63  88  89  76  41  66  80  62
 59  55  68  33  71   8  58  78  51  43  44  60  86  61  67  57  81  49
 64  25  65  42  36  38  85  24  19  37  35   6  50  87  54  29  12  10
 23  52   9  15  31  92  28  30  13  94  90  26  32  95   5  93  34  96
 22  97  98 100  14   4  99   3   2   1   0]
[1 0]
[1 2 3 4 5]
[1 0]
```

Out[4]: [None, None, None, None]

Para tener ya una visión completa del conjunto de datos que se nos ofrece, sería bueno dibujar un histograma de cada uno de los campos (puedes utilizar `plt.hist()` y observar cómo está empleado en un ejemplo [aquí](#)). Así, entenderíamos de un golpe de vista los descriptores estadísticos facilitados anteriormente con el método `describe()`. Si quieres cuidar un poco más la presentación, podrías utilizar la función `plt.subplots()` para dibujar cada plot en una malla de figuras, como se ve en [algunos de estos ejemplos](#).

In [5]:

```
# TAREA 5
# ----- Aquí debe empezar tu código. Extensión aproximada: entre una y 12 líneas de
# en función de los elementos que defines de cada gráfico.
[a, b] = [4, 1]
fig, axs = plt.subplots(a, b, figsize=(10, 12))
for i in range(a):
    for j in range(b):
        axs[i+j].hist(data_train[data_train.columns[i+j]])
        axs[i+j].set_title(str(data_train.columns[i+j]))
plt.tight_layout()
# ----- Aquí debe terminar tu código
```



## Reflexionemos sobre los datos

**TAREA 6** A la luz de las gráficas que acabas de obtener, responde a las siguientes preguntas:

- ¿Qué conclusiones extraerías del histograma de la variable 'edad del paciente'? ¿Qué rango de edades son los más susceptibles de desarrollar un episodio de sepsis? ¿Por qué dirías que es así?
- ¿Existe algún desbalance en la variable 'género'? ¿Lo esperarías? ¿Por qué?
- ¿Qué ley parece seguir el número de infecciones cuando crece el número de episodios sufridos? ¿Por qué crees que es así?
- Como ya sabes por actividades anteriores, la última variable es la variable objetivo, la que determina si un paciente sobrevive o no sobrevive. Observando la distribución de valores

de esa variable, ¿qué aspecto te resulta más significativo resaltar? ¿es la distribución de datos simétrica para esa variable? ¿qué quiere decir eso? ¿cómo crees que puede influir eso para la resolución de un problema de clasificación en el que la IA aprende de estos datos? ¿puede aprender a distinguir las dos clases con la misma precisión? ¿por qué? ¿qué se te ocurre que podría hacerse para solucionar este problema?

## Nuestro primer modelo de clasificación mediante IA

Una vez hemos entendido el problema que se nos presenta y hemos analizado la naturaleza y características esenciales de los datos suministrados, procedemos a la construcción del árbol de decisión que utilizaremos para distinguir las dos clases que tenemos definidas ('Sobrevive' y 'No sobrevive'). Para ello vamos a hacer uso de funciones del paquete scikit-learn.

No es necesario que sepas reproducir todo el código que hace falta, pues queremos centrarnos específicamente en las funciones que permiten crear el árbol de decisión, entrenarlo y emplearlo para hacer predicciones. Así que en la celda siguiente se presenta un conjunto de instrucciones necesarias para poder luego emplear las funciones del paquete scikit-learn. El código está explicado línea a línea. Léelo y pregunta al profesor/a si hay cualquier cosa que no entiendas.

In [6]:

```
feat_index = [0, 1, 2] # define los índices que u
                    # dataframe que van a extr
                    # o variables predictoras.
                    # tres primeros.

features = [data_train.columns[i] for i in feat_index] # crea una lista con los n
                                                    # predictoras, usando una

target = data_train.columns[3] # guarda en una variable e
                               # objetivo

X_train = np.array(data_train[features]) # guarda en la variable X
Y_train = np.array(data_train[target])  # variables predictoras y
                                        # los valores de la variab
                                        # vectores (arrays) que es
                                        # que necesitan luego las
                                        # emplear.
```

Ahora ya estamos preparados para construir nuestro primer árbol de decisión. Para ello, consulta la página de la función `DecisionTreeClassifier()` y sigue los siguientes pasos:

- primero aplica la función `DecisionTreeClassifier()` sin especificar el valor de ninguno de sus inputs (tomará todos los valores por defecto) para construir tu primer modelo, que puedes llamar `modelo_1`.
- segundo, entrena el modelo que has construido aplicando el método `fit()` al modelo que acabas de generar (a `modelo_1`) y pasándole los valores de las variables predictoras (`X_train`) y objetivo (`Y`) que has generado antes.
- tercero, dibuja el árbol resultante, utilizando la función `plot_tree()`. Para ello utiliza la sintaxis que se sugiere en [el ejemplo de la documentación de scikit-learn](#).

In [ ]:

```
# TAREA 7
```

```
# ----- Aquí debe empezar tu código. Extensión aproximada: 3 líneas de código
modelo_1 = tree.DecisionTreeClassifier()
modelo_1 = modelo_1.fit(X_train, Y_train)
tree.plot_tree(modelo_1)
# ----- Aquí debe terminar tu código
```

### TAREA 8. Análisis del árbol de decisión.

Responde a las siguientes preguntas:

- ¿Te parece un árbol de decisión sencillo?
- ¿Podrías estimar cuál es la profundidad del árbol? (entendida como el número de ramificaciones que hay entre el primer nodo y el último).

Ahora vamos a examinar las características del modelo generado, utilizando tres métodos que, aplicado a nuestro modelo, nos ofrecerán información valiosa.

- el método `get_depth()` te ofrece la profundidad del árbol generado por tu modelo.
- el método `get_n_leaves()` te ofrece el número de hojas del árbol generado por tu modelo. Esta, junto con el valor anterior, te ayudan a cuantificar la complejidad del modelo.

In [8]:

```
# TAREA 9
# Aquí vamos a estudiar algunas propiedades del modelo generado.
print(modelo_1.get_depth())           # Aplica el método get_depth() al modelo gen
print(modelo_1.get_n_leaves())       # Aplica el método get_leaves() al modelo ge
```

20  
683

## Evaluando el modelo

Ahora queremos saber cómo de bien funciona nuestro modelo de predicción de clases. Para ello, primero importaremos otro conjunto de datos que no haya sido utilizado para el entrenamiento del modelo y que sirva para hacer predicciones y evaluar si nuestro modelo acierta o no cuando predice las distintas clases ('Sobrevive' y 'No sobrevive'). Para ello deberás importar un archivo csv cuyo nombre es `s41598-020-73558-3_sepsis_survival_study_cohort.csv`. Recuerda que para ello deberás emplear la función del paquete `pandas`, `read_csv()`. Puedes llamar a ese conjunto de datos `data_test`, para diferenciarlo del dataset empleado para el entrenamiento, `data_train`.

In [9]:

```
# TAREA 10
# ----- Aquí debe empezar tu código. Extensión aproximada: 3 líneas de código
data_test = pd.read_csv('s41598-020-73558-3_sepsis_survival_study_cohort.csv')
# ----- Aquí debe terminar tu código
```

De manera similar a como hicimos con los datos de entrenamiento, conviene hacer una breve exploración de los datos de test, para detectar posibles errores en los datos, desequilibrios de clase o cualquier otra información que puede ser relevante. Así que procede de manera similar a antes y haz un `info()` y un `describe()` a `data_test` (puedes utilizar la función `print()` para que te muestre el resultado de cada función). Después, dibuja histogramas para cada uno

de los campos presentes en `data_test` utilizando la función de `matplotlib.pyplot` llamada `hist()`, como has hecho con anterioridad.

```
In [10]: # TAREA 11
# ----- Aquí debe empezar tu código. Extensión aproximada: entre 3 y 7 líneas de código
print(data_test.info()) # Aplica el método info
print(data_test.describe()) # Aplica el método describe
[data_test[data_test.columns[0]].unique(), # Aplica el método unique
 data_test[data_test.columns[1]].unique(),
 data_test[data_test.columns[2]].unique(),
 data_test[data_test.columns[3]].unique()]
# ----- Aquí debe terminar tu código
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19051 entries, 0 to 19050
Data columns (total 4 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   age_years                             19051 non-null  int64
1   sex_0male_1female                     19051 non-null  int64
2   episode_number                         19051 non-null  int64
3   hospital_outcome_1alive_0dead         19051 non-null  int64
dtypes: int64(4)
memory usage: 595.5 KB
None
```

	age_years	sex_0male_1female	episode_number
count	19051.000000	19051.000000	19051.000000
mean	72.503228	0.448585	1.396252
std	18.614470	0.497363	0.783749
min	0.000000	0.000000	1.000000
25%	65.000000	0.000000	1.000000
50%	77.000000	0.000000	1.000000
75%	85.000000	1.000000	2.000000
max	100.000000	1.000000	5.000000

	hospital_outcome_1alive_0dead
count	19051.000000
mean	0.810719
std	0.391742
min	0.000000
25%	1.000000
50%	1.000000
75%	1.000000
max	1.000000

```
Out[10]: [array([ 7, 17, 70, 76, 8, 41, 60, 89, 81, 55, 66, 63, 33,
          73, 48, 11, 85, 44, 45, 79, 71, 50, 46, 78, 87, 83,
          59, 13, 90, 64, 6, 68, 72, 53, 82, 75, 67, 88, 91,
          74, 49, 18, 32, 65, 54, 56, 5, 61, 69, 62, 86, 9,
          52, 40, 19, 57, 39, 80, 96, 27, 58, 37, 77, 42, 51,
           3, 22, 35, 36, 47, 23, 28, 84, 25, 94, 21, 29, 12,
          24, 16, 92, 30, 38, 93, 43, 26, 15, 95, 10, 31, 100,
          34, 97, 98, 20, 2, 1, 4, 99, 0, 14]),
          array([1, 0]),
          array([1, 2, 3, 4, 5]),
          array([1, 0])]
```

Para mostrar los histogramas de cada una de las columnas de `data_test` puedes emplear primero la función `plt.figure()` para generar una nueva figura, luego aplicar la función `plt.hist()` a la columna de `data_test` que quieras representar y luego poner un título a la gráfica con la función `plt.title()`. Cuando hayas hecho esto para cada columna, no olvides escribir la función `plt.show()` en la última línea, sin pasarle parámetros, para que se muestren todas las figuras.

```
In [11]:
```

```

# TAREA 12
# ----- Aquí debe empezar tu código. Extensión aproximada: 1 y 3 líneas de código por
plot1 = plt.figure(1) # Genera la figura con plt.figure()
plt.hist(data_test[data_test.columns[0]]) # Utiliza plt.hist() a cada columna de
plt.title(str(data_test.columns[0])) # Ponle título a la figura con plt.tit

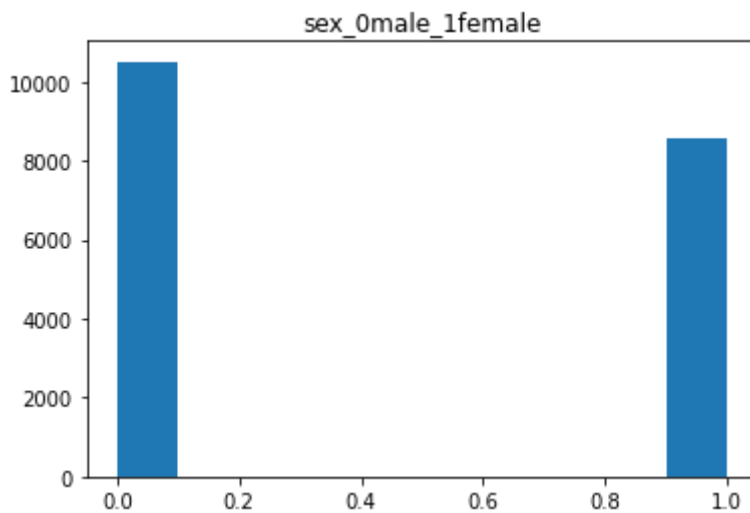
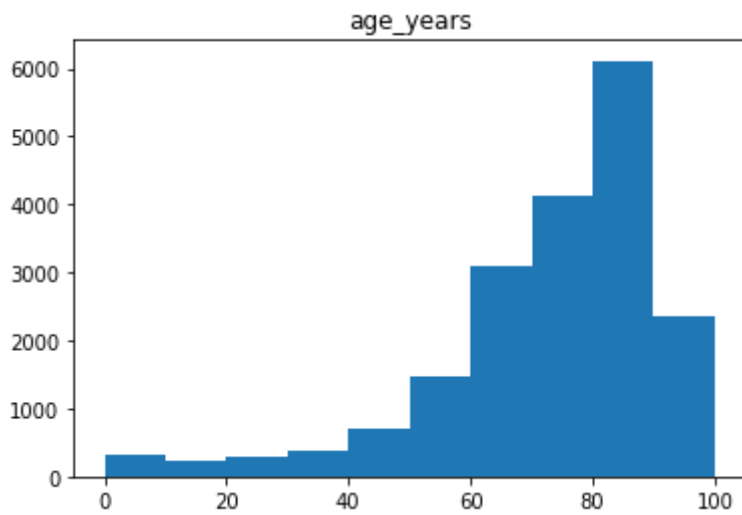
plot2 = plt.figure(2)
plt.hist(data_test[data_test.columns[1]])
plt.title(str(data_test.columns[1]))

plot3 = plt.figure(3)
plt.hist(data_test[data_test.columns[2]])
plt.title(str(data_test.columns[2]))

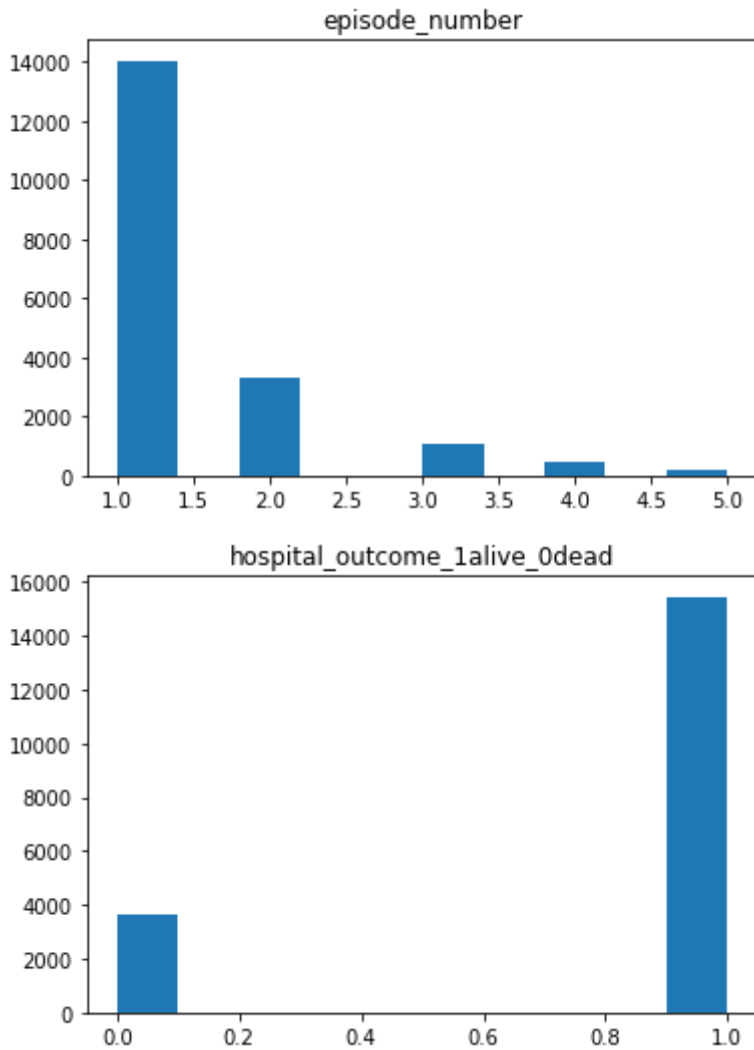
plot4 = plt.figure(4)
plt.hist(data_test[data_test.columns[3]])
plt.title(str(data_test.columns[3]))

plt.show() # Utiliza plt.show()
# ----- Aquí debe terminar tu código

```







Ahora construimos las variables a emplear para la validación del modelo.

```
In [12]: X_test = np.array(data_test[features]) # guarda en La v
         Y_test = np.array(data_test[target])
```

Para evaluar cómo de bien clasifica el modelo tenemos varias opciones:

- la más sencilla es emplear el método `score()` aplicado a tu modelo. Requiere que le pases como parámetros datos de test sobre los que el modelo no ha sido entrenado, para valorar cuánto acierta en sus predicciones.
- la versión 'manual' consiste en pedirle al modelo que genere la predicción y luego intentar hacer un gráfico que nos permita comparar visualmente la predicción de clase de la variable objetivo que sus valores reales. En la celda que tienes a continuación deberás seguir ambos caminos.

```
In [13]: # TAREA 13
         # ----- Aquí debe empezar tu código.

         # PRIMERA OPCIÓN: empleo del método score() a tu modelo (1 línea de código). Recuerda
         # pasarle como inputs el valor de las variables predictoras en la zona de test, X_te
         # de la variable a predecir Y_test

         print(modelo_1.score(X_test, Y_test))

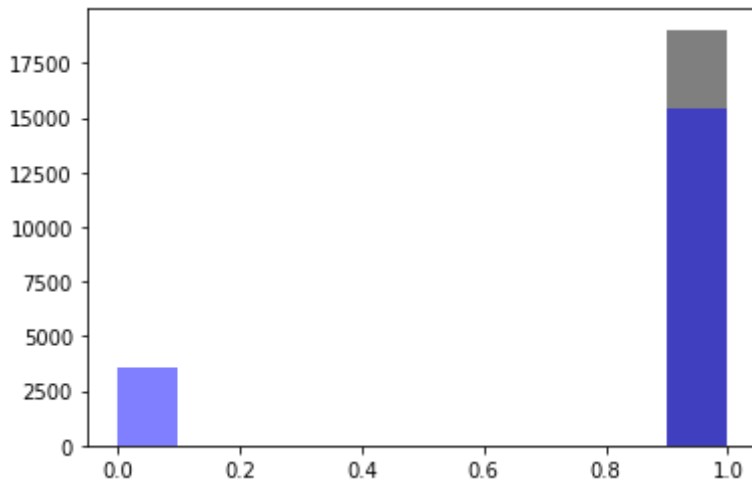
         # SEGUNDA OPCIÓN: empleo del método predict() al modelo generado, pasándole el valor
         # variables predictoras, X_test (1 línea de código). Luego grafica dos histograma ju
```

```
# (basta con que utilices dos veces seguidas la función plt.hist()) que representen,
# por un lado, la predicción del modelo y, por otro, los valores reales de la variable
# que tenemos almacenada en Y_test. Para poder comparar visualmente las gráficas
# utiliza el parámetro alpha = 0.5, como input a la función plt.hist() para añadir
# transparencia a la gráfica y el parámetro color = 'k', como otro input a plt.hist()
# el color de cada gráfica, 'k' quiere decir negro, 'r' rojo, 'g' verde, etc.

predicción = modelo_1.predict(X_test)
plt.hist(predicción, alpha = 0.5, color = 'k')
plt.hist(Y_test, alpha = 0.5, color = 'b')
# ----- Aquí debe terminar tu código
```

0.8108760694976641

```
Out[13]: (array([ 3606.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
           0., 15445.]),
          array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
          <a list of 10 Patch objects>)
```



## TAREA 14

- ¿Qué resultado arroja el método `score()`? Ese valor es un porcentaje, en tanto por uno, de los valores cuya predicción es correcta. ¿Te parece un buen resultado? ¿Por qué?
- A la vista de los histogramas que has dibujado, ¿dirías que el clasificador es bueno? ¿donde se está equivocando más, al intentar predecir si sobrevive o al intentar predecir que no sobrevive? ¿Por qué crees que puede estar comportándose así? ¿Podemos utilizarlo en la vida real? ¿Cuáles serían las consecuencias?

## Revisando el modelo

Vamos a afinar un poquito más el modelo generado, buscando que haga mejores predicciones. Para ello vamos a balancear las clases en los datos de entrenamiento, esto es, poner el mismo número de elementos clasificados como 'Sí sobrevive' como que 'No sobrevive' en los datos de entrenamiento de nuestro nuevo modelo. Así no estará sesgado por haber aprendido a reconocer *demasiado* una de las clases por encima de la otra.

```
In [15]: # Aquí vamos a construir un dataset totalmente balanceado para mejorar los resultados
# Básicamente tomamos los N primeros positivos y los N primeros negativos y luego los
# aparejamos mezclados en las matrices de valores.

data_surv = data_train[data_train.hospital_outcome_1alive_0dead == 1] #
data_not_surv = data_train[data_train.hospital_outcome_1alive_0dead == 0] #
data_balanced = pd.concat([data_not_surv, data_surv.iloc[:len(data_not_surv)]] #
```

```

data_balanced_shuffled=data_balanced.sample(frac=1)
data_train_2 = data_balanced_shuffled

X_train_2 = np.array(data_train_2[features])
Y_train_2 = np.array(data_train_2[target])

```

In [ ]:

```

# Ahora debes reproducir las tareas 7 y 13, pero esta vez empleando el nuevo modelo
# TAREA 15 (reproducción de la tarea 7 para el nuevo modelo) 3 líneas de código

modelo_2 = tree.DecisionTreeClassifier()
modelo_2 = modelo_2.fit(X_train_2, Y_train_2)
tree.plot_tree(modelo_2)

```

In [17]:

```

# TAREA 16 (reproducción de la tarea 13 para el nuevo modelo)

print(modelo_2.score(X_test, Y_test))
predicción_2 = modelo_2.predict(X_test)
plt.hist(predicción_2, alpha = 0.5, color = 'k')
plt.hist(Y_test, alpha = 0.5, color = 'b')

```

0.43472783580914387

**TAREA 17** A la vista de los resultados anteriores, responde las siguientes cuestiones:

- El nuevo modelo, ¿mejora o empeora la precisión de sus predicciones?
- ¿Cuándo se equivoca más? ¿Al intentar predecir que 'Sí sobrevive' o al intentar predecir que 'No sobrevive'?
- Si hubieras de elegir uno de los dos modelos que has calculado para ayudar al hospital de tu barrio, ¿cuál elegirías? ¿Por qué?

# **DIAGNÓSTICO MÉDICO PRECOZ MEDIANTE IA**

## **ANEXO 10**

**Instrucciones proceso 4**

## Proceso 4. Refinamos nuestro modelo con nuevas herramientas

Para desarrollar modelos de IA realistas es necesario, entre otras cosas, analizar grandes cantidades de datos. Ello introduce la necesidad de utilizar nuevas herramientas.

La actividad que inicias ahora consiste en desarrollar un modelo de IA realista utilizando los saberes y competencias adquiridas en los procesos anteriores, así como nuevas herramientas que te permitirán desarrollar un agente inteligente en Python, experimentando con distintos comandos y funcionalidades del agente inteligente a través de un Jupyter Notebook que te servirá de guía.

Este proceso 4 se realizará en un aula con ordenadores con conexión a internet **por parejas**. Cada pareja deberá entregar una copia del dossier que contiene la descripción de las distintas tareas a realizar. En ese documento que te servirá de guía didáctica encontrarás dos tipos de tareas que deberéis realizar en clase, con la ayuda de los/las compañeros/as y el profesorado:

- Tareas de programación (para construir y simular modelos de IA), donde se te muestran documentos de referencia del lenguaje Python, para que puedas entender la funcionalidad de cada comando, al tiempo que ves ejemplos de la sintaxis correspondiente y de cómo se aplica en una diversidad de ejemplos resueltos. Por supuesto, el profesorado te guiará a través de esos materiales y podrás preguntar cuanto necesites. En estas tareas se te sugiere una extensión aproximada el código que se te pide que generes en cada caso, a modo de orientación.
- Tareas de reflexión sobre el funcionamiento de los distintos modelos basados en IA. Es importante que justificuéis las respuestas y profundicéis, en un debate grupal con los miembros del equipo con el que desarrollas las tareas, en los aspectos de fondo que se están haciendo intervenir a través de las preguntas que se os plantean.

Una vez finalizado el dossier, deberás descargarlo y facilitarle el correspondiente archivo con extensión 'ipynb' al profesorado.

Para comenzar esta tarea debéis hacer click en el siguiente [enlace](#). Se te abrirá un entorno de desarrollo donde deberás hacer click en la columna de la izquierda sobre el archivo que contiene el Jupyter notebook sobre el que vais a trabajar, que lleva por nombre 'SA\_IA\_Supervivencia\_Sepsis.ipynb'.

**Nota 1:** no todo el código que se hace intervenir en el dossier que se os ha facilitado tenéis que programarlo vosotros, hay alguna parte que ya está programada y sobre la que no habrá más discusión que la que pueda suscitar su comprensión en el seno de vuestro equipo de trabajo. Presta atención a los enunciados de las tareas.

**Nota 2:** descargad el documento de trabajo cada vez que resolváis una tarea.

# **DIAGNÓSTICO MÉDICO PRECOZ MEDIANTE IA**

## **ANEXO 11**

**Escala de valoración presentación agente  
inteligente**



# PRESENTACIÓN DE LOS MODELOS DE IA GENERADOS

## ESCALA DE VALORACIÓN

Grupo:

Integrantes:

	1	2	3	4
1. La exposición presenta una estructura clara y completa.				
2. Reflexiona, con claridad y coherencia, sobre las aplicaciones de la IA en el mundo de la medicina.				
3. Presenta, de forma clara y precisa, el objetivo del modelo de IA desarrollado.				
4. Describe de forma clara las características esenciales de los datos de entrada y de salida del modelo.				
5. Presenta, de forma clara y completa, los resultados del análisis exploratorio de los datos de entrada del modelo.				
6. Describe con claridad cómo funciona el modelo generado.				
7. Describe de manera crítica y completa los resultados de los modelos generados.				

Aclaraciones:

- la escala toma valores entre 1 y 4, representando 1 el mínimo valor de desempeño y 4 el máximo.
- Todos los integrantes del equipo deberán contribuir a la presentación. El/la docente podrá intervenir para reequilibrar este aspecto cuando proceda.