Universidad Politécnica
de Madrid

Escuela Técnica Superior de
Ingenieros Informáticos

Máster Universitario en Inteligencia Artificial

Trabajo Fin de Máster

Análisis Post-Covid19 con Herramientas de Aprendizaje Automático

Autor: Joaquín Jiménez López de Castro
Tutores: Concha Bielza Lozoya y Pedro Larrañaga Múgica

Madrid, julio 2022

Este Trabajo Fin de Máster se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Máster
Máster Universitario en Inteligencia Artificial
Título: Análisis Post-Covid19 con Herramientas de Aprendizaje Automático

julio 2022

Autor: Joaquín Jiménez López de Castro
Tutor: Concha Bielza Lozoya y Pedro Larrañaga Múgica
Departamento de Inteligencia Artificial
ETSI Informáticos
Universidad Politécnica de Madrid

Agradecimientos

En primer lugar quiero agradecer a la Fundación BBVA y al Ministerio de Ciencia e Innovación a través del PID2019-109247GB-I00, que han financiado este proyecto.

En segundo lugar, a la Fundación Jiménez Díaz que ha proporcionado los datos que han hecho posible el trabajo, con una especial dedicación a los miembros del equipo médico, que ha brindado conocimiento y ánimos.

También a mis tutores Pedro y Concha, que han puesto todo su empeño en ayudarme a sacar adelante este trabajo, sin dejar de animarme a explorar también formas creativas de realizarlo.

Finalmente a mi familia, que me ha puesto los pies en la tierra en los momentos más duros de la realización del trabajo.

Resumen

Han pasado algo más de dos años desde el inicio de la pandemia por Covid-19 en Wuhan (China). Desde entonces, el panorama ha evolucionado en muchos aspectos, de entre los que podemos destacar el cambio de perfil de los pacientes, donde se diferencia la vacunación, así como el del propio virus SARS-CoV-2, que ha mutado en diversas variantes, donde las principales son Wuhan (la original), Alfa, Delta y Ómicron.

Este trabajo, que forma parte del proyecto *Outcome Prediction and Treatment Efficiency in Patients Hospitalized with COVID-19 in Madrid: A Bayesian Network Approach*, financiado por la Fundación BBVA, pretende realizar un estudio observacional del impacto de distintas variables en la mortalidad de pacientes de Covid-19, comparando los resultados por las distintas olas de la pandemia en la Comunidad Autónoma de Madrid (CAM), agrupadas según su variante del virus mayoritaria. Para ello se cuenta con un conjunto de datos proporcionado por la Fundación Jiménez Díaz, que contiene registros de pacientes de 4 de sus hospitales en la CAM, abarcando desde el 27 de febrero de 2020 hasta el 23 de marzo de 2022.

Como parte del estudio, se ha construido un *dashboard* interactivo que permite comparar por agrupación de olas de la pandemia, la importancia de distintas variables de los pacientes para la predicción de mortalidad. Para obtener las estimaciones de la importancia de cada variable se han usado *random forests*, *Generalized Boosted Models* y un método univariante. También, se han construido redes Bayesianas discretas usando los primeros registros de los pacientes correspondientes a las variantes Delta y Ómicron, que se han utilizado para visualizar el impacto entre variables, así como encontrar perfiles de riesgo en los pacientes.

Otro trabajo ha sido la comparativa del rendimiento de distintos modelos de aprendizaje automático en la predicción de la mortalidad para las distintas agrupaciones de olas, entre los que se incluyó una nueva rama de clasificadores Bayesianos desarrollada como parte de este proyecto. Se determinó que el error mínimo de predicción de mortalidad ha aumentado en las sucesivas olas de la pandemia, especialmente en la correspondiente a la variante Ómicron. También, se observó que de forma general, los modelos no interpretables tenían un mejor rendimiento en clasificación. Teniendo esto en cuenta, también se añade como parte de este trabajo el desarrollo de un algoritmo de búsqueda de explicaciones contrafactuales, con el fin de hacer este tipo de modelos más interpretable.

Abstract

A little over two years have passed since the start of the Covid-19 pandemic in Wuhan (China). Since then, the panorama has evolved in many aspects, among which we can highlight the change in the profile of patients, where vaccination differs, as well as that of the SARS-CoV-2 virus itself, which has mutated into various variants, where the main ones are Wuhan (original), Alpha, Delta and Omicron.

This work, which is part of the project "Outcome Prediction and Treatment Efficiency in Patients Hospitalized with COVID-19 in Madrid: A Bayesian Network Approach", funded by *Fundación BBVA*, aims to carry out an observational study of the impact of different variables on the Mortality of Covid-19 patients, comparing the results for the different waves of the pandemic in the Autonomous Community of Madrid (ACM), grouped according to their majority virus variant. For this purpose, *Fundación Jiménez Díaz* provided a dataset, which contains patient records from 4 of its hospitals in the Autonomous Community of Madrid (ACM), covering from February 27, 2020 to March 23, 2022.

As part of the study, an interactive dashboard has been built that allows comparing the importance of different patient variables for predicting mortality by grouping waves of the pandemic. To obtain the estimates of the importance of each variable, random forests, Generalized Boosted Models and an univariate method have been used. Also, discrete Bayesian networks have been built using the first records of patients corresponding to the Delta and Omicron variants. They have veen used to visualize the impact between variables, as well as to find risk profiles in patients.

Another work has been the comparison of the performance of different machine learning models in predicting mortality for different wave groupings, including a new branch of Bayesian classifiers developed as part of this project. It was determined that the minimum mortality prediction error has increased in the successive waves of the pandemic, especially in the one corresponding to the Omicron variant. Also, it was observed that in general, the non-interpretable models had a better performance in classification. Taking this into account, the development of a search algorithm for counterfactual explanations is also added as part of this work, in order to make this type of model more interpretable.

Tabla de contenidos

Capítulo 1

Introducción

En esta sección se pretende explicar el contexto del desarrollo de este Trabajo de Fin de Máster, así como los objetivos y recursos principales proporcionados para su cumplimiento.

1.1. Descripción del proyecto

1.1.1. Colaboración con FBBVA

En el momento de escribir este documento, han pasado ya más de dos años desde que se diagnosticaron los primeros pacientes de Covid-19, primero en Wuhan (China) en diciembre de 2019, y poco después en España, en enero de 2020. Desde marzo de 2021, se han registrado en todo el mundo aproximadamente 545.000.000 casos confirmados de Covid-19, y se han confirmado 6.330.000 muertes en personas diagnosticadas con Covid-19 (Ritchie y col., 2022), demostrando ser una enfermedad de suma importancia, debido a su alto grado de contagiosidad y mortalidad que ha provocado una crisis social y sanitaria a nivel global.

Es por ello que poco después de la aparición del Covid-19, en 2020, la Fundación BBVA (FBBVA) adjudicó una serie de ayudas a cuatro proyectos nacionales de investigación (de entre 150 presentados a la convocatoria competitiva) en Covid-19 dentro de la sección de *Big Data* e Inteligencia Artificial. Entre los proyectos que fueron escogidos, se encuentra *Outcome Prediction and Treatment Efficiency in Patients Hospitalized with COVID-19 in Madrid: A Bayesian Network Approach*[1], que arrancó el 9 de octubre de 2020 (duración de 2 años), dirigido por Concha Bielza del DIA-UPM, y del que forma parte el desarrollo de este trabajo.

[1]Fundación BBVA (2020). Adjudicación de ayudas a equipos de investigación en Covid-19. URL: https://www.fbbva.es/noticias/adjudicadas-4-ayudas-a-equipos-de-investigacion-cientifica-sars-cov-2-y-covid-19-en-big-data-e-inteligencia-artificial/

Como se puede deducir del título, el proyecto estaba originalmente enfocado a la predicción del resultado del ingreso de pacientes de Covid-19 en Madrid en cuanto a diversos criterios: mortalidad, necesidad de ventilación, ingreso a la Unidad de Cuidados Intensivos (UCI), etc. También se pretendía analizar el impacto del uso de diferentes medicamentos en los pacientes diagnosticados de Covid-19.

1.1.2. Colaboración con la Fundación Jiménez Díaz

Para llevar a cabo un proyecto de este tipo, se necesitan unos datos sobre los que aplicar las técnicas de aprendizaje automático. De esta primera necesidad surgió la colaboración con la Unidad de Investigación Clínica del Hospital Universitario Fundación Jiménez Díaz, perteneciente al Instituto de Investigación Sanitaria - Fundación Jiménez Díaz (FJD).

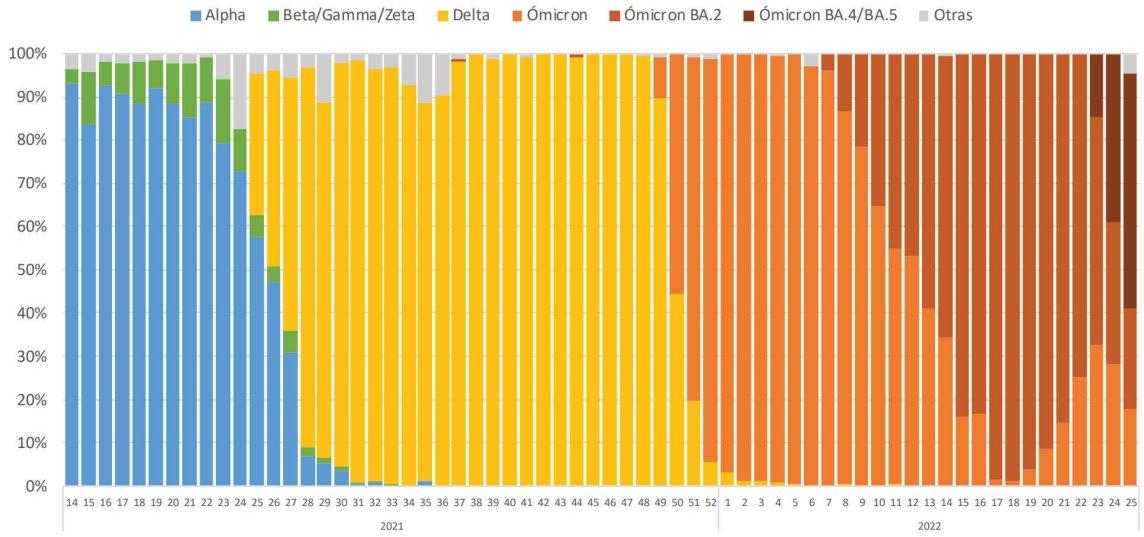
La FJD donó un conjunto de datos de pacientes completamente anonimizado, incluyendo varios diagnosticados por Covid-19. Este se envió en dos tandas: el primer año se enviaron datos de pacientes de 4 hospitales de la Fundación cuyo ingreso abarcaba desde el 27 de febrero de 2020 hasta el 21 de febrero de 2021, con 277732 registros; el segundo año se enviaron datos de pacientes cuyo ingreso abarcaba desde el 15 de febrero de 2021 hasta el 23 de marzo de 2022, con 174687 registros. Al combinar ambos conjuntos de datos, se tiene un total de 446555 registros y 532 variables. Cabe mencionar que los datos del segundo año incluían información en lenguaje natural de la vacunación de los pacientes, así como de casos de neumonía bilateral, una de las peores complicaciones en los pacientes de Covid-19, que además puede producir secuelas (Wu y col., 2021).

Además de para obtener datos, en esta colaboración se busca la interacción y consulta con los médicos a la hora de tomar decisiones apropiadas con los datos, así como concretar los objetivos del proyecto. Por este motivo, se mantuvieron algunas reuniones con el equipo médico asociado al proyecto y se intercambiaron varios correos.

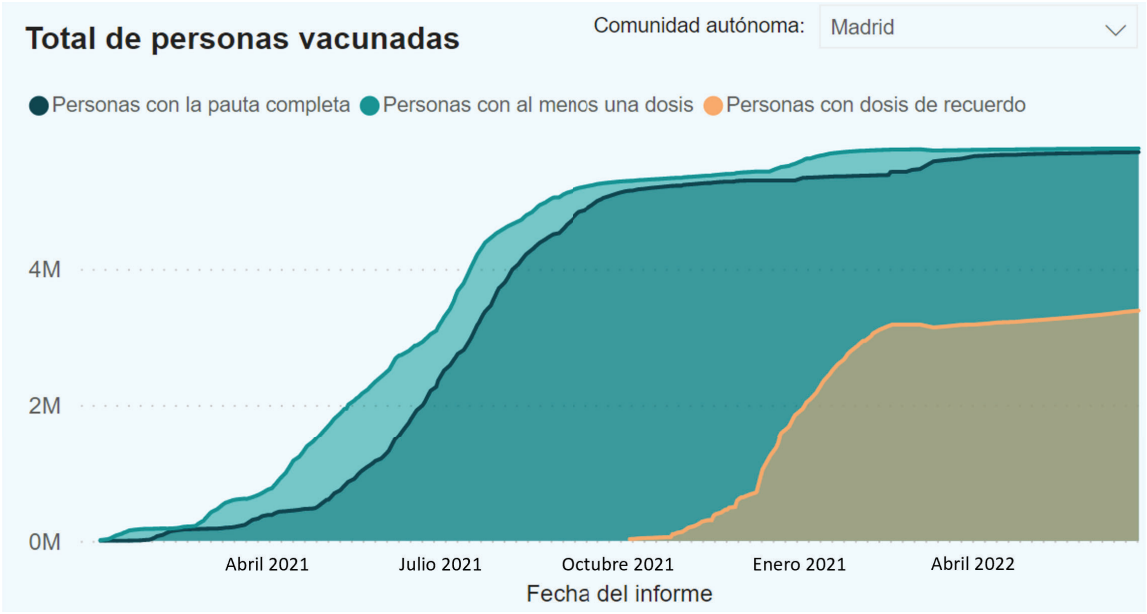
1.1.3. Cambios en el contexto del proyecto

Un año después del inicio del proyecto, han surgido nuevas circunstancias y objetivos para el desarrollo del mismo. El cambio más importante respecto al año anterior, es el del propio SARS-CoV-2, que ha ido mutando a través de las distintas olas de la pandemia, cambiando sus propiedades y riesgos asociados. También ha cambiado el perfil de la población, con la aparición de las vacunas en las últimas olas (Ritchie y col., 2022). Esto queda ilustrado en la Figura 1.1a, donde se ve la evolución de la variante del virus mayoritaria en Madrid a lo largo del tiempo, así como en la Figura 1.1b, que demuestra la evolución del número de personas vacunadas en Madrid.

Por este motivo, una nueva consideración en el proyecto es la variante del virus



(a) Variante mayoritaria de Covid-19 en Madrid[2].



(b) Total de personas vacunadas en Madrid[3].

Figura 1.1: Evolución de la pandemia en Madrid

[2]Ministerio de Salud (2022). Informe epidemiológico semanal. URL: https://www.comunidad.madrid/sites/default/files/doc/sanidad/epid/informe_epidemiologico_semanal.pdf

[3]Ministerio de Salud (2022). Cuadro de mando resumen de datos de vacunación URL: https://www.sanidad.gob.es/profesionales/saludPublica/ccayes/alertasActual/nCov/pbiVacunacion.htm

estudiada, así como la información de vacunación de los pacientes.

Otro cambio se ha dado tras la experiencia con los modelos desarrollados y objetivos perseguidos en el primer año del proyecto. En un principio, se dio prioridad al desarrollo de modelos predictivos, con un enfoque especial en la clasificación de pacientes como críticos o no críticos con el objetivo de ser utilizados posteriormente en el ámbito médico y evitar la saturación de los hospitales. Dichos modelos fueron desarrollados con éxito en cuanto a rendimiento e interpretabilidad de los modelos utilizados, como mostraron Bernaola y col. (2022), así como Riaño (2021) en su tesis del Máster en Ciencia de Datos de la UPM, que formó parte del primer año de desarrollo de este proyecto.

Sin embargo, quedó clara posteriormente la dificultad de integrar con éxito este tipo de modelos en el ámbito clínico, puesto que la mejora de las predicciones de los médicos al contar con el modelo, no se consideraba suficientemente alta como para invertir el tiempo necesario para integrarlo en el ámbito clínico. Además, siendo que la enfermedad está en constante cambio, así como el perfil de la población, para cuando el modelo esté implantado en el ámbito clínico, es posible que el rendimiento del modelo decrezca significativamente.

Por todo ello, la prioridad de desarrollar modelos para la predicción del resultado del ingreso en futuros pacientes en el ámbito clínico es menor, y se le da mayor prioridad a su uso para un estudio observacional.

1.2. Objetivos

Tras poner en contexto el proyecto en el que se basa este Trabajo de Fin de Máster, en esta sección se discuten los objetivos señalados para el mismo, que se procuró que estuvieran en concordancia con los del propio proyecto de la FBBVA. Para seleccionar los posibles objetivos, se consultó con un miembro del equipo médico de la FJD, que tomó el rol de *stakeholder* e indicó qué líneas de investigación le parecían de mayor interés sobre el conjunto de datos proporcionado. A priori, hay un conjunto de líneas de investigación a seguir que fueron descartadas:

- **Las relacionadas con el tratamiento de los pacientes**. Aunque el conjunto de datos contiene información de los medicamentos suministrados, se determinó que para hacer un estudio del impacto de los medicamentos, era necesario tener en cuenta el aspecto dinámico del conjunto de datos, para buscar así una relación causa-efecto. Sin embargo, solamente se disponía de la fecha de prescripción de los medicamentos, no la fecha de administración, y tampoco se disponía de información sobre cuándo dejaba de suministrarse cada medicamento. Por tanto se decidió descartar los objetivos relacionados con la determinación del

impacto de diversos medicamentos.

- **Las relacionadas con la evolución de los pacientes**. Muchas otras de las líneas de investigación propuestas estaban relacionadas con un análisis de la estancia completa de los pacientes, para responder preguntas como "¿Por qué la inflamación ha vuelto a subir, si se había estabilizado?". El motivo por el que se descartó esta rama de objetivos es que otro colaborador del grupo de investigación CIG-UPM[2] ya iba a hacer un análisis con modelos dinámicos que usarían datos de la estancia completa de los pacientes. Por tanto, se le asignó este grupo de objetivos, para poder enfocarse en este trabajo en otros objetivos de interés de forma más eficiente.

El tercer grupo de líneas de investigación planteado por el miembro del equipo médico era el relacionado con la **búsqueda de perfiles y factores de riesgo**. Concretando más, se tenía interés en:

- Evolución de factores de riesgo por olas, con especial interés en la edad.
- Impacto de distintas variables en la mortalidad según la variante y factores de riesgo asociados, así como presencia de niveles altos de inflamación.
- Impacto de la vacunación en la mortalidad de pacientes ingresados.
- Predicción de mortalidad, tiempo de ingreso, estancia en la UCI, Unidad de Cuidados Intermedios Respiratorios (UCIR), así como necesidad de ventilación.

Utilizando estas líneas de investigación propuestas como base, se definieron los siguientes objetivos para este TFM:

1. Estudio de la evolución por olas del impacto de las distintas variables en la mortalidad (con mayor interés en la edad y vacunación), tiempo de ingreso, estancias en UCI y UCIR; con interés en descubrir posibles nuevos perfiles de riesgo.
2. Estudio del rendimiento de distintos modelos predictivos para las variables de interés señaladas en el objetivo anterior.
3. Investigar el uso de los modelos predictivos para extracción de conocimiento y la factibilidad de su uso en el ámbito clínico.

Sobre estos objetivos, cabe realizar algunas aclaraciones. En primer lugar, el estudio de la necesidad de ventilación quedó descartado por la determinación del equipo médico de que la información no estaba correctamente codificada en el conjunto de datos. También había información en lenguaje natural sobre la ventilación de los pacientes, pero únicamente para las últimas dos variantes del virus, por lo que no

[2] *The Computational Intelligence Group.* `http://cig.fi.upm.es/`

servía para el estudio global. En segundo lugar, para lograr el segundo y tercer objetivo hay un especial interés en el uso de redes Bayesianas (Pearl, 1988; Koller y col., 2009), debido a su potencia para predicciones y extracción de conocimiento, que se explorará en mayor profundidad en la Sección 2.2.1. Finalmente, como se ha mencionado antes, los modelos predictivos demuestran ser difíciles de integrar en el ámbito clínico; por ello, como parte del tercer objetivo, se explorará el uso de contrafactuales para mejorar la interpretabilidad de las salidas de los modelos predictivos.

1.3. Estructura del documento

A continuación se enumeran los distintos capítulos de este documento junto con una breve descripción de los mismos.

1. Introducción. La introducción al documento.
2. Estado del arte. En este capítulo se exploran los distintos trabajos de la literatura relacionados con el cumplimiento de estos objetivos.
3. Desarrollo. En este capítulo se explica el trabajo realizado sobre el conjunto de datos proporcionado.
    - 3.1. Preprocesamiento del conjunto de datos. Aquí se detallan las transformaciones realizadas sobre el conjunto de datos, así como su estructura final.
    - 3.2. Importancia y selección de variables por olas. En esta sección se detalla la construcción de un *dashboard* interactivo que muestra la evolución de la relevancia de las variables en cada variante del virus y justifica la selección de variables para el resto de secciones.
    - 3.3. Estudio de variantes Ómicron y Delta con redes Bayesianas discretas. Construcción de redes Bayesianas discretas con discretizaciones proporcionadas por el equipo médico para el análisis de las dos variantes más recientes.
    - 3.4. Diseño de clasificadores Bayesianos Híbridos semiparamétricos. Adaptación del *software* de redes Bayesianas semiparamétricas híbridas *PyBNesian* de Atienza y col. (2022) a la API de clasificadores de *scikit-learn* (Pedregosa y col., 2011); con el fin de evaluarse como un posible nuevo tipo de modelo interpretable para la predicción de mortalidad en Covid-19.
    - 3.5. Construcción de modelos predictivos por olas. Detalle de la obtención y comparativa de rendimiento de distintos modelos predictivos aplicados al conjunto de datos.
    - 3.6. Explicaciones contrafactuales. Descripción del diseño e implementación de una adaptación del algoritmo de Dandl y col. (2020) de búsqueda de expli-

caciones contrafactuales para cualquier modelo. Fue desarrollado como un algoritmo para explicar la salida de modelos predictivos en el ámbito clínico, especialmente los no interpretables.

4. Conclusiones. Evaluación del grado de cumplimiento de los objetivos establecidos y discusión de posibles líneas futuras seguir.

Capítulo 2

Estado del arte

En esta sección se hace una revisión de otros trabajos del estado del arte relevantes al cumplimiento de los objetivos.

2.1. Preprocesamiento de los datos

Es frecuente en el contexto clínico que los conjuntos de datos requieran realizar una serie de transformaciones de forma previa a su uso en conjunto con técnicas de aprendizaje automático. El trabajo de Ferrao y col. (2016) señala las cinco dificultades más importantes asociadas al procesamiento de este tipo de conjunto de datos:

1. Recogida e integración de datos. Referido a la tarea de selección de información relevante de los pacientes, así como la homogeneización de la representación de las variables.

2. Manejo de variables de distintos tipos. Es frecuente que estos conjuntos de datos mezclen variables de diferentes tipos: categóricas, ordinales, y continuas. Es preciso identificarlas y manejar esta heterogeneidad, ya que tiene un impacto posteriormente en los procesos de imputación, selección de variables y predicción, entre otros (Bellazzi y col., 2008).

3. Redundancia y distinta granularidad de las variables. Otro problema común es que se usen distintas terminologías e incluso variables para nombrar el mismo concepto, así como la posibilidad de que se trate de una misma variable con distintos niveles de granularidad, llegando a ser categórica en algunos casos y continua en otros.

4. Datos faltantes. Los conjuntos de datos de este tipo suelen tener muchas variables con valores no disponibles para algunos pacientes.

5. Presencia de varios valores en algunas variables por paciente. Siendo que estos

conjuntos de datos representan la estancia de un paciente, muchas variables son dinámicas, con varios valores a lo largo de la estancia del paciente, mientras que otras son constantes a lo largo del ingreso (estáticas). Esto se conoce como conjunto de datos multinivel.

La recogida e integración de datos en este caso quedaba a cargo del equipo médico, por lo que no se revisarán aquí los trabajos relevantes.

Para manejar las variables de distintos tipos se suelen considerar dos aproximaciones: la discretización de las variables continuas y el uso de algoritmos híbridos. El problema más evidente que tiene la discretización es la pérdida de información, que se puede mitigar haciendo una discretización con más particiones, siempre y cuando se disponga de suficientes datos para cubrir los distintos niveles; a costa de la explosión combinatoria en los modelos que tienen en cuenta relaciones entre las variables. A cambio, trabajar con modelos discretos generalmente reduce los costes computacionales asociados a los algoritmos de predicción y selección de variables (Talvitie y col., 2019), además de que a menudo son más interpretables (Dash y col., 2011).

Hay varias técnicas para discretizar conjuntos de datos con variables continuas, pero generalmente se pueden diferenciar los métodos supervisados y no supervisados (Dash y col., 2011) y los univariantes y multivariantes (Ramírez-Gallego y col., 2015). Los métodos supervisados son los más tradicionales, y se corresponden con una discretización de las variables que tiene como objetivo minimizar la información perdida con respecto a una variable clase, mientras que los métodos no supervisados simplemente buscan una discretización que no pierda la información más relevante de la distribución original. Los métodos univariantes solo tienen en cuenta la variable a discretizar y la variable clase, en caso de ser supervisados; mientras que los multivariantes tienen en cuenta todas las variables de la distribución conjunta.

En el contexto del Covid-19, apenas se han encontrado trabajos que realicen una discretización del conjunto de datos. La excepción se da en trabajos que utilizan redes Bayesianas discretas, cuyo aprendizaje y uso ya tiene muchos años de investigación (Marcot y col., 2019) y es más sencillo en contraste con el de las redes Bayesianas híbridas y continuas, que no requieren realizar discretizaciones. En la Sección 2.2.1 se darán algunos ejemplos.

En cuanto a la redundancia y diferencias de granularidad, no parece que sea un problema al que se le haya prestado atención en los trabajos de Covid-19. De hecho, no se ha encontrado mención alguna de la necesidad de solventar un problema de este tipo en la literatura. Es decir, la solución predominante sería a priori ignorar el problema. Esto no concuerda con las únicas dos soluciones que propone Ferrao y col. (2016), que son o reducir la granularidad a un nivel común (potencialmente discretizar), o utilizar información del dominio para buscar un nuevo formato representativo

de la variable.

La complejidad inherente al manejo de los datos faltantes en conjuntos de datos del ámbito médico reside principalmente en su heterogeneidad. El trabajo de Rubin (1976) diferencia tres tipos de patrones en la distribución de los valores faltantes del conjunto de datos:

- *Missing completely at random* (MCAR). Es un patrón que se da cuando los valores faltantes de una variable tienen una distribución independiente de los datos observados.
- *Missing at random* (MAR). Es un patrón que se da cuando la distribución de los valores faltantes de una variable es dependiente de al menos una de las variables observadas. Supóngase por ejemplo que fuera menos probable que una persona mayor quisiera comunicar su peso.
- *Missing not at random* (MNAR). Este concepto engloba el resto de los casos, cuando la probabilidad de tener valores faltantes cambia por razones desconocidas.

Los algoritmos generales de imputación, que es el proceso de asignar los valores faltantes de la forma más plausible posible, normalmente asumen una distribución MCAR de los valores faltantes, a menudo implausible (Van Buuren, 2018). Esto puede ser un problema en los conjuntos de datos del ámbito médico, donde es frecuente que exista algún patrón MAR o MNAR en los datos, que típicamente requieren soluciones *ad hoc*, porque se pueden introducir sesgos si se usan los algoritmos generales. También está la opción de eliminar las entradas con valores faltantes; pero al igual que la opción anterior, esto puede no ser apropiado (Allison, 2001). Incluso asumiendo una distribución MCAR, también existe el problema de que, como se comentó anteriormente, el conjunto de datos puede ser multinivel, y esto puede afectar al procedimiento de imputación, puesto que los datos de cada paciente suelen ser similares entre sí (Van Buuren, 2011). Finalmente, también se pueden utilizar modelos que no requieren imputar los datos, típicamente los basados en conjuntos de árboles.

Los algoritmos de imputación que asumen MCAR se pueden clasificar como imputación univariante e imputación multivariante. Los métodos de imputación univariante son los más tradicionales, y consisten en asignar el valor más plausible de la distribución marginal de la variable a imputar, que típicamente es la mediana o media aritmética en el caso de las variables continuas y la moda en el caso de las variables categóricas. La imputación multivariante en cambio, tiene en cuenta la relación entre las variables a imputar. Para ello, se puede seguir un proceso iterativo, comúnmente conocido como imputación múltiple, donde tras realizar inicialmente una imputación univariante de los datos (por ejemplo con la mediana para variables continuas), se construye para cada variable a imputar, un modelo predictivo que

emplea como predictoras el resto de variables, que se utiliza para generar una imputación para la siguiente iteración (Rubin y col., 1991). De esta forma, la imputación de cada variable se basa en el valor del resto (en la iteración anterior).

En el contexto de la predicción de mortalidad de pacientes de Covid-19, típicamente se hace la suposición MCAR y o se imputa o se usan modelos no sensibles a la falta de datos. En la revisión de Bottino y col. (2021) se citan algunos trabajos enfocados a la predicción de mortalidad que emplean variables continuas y categóricas.

Un primer ejemplo es el trabajo de Vaid y col. (2020), que utiliza sobre las variables con no más de un 70 % de valores faltantes, un modelo XGBoost (Chen y col., 2015), que no requiere imputar los datos; es una solución a los datos faltantes similar a la de Li y col. (2021), que empleaba *Gradient Boosting Decision Trees* (GBDT). En el trabajo de Vaid y col. (2020) también se emplearon otros modelos sensibles a la falta de datos, para los cuales se realizó previamente una imputación con k-vecinos más cercanos (kNN) (Zhang, 2012), que puede considerarse como imputación multivariante; es un método también utilizado en el trabajo de Armañanzas y col. (2021), que igualmente mantenía solo variables con al menos un 30 % de valores disponibles. En este caso, como se realizó el estudio con una versión estática de los datos, usando los primeros datos conocidos, no hubo problema con la faceta multinivel de los datos. Cabe mencionar que no se indicó la función distancia empleada en el algoritmo kNN, información de interés para el conjunto de datos que utilizaron, puesto que integraba variables continuas y categóricas.

Un trabajo más es el de Zhu y col. (2020), que utilizó imputación múltiple con ecuaciones encadenadas (Van Buuren y col., 2011). Este método de imputación también se ha visto en muchos otros trabajos sobre el Covid-19 (Vepa y col., 2021; Bolt y col., 2022; Feng y col., 2021). Otra opción para la imputación multivariante es el uso de redes Bayesianas en conjunto con un algoritmo de esperanza-maximización (EM) (Ruggieri y col., 2020; Dempster y col., 1977). Aunque no se han encontrado trabajos que apliquen este procedimiento al ámbito del Covid-19, sí se ha probado su uso en conjuntos de datos clínicos (Rancoita y col., 2016).

Sin embargo, no todos los trabajos emplearon imputación multivariante. El trabajo de Ko y col. (2020), enfocado a predicción de mortalidad, imputó los valores faltantes utilizando la media aritmética, trabajando una vez más sobre un fragmento estático del conjunto de datos con los primeros datos conocidos del paciente.

2.2. Modelos predictivos

En esta sección se hace una revisión de trabajos del estado del arte que crean modelos predictivos en el contexto del Covid-19. Adicionalmente, se discute el uso de estos modelos para la extracción de conocimiento, así como los trabajos en contrafactuales

para interpretar las salidas de los modelos. Cabe destacar que Alderisi (2020) dedicó su tesis del Máster en Ciencia de Datos de la UPM a profundizar en el estado del arte del uso de herramientas de aprendizaje automático en el ámbito del Covid-19, incluyendo el desarrollo de modelos predictivos. Otro trabajo de relevancia general es el de Wynants y col. (2020), que hace una revisión y crítica de una gran cantidad de trabajos en modelos predictivos de Covid-19. Se destaca de sus resultados que en todos los trabajos que estudiaron, los modelos tenían sesgos en la predicción bien confirmados, bien dudosos, debido a la selección no representativa de los pacientes.

2.2.1. Redes Bayesianas

Las redes Bayesianas (Pearl, 1988; Koller y col., 2009) son esencialmente una tupla $(G, \boldsymbol{\theta})$, donde G es un grafo acíclico dirigido (DAG), cuyos nodos $X_1, X_2, ..., X_n$ representan n variables aleatorias. Cada arco (X_i, X_j) de una variable X_i a otra variable X_j indica que X_j tiene una relación de dependencia directa condicional con respecto a X_i; otra forma de decirlo es que X_i pertenece al conjunto de padres $\mathbf{Pa}_j$ de X_j, es decir, aquellos nodos con un arco hacia X_j. Los parámetros de la red $\boldsymbol{\theta}$ representan el conjunto de parámetros $(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_i, ..., \boldsymbol{\theta}_n)$ de la distribución condicional de cada variable X_i dados sus padres $\mathbf{Pa}_i$, $P(X_i|\mathbf{Pa}_i)$; o, si X_i no tiene padres, la distribución marginal de X_i, $P(X_i)$. Su principal utilidad es que proporcionan una factorización de la distribución conjunta, que se muestra en la Ecuación 2.1:

$$P(X_1, X_2, , ..., X_n) = P(X_1|\mathbf{Pa}_1) \cdot P(X_2|\mathbf{Pa}_2) \cdots P(X_n|\mathbf{Pa}_n) \tag{2.1}$$

Esta factorización es de especial utilidad porque normalmente no hay datos o recursos computacionales suficientes para calcular y almacenar todos los parámetros de la distribución conjunta real, pues puede aumentar exponencialmente en el número de variables. Las redes Bayesianas, una vez construidas, destacan porque permiten realizar inferencia: consultas de la probabilidad de que ocurra un evento, potencialmente dada una evidencia. Son aún más potentes si son del tipo causal.

Se dice que una red Bayesiana es causal cuando cada arco (X_i, X_j) indica que la variable X_i tiene una relación causal de carácter probabilístico con la variable X_j (Koller y col., 2009). Existen algoritmos para aprender la estructura y parámetros de una red Bayesiana a partir de datos observacionales, es decir, que no se han obtenido a partir de experimentos. Sin embargo, existen una serie de obstáculos que pueden llegar a ser insalvables (Glymour y col., 2019). El más frecuente es el sesgo de la selección, que se produce cuando la probabilidad de introducir una muestra es dependiente de alguno de sus valores. Esto puede llevar a extraer conclusiones erróneas a partir de los datos sobre la relación causal de una variable con respecto a otra. Otro muy frecuente es que se encuentre relación entre dos variables (*confounding variables*) donde

ninguna es causa de la otra, siendo el motivo de la relación que existe una variable oculta que es causa de ambas. Por ejemplo, tomar helado y tomarse vacaciones de verano son dos variables relacionadas, pero ninguna es causa de la otra; la causa común sería el calor.

En el contexto del Covid-19, Fenton (2020) explica cómo un sesgo en la selección creó la ilusión de que ser fumador reduce el riesgo de contraer la enfermedad, cuando el problema residía en que las muestras de fumadores con la enfermedad tenían menor probabilidad de ser incluidas en los datos que la probabilidad en sí de ser fumador y tener la enfermedad. Es por este motivo que frecuentemente las redes causales se obtienen a partir de datos experimentales, o con una mezcla con datos observacionales (Meganck y col., 2006) cuando se quiere sacar conclusiones extrapolables al ámbito global de las variables; en el resto de los casos aún pueden utilizarse como herramienta para buscar asociaciones, pero su uso para la extracción de relaciones causales difícilmente estaría justificado (Fenton y col., 2020).

Las barreras anteriores no han impedido el uso de redes Bayesianas en el contexto del Covid-19. Fenton y col. (2021) construyeron una red Bayesiana causal, utilizando una mezcla de datos experimentales, observacionales y conocimiento experto. En este caso se utilizaba como herramienta para diagnosticar la probabilidad de tener Covid-19 con mayor o menor severidad, así como para ayudar a prevenir los contagios.

En el contexto de la predicción de la mortalidad, es más complicado recolectar datos no sesgados, y los procedimientos experimentales naturalmente tienen barreras éticas. Esto no impidió que Vepa y col. (2021) construyeran una red Bayesiana causal enfocada a la predicción de mortalidad y búsqueda de factores de riesgo. En el trabajo se justifica en primer lugar que dada la dimensión del conjunto de datos, en este caso 44 variables, el riesgo de tener *confounding variables* es menor. También se argumenta que el hecho de tener tanto pacientes que fallecieron como supervivientes reduce el riesgo del sesgo de selección.

Las variables continuas se discretizaron utilizando el método multivariante no supervisado de Chen y col. (2017) y se utilizó *recursive feature elimination* (RFE) (Guyon y col., 2002) en conjunto con *random forests* (RF) (Breiman, 2001) para simultáneamente ordenar las variables de menor a mayor importancia y seleccionar el subconjunto que permitía dar mejores resultados para clasificación de mortalidad. Posteriormente se construyó una red Bayesiana a partir de los datos con algoritmos del estado del arte (Scutari, 2009) y se modificó la estructura G con la opinión de expertos para que fuera coherente con la realidad. La red se utilizó para calcular la probabilidad de muerte en presencia de factores de riesgo conocidos, con interés en comprobar si las variables ordinales tenían propiedades aditivas; es decir, que al incrementarse o reducirse su valor, incrementaban la probabilidad de fallecer de forma monótona e independientemente del valor del resto de variables. Cabe notar que se obtuvieron

mejores resultados en cuanto a predicción de mortalidad cuando se usó la red con estructura G corregida por expertos.

Sin embargo, cuando el objetivo es maximizar el rendimiento en la clasificación, y no buscar una factorización de la distribución conjunta más cercana a la realidad para la extracción de conocimiento, se pueden usar los clasificadores Bayesianos (BNC) (Friedman y col., 1997). Estos buscan obtener estructuras más simples, eficientes y con mayor rendimiento en clasificación, permitiendo a cambio que la factorización de la distribución conjunta pueda ser menos cercana a la realidad. El trabajo de Bielza y col. (2014) recoge los trabajos principales de BNC. Se categorizan como *augmented naive Bayes*, que son aquellos que no admiten que la variable clase tenga padres; los *unrestricted Bayesian network classifiers* (UBNC), que no imponen restricciones a la estructura de la red; y los *Bayesian multinets*, que admiten estructuras distintas de la red en función del valor de la clase. Sobre los UBNC, se puede añadir que en la clasificación únicamente necesitan el manto de Markov, un subconjunto de las variables predictoras tal que si se conocen, la clase es independiente del resto.

En el contexto del Covid-19, el trabajo de Sánchez-Montañés y col. (2020) explora el uso de distintos algoritmos de aprendizaje automático para predicción de mortalidad. Entre ellos, se encontraba el *tree augmented naive Bayes* (TAN) (Friedman y col., 1997), de la rama *augmented naive Bayes* ya mencionada. Las variables se discretizaron con intervalos utilizados por los médicos, y a pesar de la pérdida de granularidad asociada, resultó ser uno de los modelos con mejor capacidad predictiva, además de tener baja varianza en sus resultados. En este caso, se mostró la estructura del clasificador resultante, que permitía visualizar el comportamiento del modelo para la toma de decisiones.

Otro trabajo de naturaleza similar es el de Schirato y col. (2021), que compara varios modelos de aprendizaje automático para clasificación de mortalidad en Covid-19, entre ellos, *naive Bayes* (Minsky, 1961), un BNC de la rama *augmented naive Bayes*, y un UBNC. Ambos modelos funcionaron significativamente peor que el resto de modelos evaluados (árboles de decisión, kNN, máquinas de vectores de soporte, redes neuronales y RF) en cuanto a tasa de aciertos. Esto contrasta con el trabajo de Shanbehzadeh y col. (2021), donde también se comparan varios modelos para predicción de mortalidad, incluyendo *naive Bayes* y un UBNC, y en los que el UBNC obtuvo los mejores resultados en clasificación en cuanto a tasa de aciertos, superando a las redes neuronales y máquinas de vectores de soporte. Ninguno de los dos trabajos anteriores hizo uso de la estructura o capacidades de inferencia de las redes Bayesianas para extracción de conocimiento, a diferencia del de Fenton y col. (2021), que sí empleó la red construida con este propósito.

2.2.2. Otros modelos

En la literatura se han utilizado toda clase de modelos predictivos de la mortalidad en el contexto del Covid-19, como se puede ver en los trabajos comparativos ya citados de Sánchez-Montañés y col. (2020), Schirato y col. (2021) y Shanbehzadeh y col. (2021). En esta sección se revisa el rendimiento en clasificación, y el uso para extraer conocimiento de algunos de los modelos más utilizados en el ámbito del Covid-19.

Uno de los modelos más populares para clasificación en el ámbito médico es la regresión logística (Berkson, 1944). El motivo principal para su uso es su simpleza y facilidad de construcción e interpretación. Esencialmente modela la probabilidad de obtener un valor de la variable clase $Y = y$ dados unos valores $\boldsymbol{x} = (x_1, ..., x_n)$ de las predictoras $X_1, ..., X_n$ como se muestra en la Ecuación 2.2:

$$P(Y = y|\boldsymbol{x}) = 1/(1 + exp(-(\beta_0 + \beta_1 \cdot x_1 + \cdots + \beta_n \cdot x_n))) \tag{2.2}$$

El modelo en sí tiene una interpretación asociada en cada parámetro $\beta_i, i = 1, ..., n$, pues si $\beta_i > 0$, nos indica que valores de $X_i > 0$ aumentan la probabilidad $P(Y = y|\mathbf{x})$, y lo análogo para $\beta_i < 0$. También, $\beta_i \sim 0$ nos indica que la variable X_i tiene poca influencia en la distribución a posteriori $P(Y = y|\mathbf{x})$. La limitación del modelo reside en que no tiene en cuenta la relación entre las predictoras, lo que puede llevar a conclusiones erróneas en algunos casos. Esto puede llegar a paliarse extendiendo el modelo con variables auxiliares que modelen la interacción entre las predictoras, por ejemplo, mediante el producto de cada pareja (X_i, X_j).

En el contexto de predicción de mortalidad de pacientes de Covid-19, el trabajo de Armañanzas y col. (2021) consiguió buenos resultados de predicción usando una regresión logística. Pese a que el propio modelo ya proporciona información de este tipo, se decidió utilizar gráficos SHAP (Lundberg y col., 2017) para visualizar el efecto de las variables en la predicción del modelo, siendo SHAP un método independiente del modelo.

SHAP permite explicar de forma individual la contribución del valor de una predictora $x_i \in \mathbf{x}$ en la predicción de un modelo $\phi(\mathbf{x}) \in \mathbb{R}$, que podría ser una probabilidad a posteriori de una clase, o un valor real en una regresión. Suponiendo que $\boldsymbol{F} \subseteq \{X_1, ..., X_n\}$ es un subconjunto de las predictoras, se puede hacer la predicción $\phi(\mathbf{x}^{(\boldsymbol{F})})$, donde $x_i^{(\boldsymbol{F})} = x_i$ si $X_i \in \boldsymbol{F}$ y $x_i^{(\boldsymbol{F})} = \mathbb{E}[X_i]$ en otro caso. Definiendo $F(X_i)$ como todos los posibles subconjuntos $\boldsymbol{F}$ que verifican $X_i \in \boldsymbol{F}$, la contribución de $x_i \in \mathbf{x}$ a la predicción $\phi(\mathbf{x})$ se evalúa como se ve en la Ecuación 2.3:

$$\texttt{SHAP}(x_i) = \sum_{\boldsymbol{F} \in F(X_i)} \left[|\boldsymbol{F}| \cdot \binom{n}{|\boldsymbol{F}|} \right]^{-1} \left[\phi(\mathbf{x}^{(\boldsymbol{F})}) - \phi(\mathbf{x}^{(\boldsymbol{F} \setminus X_i)}) \right] \tag{2.3}$$

Es decir, es una suma ponderada de la diferencia de la predicción cuando se tiene X_i y cuando no se tiene X_i en los distintos posibles subconjuntos de predictoras. La ponderación se hace de tal forma que los pesos sean iguales para el mismo valor de $|\boldsymbol{F}|$, la suma de pesos correspondiente a mismos valores de $|\boldsymbol{F}|$ siempre sea igual para cualquier $|\boldsymbol{F}|$, y, para que todos los pesos sumen 1. Calculando el SHAP de cada $x_i \in \mathbf{x}$ para distintos $\mathbf{x}$ en un conjunto de datos, se puede generar una explicación a nivel global del modelo, observando para cada predictora X_i la distribución de sus valores SHAP.

También es frecuente el uso de los árboles de decisión. Esencialmente es un grafo con estructura de árbol, donde los nodos representan particiones de las predictoras, y las hojas representan la distribución observada en los datos que se ajustaban a dichas particiones (Quinlan, 1987). En este caso, es un modelo directamente interpretable, pues la visualización del árbol indica el criterio que sigue para generar una decisión, y permite observar en los nodos más cercanos a la raíz las variables consideradas como más relevantes. Huyut y col. (2022) usan un árbol de decisión para realizar diagnóstico y prognosis de Covid-19. Se demuestra mediante visualizaciones del árbol el funcionamiento del modelo. Un trabajo similar a este es el de Bernaola y col. (2022), que usaron un algoritmo evolutivo para la construcción del árbol de decisión. Se buscaba maximizar una suma ponderada de la especificidad y sensibilidad del modelo, con pesos 1 y 3 respectivamente.

Una extensión muy popular de los árboles de decisión es el uso de *ensembles* de árboles de decisión, siendo algunos de los más populares los ya mencionados RF y *XGBoost*. Aunque estos modelos destacan por su rendimiento en clasificación, no tienen interpretabilidad inherente, debido a que integran generalmente muchos árboles (Marchese Robinson y col., 2017). Sin embargo, existen algoritmos para extraer una estimación de la importancia de las predictoras en ambos casos (Menze y col., 2009). Por ello, cuando se usan estos modelos en el ámbito médico, donde la interpretabilidad es más importante, normalmente es para un proceso de selección de variables, o para obtener un *ranking* de la importancia de las predictoras, tal y como se muestra en Vepa y col. (2021). Otro ejemplo es el trabajo de Wang y col. (2020), que también se apoya en RF para estimar las variables más relevantes, sobre las que después se hacían estudios estadísticos más profundos.

También existen algoritmos generales para determinar la importancia de cada predictora en cualquier modelo predictivo, que demuestran su utilidad en modelos opacos como las redes neuronales profundas o los ya mencionados RF y *XGBoost*, como por ejemplo *permutation importance* de Altmann y col. (2010), que se usó en el trabajo de Zhu y col. (2020) con redes neuronales profundas para predicción de mortalidad de Covid-19. El algoritmo parte de un clasificador ϕ entrenado con un conjunto de datos $\mathcal{D}^{(train)}$. Su rendimiento se evalúa con una métrica M en un conjunto de validación

$\mathcal{D}^{(val)}$, obteniendo $m = M(\phi, \mathcal{D}^{(val)})$. Para obtener la importancia de cada predictora X_i, primero se calcula para $k = 1, ..., K$, $m_k = M(\phi, \mathcal{D}^{(val,i)})$, donde $\mathcal{D}^{(val,i)}$ es una versión de $\mathcal{D}^{(val)}$ donde las entradas de la variable X_i están desordenadas. La importancia de X_i viene dada por $m - \frac{1}{K}\sum_{k=1}^{K} m_k$.

2.2.3. Explicabilidad de las predicciones mediante contrafactuales

Hasta ahora, se ha discutido el uso de modelos predictivos de aprendizaje automático en el contexto de Covid-19, y cómo se puede interpretar el modelo para extraer conocimiento, posiblemente con técnicas independientes del modelo como SHAP. En esta sección se revisa el uso de explicaciones contrafactuales para interpretar salidas concretas de los modelos.

Las explicaciones contrafactuales se utilizan cuando la predicción de un modelo no es la esperada, y consisten en obtener una serie de ejemplos lo más similares posible a la entrada original para los que el modelo hubiera predicho la salida esperada. Al igual que con los métodos de extracción de variables más relevantes, existen algoritmos de obtención de explicaciones contrafactuales agnósticos al modelo, y específicos al modelo; aunque esta última categoría puede refinarse diferenciando los algoritmos que solo necesitan acceso a un gradiente en el modelo de los que no (Verma y col., 2020).

Normalmente, los algoritmos que acceden al modelo son los pensados para modelos basados en árboles, puesto que requieren un acceso a los nodos de los árboles para encontrar las explicaciones. Un ejemplo es el trabajo de Fernández y col. (2020), que utiliza RF. También se han desarrollado algoritmos para explicar la salida de redes neuronales enfocándose en el resaltado de sub-grafos, como se ve en el trabajo de Lucic y col. (2022).

Por otra parte, los algoritmos agnósticos al modelo, normalmente se basan en el uso de algoritmos de optimización para encontrar explicaciones contrafactuales que sean plausibles y cercanas a la entrada original; algunos de los trabajos más relevantes en este aspecto quedan recogidos en Molnar (2022). El trabajo de Wachter y col. (2017), propone una función de pérdida a minimizar, que se muestra en la Ecuación 2.4:

$$L(\mathbf{x'}) = \lambda \cdot d_1(f(\mathbf{x'}, y') + (1 - \lambda) \cdot d_2(\mathbf{x}, \mathbf{x'}) \tag{2.4}$$

$\mathbf{x}$ es la entrada original al modelo, $\mathbf{x'}$ es la explicación contrafactual candidata, y' es la salida deseada del contrafactual, $f(\mathbf{x'})$ es la salida del modelo para el contrafactual $\mathbf{x'}$, d_1 es la distancia Euclídea, d_2 es una distancia Manhattan, aplicada sobre x_i, x'_i previamente normalizadas con la de la desviación mediana absoluta de la variable $DMA(X_i)$; finalmente, $\lambda > 0$ pondera la importancia relativa de obtener explicaciones

cercanas a la entrada original **x**, y explicaciones cuya predicción es más cercana a la deseada y'. L se puede minimizar con casi cualquier algoritmo de optimización (Nelder y col., 1965; Kirkpatrick y col., 1983).

Dandl y col. (2020) proponen utilizar un algoritmo de optimización multi-objetivo (Deb y col., 2002), que minimiza la distancia $d(\mathbf{x}, \mathbf{x'})$, la distancia a la predicción deseada $d(f(\mathbf{x'}), y')$, el número de variables cuyo valor se ha modificado $\gamma(\mathbf{x}, \mathbf{x'})$, y maximiza la probabilidad de la explicación contrafactual $P(\mathbf{x'})$. Destaca con respecto al trabajo anterior el tener en cuenta la plausibilidad de la explicación.

En el contexto clínico, se pueden mencionar pocos trabajos de su uso. Tsirtsis y col. (2021) exploran el uso de explicaciones contrafactuales de cara a la intervención y toma de decisiones en distintos ámbitos, incluyendo el médico. Jia y col. (2021) demuestran su uso para justificar diagnósticos y predicciones en la medicina. Kovalev y col. (2021) proponen una metodología general con contrafactuales para explicar predicciones de supervivencia en modelos dinámicos.

El trabajo de Wang y col. (2021) usa explicaciones contrafactuales para integrar modelos opacos en el ámbito médico capaces de predecir la supervivencia en pacientes en la UCI por problemas cardiovasculares. No se han encontrado sin embargo trabajos análogos para pacientes de Covid-19.

Capítulo 3

Desarrollo

3.1. Preprocesamiento del conjunto de datos

En la Sección 2.1 ya se discutió la necesidad de preprocesar adecuadamente un conjunto de datos del ámbito clínico, así como las frecuentes dificultades asociadas al proceso. El conjunto de datos proporcionado por la FJD no es una excepción a la regla, y ha sido necesario aplicarle varias transformaciones antes de poder utilizarse con los métodos descritos en las secciones siguientes. En esta sección se explican los problemas encontrados en este proceso, así como las decisiones tomadas para remediarlos.

3.1.1. Unión de los datos nuevos y viejos

Como ya se mencionó en la introducción, el primer paso fue combinar el conjunto de datos del primer año del proyecto con el del segundo año. La estructura no era realmente la misma, pues los nombres de las variables se encontraban bajo una codificación diferente en el conjunto de datos nuevo, además de estar separado por hojas en un fichero Excel, donde había una hoja para los pacientes de cada uno de cuatro hospitales de la FJD, una hoja con la información en lenguaje natural de los pacientes vacunados y otra con la información en lenguaje natural de pacientes que sufrieron una neumonía bilateral. El conjunto de datos del año anterior en cambio, era un Excel de una sola hoja con los datos de los pacientes de los cuatro hospitales.

Por tanto, se concilió en un mismo documento en formato CSV los datos nuevos y viejos de los cuatro hospitales, usando la convención de nombres de los datos viejos, de la que se eliminaron algunos caracteres problemáticos como los espacios y tildes para el *software* que se utilizaría después; y en un fichero Excel aparte se guardó la información de vacunación en lenguaje natural de 6329 pacientes, que sería procesada más adelante. Las variables no pertenecientes a la intersección del

conjunto de variables de los datos viejos y nuevos fueron descartadas, aunque en este caso eran metadatos como comentarios de los pacientes. En las secciones siguientes se desarrollan los pasos seguidos para procesar estos dos conjuntos de datos.

3.1.2. Procesamiento de la información de vacunas

En los datos del segundo año se incluía información de vacunación de algunos de los pacientes en lenguaje natural. Esto era básicamente aquellos comentarios realizados por los enfermeros y/o médicos acerca del paciente que incluían la palabra clave "vacuna" y sus derivaciones. Mientras que una de las columnas contenía el comentario completo, otra de ellas contenía el fragmento del comentario con la palabra clave "vacuna" y los 30 caracteres posteriores incluyendo la palabra clave y los 30 anteriores sin incluir la palabra clave. Un ejemplo de los comentarios de un paciente vacunado es el siguiente:

> *"Odinofagia y fiebre de 2 días de evolución. No aparente contacto con paciente COVID-19. Vacunado: 2 dosis Pfizer"*

Con un fragmento correspondiente:

> *"...ntacto con paciente COVID-19. Vacunado: 2 dosis Pfizer..."*

A continuación se muestra un ejemplo del fragmento de comentario de un paciente no vacunado:

> *"...as. Covid julio 2021. No está vacunado. Refiere bultoma late..."*

Al ser lenguaje natural, son frecuentes las faltas ortográficas, cambio de mayúsculas y minúsculas, etc. Para facilitar la aplicación de las técnicas de procesamiento de la información, se transformaron ambas variables convirtiendo todos los caracteres a minúsculas y eliminando espacios sobrantes.

Como se quería evitar tener que etiquetar a mano todas las entradas, inicialmente se probó a etiquetar a mano 900 entradas para entrenar una red neuronal que clasificara el resto de las instancias. El objeto de probar esta metodología sería poder utilizarla posteriormente para la codificación de otros datos en lenguaje natural sin esfuerzo añadido.

Las redes neuronales son modelos de aprendizaje automático que utilizan el descenso por gradiente para minimizar el error de su función de predicción. Toman como entrada un vector $\mathbf{x}$ de números reales y producen, para un problema de clasificación binaria como este, una estimación de la probabilidad $P(Y = y|\mathbf{x})$, donde Y sería la variable aleatoria "¿Está vacunado?" e y por ejemplo la etiqueta "sí". Cuando se quiere clasificar un texto, hay que hacer una vectorización de algún tipo para que la red entienda la entrada, es decir, hacer una transformación $\tau(T) = \mathbf{x}$, siendo T una variable aleatoria que representa los comentarios en lenguaje natural.

En el caso de las redes neuronales, esta transformación consiste normalmente en hacer una tokenización, y añadir una capa de *embedding* (Mikolov y col., 2013) al principio de la red. La tokenización se realiza de forma previa a la fase de entrenamiento de la red neuronal y consiste esencialmente en asociar un número entero a cada posible palabra del texto (añadiendo un identificador para palabras desconocidas) para así tener una transformación de $T = t$ a un vector $\boldsymbol{u}$ de números enteros.

La capa de *embedding* busca realizar una transformación de cada posible término $u_i \in \boldsymbol{u}$ a un vector $\boldsymbol{v}^{(u_i)}$ de números reales tal que si dos términos u_i, u_j son similares, el ángulo entre $\boldsymbol{v}^{(u_i)}$ y $\boldsymbol{v}^{(u_j)}$ sea pequeño, mientras que si son opuestos, el ángulo sería mayor; siendo el tamaño E de cada vector un parámetro de la capa a establecer. Es decir, tras aplicar la tokenización y *embedding* a un texto t, se tendría una matriz bidimensional de números reales $\boldsymbol{M}$, donde la fila i-ésima correspondería al *embedding* $\boldsymbol{v}^{(u_i)}$. Normalmente se reduce su dimensionalidad para obtener la entrada deseada $\mathbf{x}$ mediante el promedio o máximo de cada fila, e.g. $x_i = \max(\boldsymbol{M_i})$, proceso conocido como *pooling* global. La optimización del *embedding* puede realizarse como parte del proceso de entrenamiento de la red con el descenso por gradiente.

Sin embargo, al proceder de esta forma, no se hace uso eficientemente de las palabras. Para solventar el problema, se puede emplear una capa recurrente R, que transforma una matriz $\boldsymbol{M}$, donde la fila $\boldsymbol{M_i}$ representa un vector de predictoras en el instante i-ésimo, a un vector $\mathbf{x}$ de tamaño dependiente de los parámetros de R. Se denomina recurrente porque R se descompone en tantas capas R_i como filas tenga $\boldsymbol{M}$, donde cada R_i utiliza como entrada la fila $\boldsymbol{M_i}$ y el resultado de aplicar R_{i-1} a $\boldsymbol{M_{i-1}}$. R produce como salida la correspondiente a la última fila de $\boldsymbol{M}$, que es el mencionado $\mathbf{x}$. Opcionalmente, para reducir los recursos computacionales necesarios, se puede reducir parcialmente el tamaño de la entrada de R (en este caso, de $\boldsymbol{M}$) aplicando un *pooling* local de cada p elementos a lo largo de las filas de $\boldsymbol{M}$, típicamente mediante el máximo o el promedio y siendo p un parámetro a establecer. Se decidió utilizar esta metodología, empleando como R una *Gated Recurrent Neural Network* (GRU) (Bahdanau y col., 2014).

Para intentar aplicar este modelo, se partió del tutorial de *TensorFlow*[1]. Se separó del conjunto de entrenamiento (las 900 muestras etiquetadas a mano), un 15 % como conjunto de test, y del 85 % restante, se utilizó un 30 % como conjunto de validación para búsqueda de hiperparámetros y el resto como conjunto de entrenamiento. El mejor resultado en el conjunto de validación se obtuvo con una tasa de aciertos del 89 %, y al aplicar el modelo final sobre el conjunto de test, una tasa de aciertos del 80 %.

La Figura 3.1 muestra la matriz de confusión correspondiente a la clasificación del conjunto de test.

[1] `https://www.tensorflow.org/tutorials/keras/text_classification`

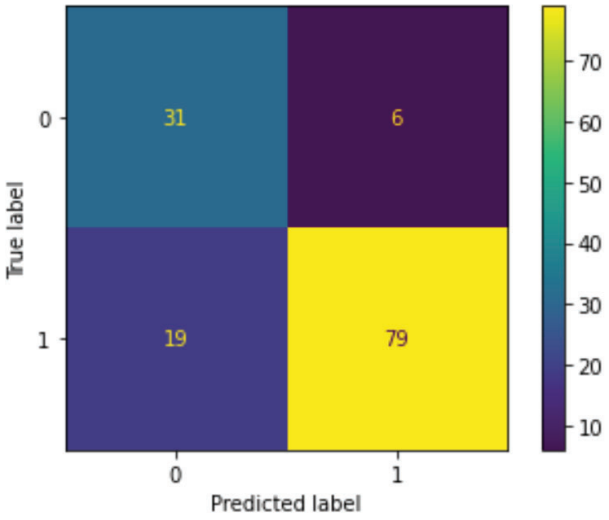


Figura 3.1: Matriz de confusión de la clasificación sobre el conjunto de test de información de vacunación en lenguaje natural usando una red neuronal recurrente con *embeddings* y GRU. 1=Sí, 0=No.

Como el impacto de la vacunación era una de las líneas a investigar, se decidió que una tasa de aciertos del 80 % no era suficientemente buena, con lo que se cambió la metodología a una manual, pero también más robusta. Simplemente se etiquetaba una entrada como "vacunado" cuando aparecía cualquiera de un conjunto de palabras clave A, pero ninguna de un conjunto de palabras clave B. Los conjuntos A y B se estimaron a mano de la siguiente forma:

1. Establecer $A = \emptyset$, $B = \emptyset$.
2. Mientras no se cumpla la condición de parada:
   - 2.1. Desordenar el conjunto de registros de vacunación en lenguaje natural.
   - 2.2. Estimar a mano si añadir palabras clave a A o B.
   - 2.3. Etiquetar el resto de los datos.
   - 2.4. Estimar a mano la tasa de aciertos usando 100 entradas con 50 de cada tipo (vacunado y no vacunado).
   - 2.5. Repetir el procedimiento si la tasa de aciertos no es suficiente.

El proceso se finalizó cuando tras dos iteraciones seguidas sin cambiar A o B, solamente se fallaba en la etiquetación de 1 de las 50 de entradas de cada tipo.

Los conjuntos resultantes fueron:

A = {"1 dosis", "2 dosis", "3 dosis", "una dosis", "dos dosis", "tres dosis", "1ª dosis", "2ª dosis", "3ª dosis", "1º dosis", "2º dosis", "3º dosis", "rimera dosis", "segunda dosis", "tercera dosis", "doble dosis", "triple dosis", "janss", "janse", "pfiz", "pfaiz", "astra", "moderna", "covi", "covd", "coronavirus", "sars", "conviviente", "pauta vacunal

completa", "doble pauta", "triple pauta", "pauta de vacuna completa", "pauta completa de vacuna", "pauta de vacunación completa", "pauta completa de vacunación", "vacunación completa", "vacunacion completa", "x3", "x2", "x 3", "x 2"}

B = {"sin vac", "ninguna vac", "no vac", "ni vac", "tampoco vac", "falta vac", "niega vac", "no se ha vac", "no está vac", "no esta vac", "tampoco se ha vac", "tampoco está vac", "tampoco esta vac", "ni se ha vac", "ni está vac", "ni esta vac", "niega que se ha vac", "niega estar vac", "niega que está vac", "niega que esta vac", "sin cumplimiento", "pollinex", "pendiente vac", "pendiente de vac"}

3.1.3. Filtrado de pacientes y registros

Tras integrar la información de vacunación ya codificada al conjunto de datos principal, para facilitar las tareas de limpieza posteriores, el siguiente paso aplicado fue el filtrado de pacientes y registros que no eran de interés para el estudio, ya sea por semántica, o por ser de carácter ruidoso. Los pasos del filtrado se enumeran a continuación:

1. Inicialmente se tienen 446555 registros de 83685 pacientes, con 532 variables.
2. Se seleccionaron los datos de pacientes ingresados, con lo que quedaron 239150 registros de 13321 pacientes.
3. De acuerdo con la mayoría de trabajos de la literatura, se seleccionaron solamente los datos de pacientes mayores de 18 años, quedando 237952 registros de 13058 pacientes.
4. Se observó también que había muestras ruidosas de 112 pacientes donde la fecha de alta estaba intercambiada con la de ingreso, tras descartarlas, quedaron datos de 12980 pacientes con un total de 233908 muestras. Dado que no todas las muestras de los pacientes mencionados eran ruidosas, no se perdió el total de los 112 pacientes.
5. Se filtraron también los datos de pacientes que no habían tenido ninguna PCR positiva con lo que quedaron 125307 muestras de 6845 pacientes.
6. Para este trabajo había que escoger la ventana de tiempo a estudiar en el ingreso de los pacientes. Se acordó con el equipo médico realizar un estudio de tipo pronóstico temprano usando las primeras muestras conocidas de los pacientes, más concretamente, utilizando muestras entre dos días anteriores a la fecha de ingreso del paciente y siete días después. De acuerdo con el equipo médico, normalmente las muestras previas al ingreso del paciente se deben a que el paciente volvió a su casa tras unas pruebas iniciales y de allí de vuelta al hospital tras empeorar. Tras realizar este filtrado, se tienen datos de 6293 pacientes y 54935 muestras.

7. Finalmente, para los pacientes con varios ingresos, se seleccionaron las muestras del último ingreso, quedando 50163 muestras de los 6293 pacientes.

3.1.4. Asignación de la ola correspondiente a cada ingreso

En este punto, tocaba asignar a cada ingreso la ola pandémica correspondiente, puesto que los estudios posteriores se harían teniendo en cuenta esta información. Para no seleccionar de forma arbitraria los intervalos de fecha correspondientes, se utilizaron datos de casos confirmados de la Comunidad Autónoma de Madrid[2], promediando semanalmente para suavizar la evolución. Es decir, se trabajó con un conjunto de datos donde se indicaba para cada día el promedio de casos confirmados de los últimos 7 días.

Para transformar el conjunto de datos en los intervalos asociados a cada ola, se empleó un algoritmo de clústering probabilístico usando mixturas de distribuciones Gaussianas con la implementación de *scikit-learn*. El algoritmo es capaz de producir dado un conjunto de N observaciones $\mathbf{x}^{(1)}, ..., \mathbf{x}^{(N)}$, $\mathbf{x}^{(i)} \in \mathbb{R}$ y un número m de clústeres, una mixtura de m distribuciones Gaussianas $\mathcal{N}_1, ..., \mathcal{N}_m$, tal que cada punto $\mathbf{x}^{(i)}$ está asociado a cada distribución anterior con un peso que resulta ser su probabilidad de pertenencia a dicha conformante de la mixtura.

Aplicado a este problema, simplemente se toma la entrada $\mathbf{x}$ como una variable aleatoria unidimensional que toma valores en el rango de días estudiado y su función de densidad es directamente proporcional al número de casos confirmados en ese día; y para extraer una muestra de dicha distribución, se toma para cada día del rango de estudio una fracción del número de casos confirmados ese día. Sabiendo que $m = 6$ pues en el rango de estudio hay seis olas pandémicas, se utiliza el algoritmo para asociar a cada día la distribución Gaussiana más probable, que en este caso se corresponde con una ola de la pandemia. El resultado de aplicar este proceso se muestra en la Figura 3.2. Intuitivamente, el algoritmo ha estimado cada "campana" correspondiente a una ola como una distribución Gaussiana.

Después de asignar a cada paciente su ola pandémica correspondiente utilizando los resultados anteriores, la distribución de los pacientes a lo largo de las olas, así como la de la mortalidad y estancia en UCI/UCIR queda representada en la Tabla 3.1.

Tras observar que el número de datos de las olas 3, 4 y 5 era bastante escaso, se decidió agruparlas, pues ambas correspondían de forma mayoritaria a la variante Delta. De esta forma, la ola 1 corresponde mayoritariamente a la variante original de Wuhan, la ola 2 a la variante Alfa, las olas 3, 4 y 5 a la Delta, y la ola 6 a Ómicron. En este momento, se decidió descartar la continuación del estudio de la estancia en UCI/UCIR, debido a que se disponía de muy pocos datos de pacientes ingresados,

[2] https://www.comunidad.madrid/servicios/salud/coronavirus

Patients	Variable	Yes (%)	No (%)
Wave 1			
1980	ICU	166 (08.38)	1814 (91.62)
1980	Death	612 (30.91)	1368 (69.09)
Wave 2			
1965	ICU	149 (07.58)	1816 (92.42)
1965	Death	319 (16.23)	1646 (83.77)
Wave 3			
695	ICU	36 (05.18)	659 (94.82)
695	Death	95 (13.67)	600 (86.33)
Wave 4			
197	ICU	22 (11.17)	175 (88.83)
197	Death	10 (05.08)	187 (94.92)
Wave 5			
459	ICU	54 (11.76)	405 (88.24)
459	Death	49 (10.68)	410 (89.32)
Wave 3,4,5			
1351	ICU	112 (08.29)	1239 (91.71)
1351	Death	154 (11.40)	1197 (88.60)
Wave 6			
997	ICU	64 (06.42)	933 (93.58)
997	Death	126 (12.64)	871 (87.36)
Wave 1,2,3,4,5,6			
6293	ICU	491 (07.80)	5802 (92.20)
6293	Death	1211 (19.24)	5082 (80.76)

Tabla 3.1: Distribución de las variables clase por olas.

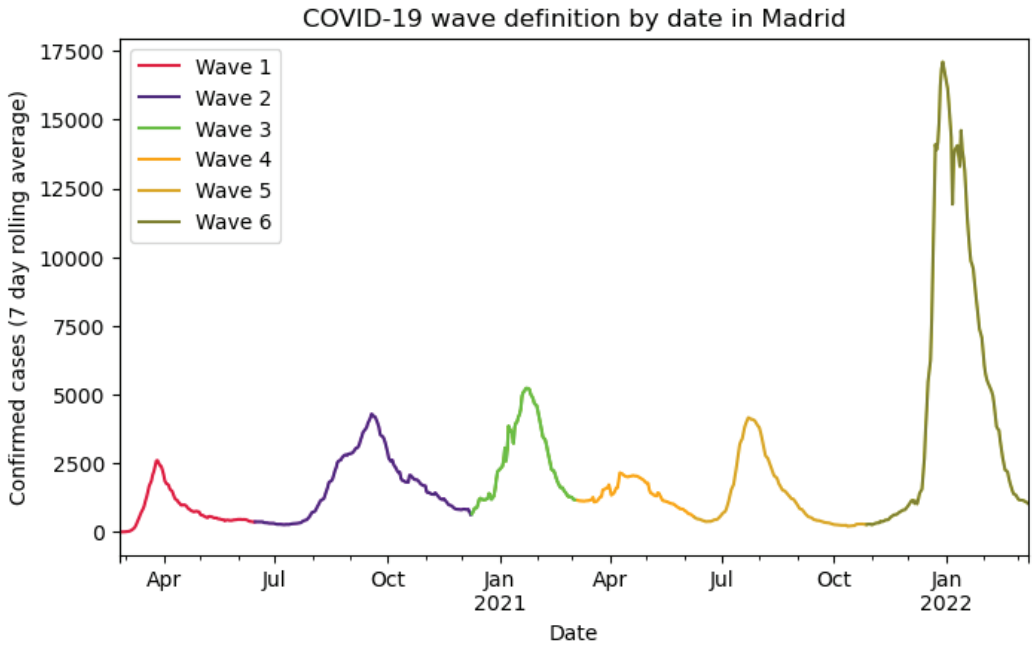


Figura 3.2: Asociación de olas por fechas en la Comunidad de Madrid usando clústering probabilístico.

particularmente en la ola 6, y se consideró que las medidas a tomar para paliar este problema entorpecerían y dificultarían el estudio de la mortalidad.

3.1.5. Limpieza, imputación y selección de variables preliminar

Tras realizar un filtrado del conjunto de datos por filas, procede comenzar a procesar las variables del conjunto de datos y tratar los valores faltantes y ruidosos de algunas de ellas.

En primer lugar, se observó que algunas de las variables continuas tenían distintos niveles de granularidad, con valores nominales como > 3.5 y < 7.4, pero también continuos como 2.8. Este problema se ignoró porque todas las variables afectadas o iban a ser descartadas más adelante por no disponer de valores suficientes, o porque tenían menos de un 5 % de valores discretos, por lo que se estimó que convertirlos a su contrapartida numérica no afectaría demasiado al tratamiento de la variable.

La excepción se dio con las variables correspondientes al filtrado glomerular estimado, que tenía dos variables asociadas en el conjunto de datos: la medida con la ecuación MDRD4, y la medida con la ecuación CKD-EPI. En este caso, la variable con la ecuación MDRD4 tenía varios valores marcados con > 60, y se observó que a no ser que tomara este valor, la variable con CKD-EPI tenía siempre un valor ausente. En este caso, los valores superiores a 60 son los normales, con lo que probablemente, una vez estimado un valor por debajo de lo normal con la ecuación MDRD4, ya no se utiliza la ecuación CKD-EPI; es decir, se utilizaría esta segunda medida para asegurar

que el paciente no tiene valores por debajo de lo normal del filtrado glomerular. Por tanto, siendo que la estimación con CKD-EPI se considera una medición más precisa de la función renal de los pacientes (Burballa y col., 2018), se decidió usar los valores con MDRD4 para imputar los faltantes de la estimación con CKD-EPI y eliminar la variable con la estimación con MDRD4. En este caso, la imputación consiste simplemente en asignar el valor de la estimación con MDRD4 donde falte en la estimación con CKD-EPI.

Tras comprobarlo con el equipo médico, se determinó que la imputación de valores ausentes en comorbilidades era mejor hacerla con el caso negativo (no fumador, no diabetes, etc.). La información de vacunación prácticamente estaba solo disponible para las olas 5 y 6. En dichas olas, se observó que había muchos valores faltantes (véase Tabla 3.2), del 28 % y 37 % respectivamente. Siendo una variable de especial interés de estudio, se imputaron estas variables con una etiqueta especial de "no disponible", para minimizar la posibilidad de una extracción posterior de conclusiones erróneas tras aplicar un algoritmo de imputación. Con esto finalizó una primera imputación de los datos utilizando la propia semántica de las variables.

Wave	Patients	Vaccinated	Not vaccinated	Not available
1	1980	0	0	1980
2	1965	0	0	1965
3	695	0	0	695
4	197	0	0	197
5	459	229	103	127
3,4,5	1351	229	103	1019
6	997	459	165	373

Tabla 3.2: Información de vacunación por olas.

En este punto se hizo un primer filtrado de las variables a incluir en el estudio utilizando como criterio el mínimo de valores disponibles a lo largo del ingreso del paciente. Se decidió tras hablar con un miembro del equipo médico establecer el límite de valores faltantes al 30 %. Tras aplicar este filtro, se eliminaron un total de 437 variables. También se eliminaron metadatos sin interés y otras variables como la información de medicamentos que ya se había decidido que no se tendrían en cuenta para el estudio. Del conjunto de variables resultante, 77 eran variables candidatas para la predicción de las posibles variables clase, incluyendo la mortalidad; estas variables se pueden consultar en el Apéndice A.

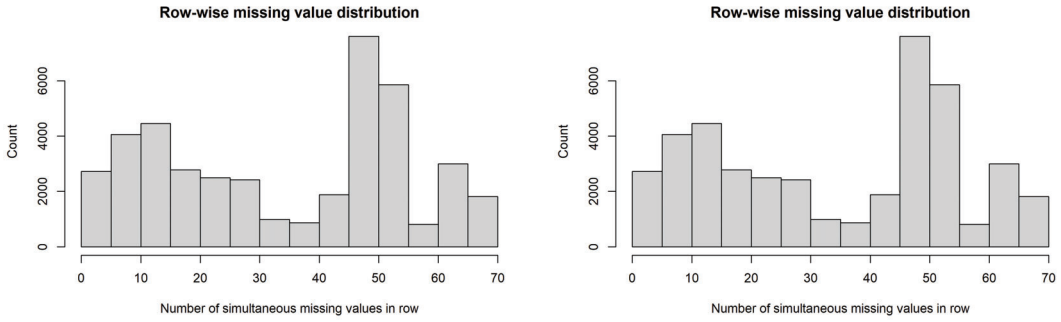
Después, se buscaron datos ruidosos de variables que podrían afectar al resto de pasos. En primer lugar, se observó que algunas variables no eran consistentes a lo largo del ingreso, por ejemplo pacientes marcados inicialmente como no fallecidos y después como fallecidos; en estos casos se asignó el valor más coherente. También, se observó que algunos grupos de los registros estaban marcados con la misma fecha de emisión de prueba de laboratorio para el mismo paciente. En estos casos se seleccionó la primera muestra conocida por orden de aparición en el conjunto de datos,

utilizando los valores de las siguientes muestras en caso de tener algún valor faltante. Este último paso redujo el número de registros a 41768.

Finalmente, se observó que algunas variables tenían puntualmente valores absurdos, como por ejemplo, valores del índice de masa corporal de 10000, que nunca ha alcanzado un ser humano. Para detectar estos casos y tratarlos como valores faltantes, se dibujó un histograma de cada variable numérica para detectar este tipo de casos en el resto de variables continuas y definir unos intervalos de corte para el descarte de anomalías.

3.1.6. Imputación múltiple de los valores faltantes

El paso final del preprocesamiento de los datos es la imputación de los datos faltantes que no puede realizarse con la propia semántica de las variables. Para la imputación de estos datos, en la literatura se suelen utilizar algoritmos de imputación múltiple, pero antes de proceder, se dibujaron algunos gráficos para entender mejor la distribución de los valores faltantes. Se observó que muchas de las entradas tenían varios valores faltantes simultáneos en el mismo registro, como se puede ver en la Figura 3.3a. Sin embargo, también se comprobó que frecuentemente los registros de los pacientes se enviaban en tandas de cerca de 24 horas, como se puede ver en la Figura 3.4. Por tanto, se decidió paliar el problema de los valores faltantes simultáneos agrupando las muestras de los pacientes por día transcurrido desde el ingreso utilizando la mediana, quedando 20042 registros. El resultado queda reflejado en la Figura 3.3b.



(a) Original.

(b) Agrupando por día desde ingreso con mediana.

Figura 3.3: Distribución del número de valores faltantes simultáneo.

Una vez agrupados los datos, el paso final es la imputación del resto de valores faltantes, en este caso pertenecientes a 67 variables continuas (las vitales y variables de laboratorio en la Tabla A.1, edad (aunque no tenía valores faltantes) e índice de masa corporal (IMC)), pues el resto de variables discretas se habían imputado por su semántica. Para seleccionar un método de imputación, se separó en este punto un conjunto de test del 10 % del tamaño del conjunto de datos, y del 90 % resultante, se fraccionó un 20 % como conjunto de validación y un 80 % como conjunto de

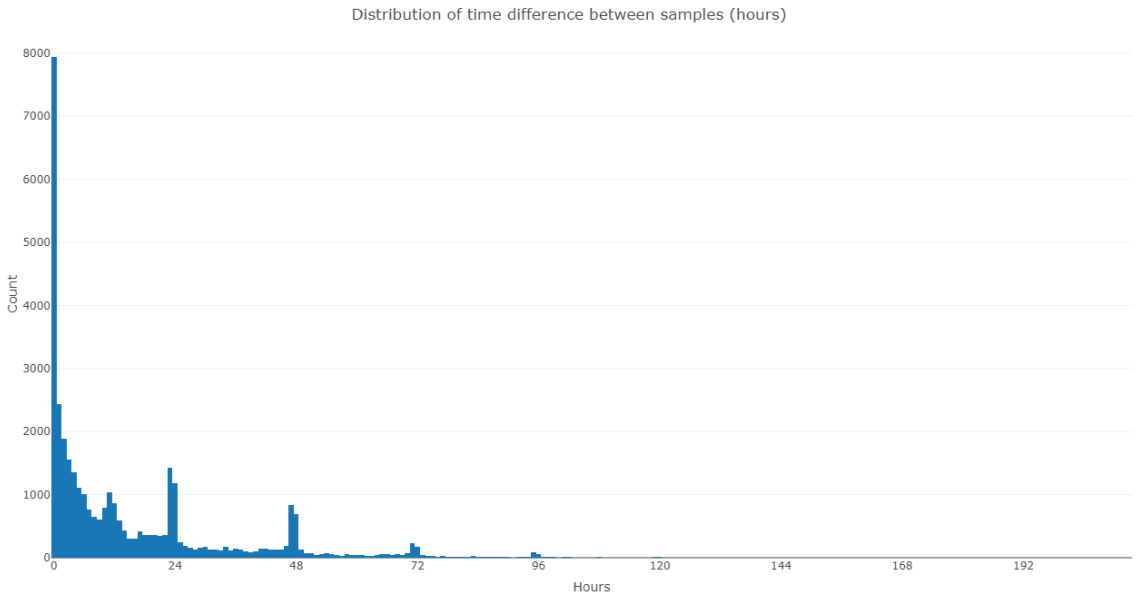


Figura 3.4: Distribución de tiempo transcurrido en horas entre registros consecutivos.

entrenamiento. El conjunto de validación se utilizó para seleccionar un método de imputación apropiado y posteriormente para comparar el rendimiento de distintos modelos predictivos. El conjunto de test se utilizaría únicamente para evaluar el rendimiento final del mejor modelo predictivo para cada ola en cuanto al conjunto de validación.

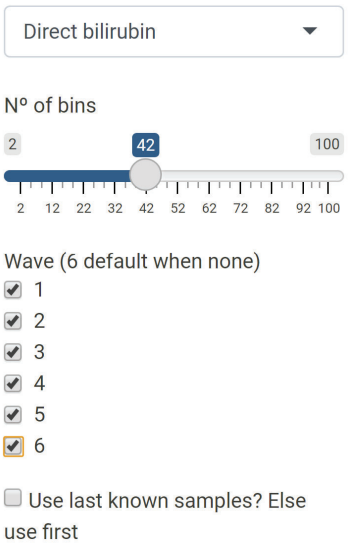
Como algoritmo de imputación, se utilizó la imputación múltiple de *scikit-learn*, basada en MICE (Van Buuren y col., 2011). Este procedimiento es iterativo, y parte de una asignación de valores plausibles a cada valor faltante utilizando la media o mediana, con lo que se aprende para cada variable un estimador que asignará los valores para la iteración siguiente. Se utilizó el conjunto de entrenamiento y la primera muestra conocida de cada paciente para construir el modelo de imputación, y se utilizó para imputar los datos de entrenamiento, validación y test; posteriormente, se repitió el proceso con la última muestra conocida del paciente (hasta 7 días después del ingreso). Al finalizar, se realizó una interpolación lineal dentro de los datos de cada paciente para rellenar los valores faltantes intermedios. Este proceso se realizó de manera independiente para cada grupo de olas de la pandemia.

La calidad de la imputación de las variables se valoró en cuanto a que la imputación de los datos pareciera plausible dentro de la distribución conocida de cada variable y a que no se apreciaran diferencias significativas en el rendimiento de distintos clasificadores en cuanto a su comportamiento en el conjunto de entrenamiento y en el de validación, cuya información no se utiliza para entrenar el modelo de imputación.

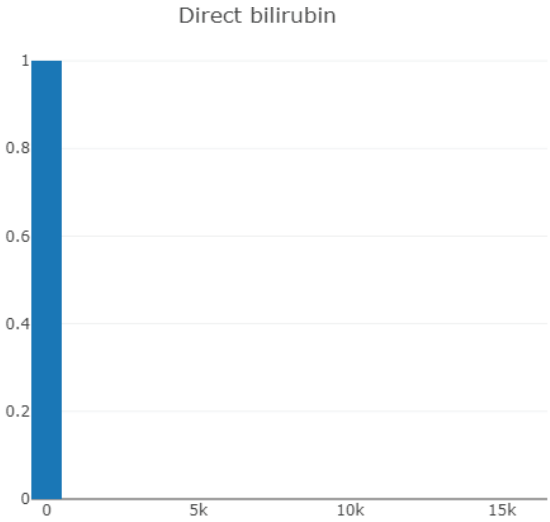
Con estos dos criterios en mente, se escogió el uso del estimador de k-vecinos más cercanos con $k = 5$ y usando como predictoras las dos predictoras más correladas

con la variable a imputar. Siendo un algoritmo basado en distancias, las variables se estandarizaron previamente usando la desviación mediana absoluta. Este método producía resultados plausibles y los mejores en cuanto a la métrica de área bajo la curva ROC (AUC) al predecir la mortalidad en la ola 6 usando los primeros datos conocidos de cada paciente, tanto en el conjunto de validación, como en una validación cruzada con 10 particiones sobre el conjunto de entrenamiento. Por ello, se escogió como método de imputación para el resto de grupos de olas.

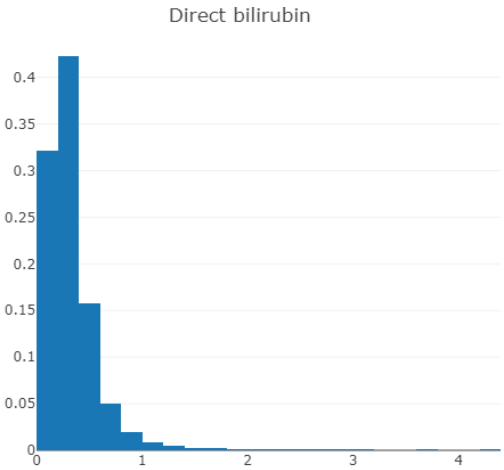
Como en todo el proceso de limpieza se modifican muchas variables y es difícil controlar la introducción de errores, se construyó un *dashboard* interactivo que permite seleccionar una variable del conjunto de datos final y conjunto de olas mediante el formulario de la Figura 3.5a para comprobar las sucesivas transformaciones aplicadas. Se muestra un ejemplo con la bilirrubina directa, que como se ve en la Figura 3.5b, parece tener valores ruidosos extremos; tras su filtrado, queda una distribución como la que se ve en la Figura 3.5c. Tras la agrupación de las muestras por día desde el ingreso usando la mediana, la distribución queda como se ve en la Figura 3.5d. Finalmente, la Figura 3.5e representa la distribución final de la variable tras la imputación, mientras que se visualiza en la Figura 3.5f una comparativa de la distribución de la imputación de los valores faltantes y la distribución de los valores conocidos. Como se ve, los valores de la imputación, aunque más sesgados a los valores intermedios, parecen plausibles; esto se confirmó con un miembro del equipo médico.



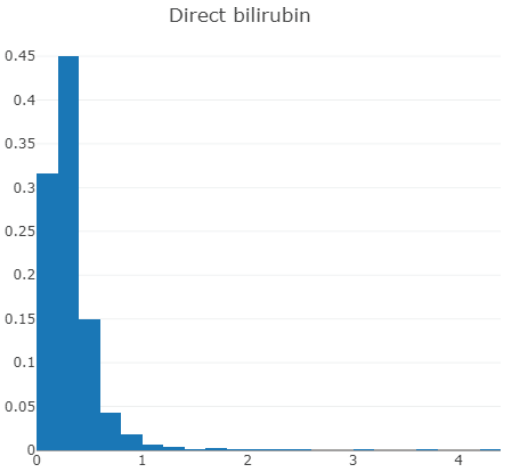
(a) Formulario de selección.



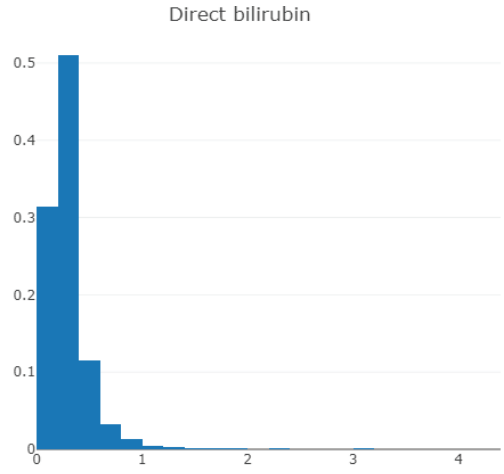
(b) Distribución original.



(c) Tras filtrar ruido.



(d) Tras agrupar por día con la mediana.



(e) Tras aplicar imputación múltiple.

(f) Comparativa de la imputación.

Figura 3.5: Transformación de la bilirrubina directa (primera muestra de cada paciente).

3.2. Importancia y selección de variables por olas

3.2.1. Importancia y eliminación recursiva de variables

Tras realizar las operaciones de preprocesamiento mencionadas se cuenta con un total de 77 predictoras candidatas (véanse en la Tabla A.1). Es muy frecuente que muchas de ellas sean redundantes, además de que 77 variables dificultan la construcción de modelos y su interpretabilidad. Por este motivo es conveniente realizar un proceso de selección de variables para obtener un subconjunto con las variables más relevantes y no redundantes. Para seleccionar variables, una metodología frecuente es *recursive feature elimination* (RFE) (Guyon y col., 2002).

El paquete *caret* en R (Kuhn, 2008) incluye una implementación de dicho algoritmo. También, se incluye en la documentación de dicho paquete un pseudocódigo del funcionamiento del algoritmo, adaptado en el Algoritmo 3.1.

Algoritmo 3.1 Pseudocódigo del algoritmo RFE.

Entrada:

- $\mathcal{D}$. Conjunto de datos con n predictoras.
- I. Función para estimar la importancia de una predictora X dado un modelo ϕ.
- M. Función para estimar el rendimiento en clasificación de un modelo ϕ dados unos datos $\mathcal{D}$.

Procedimiento:

1. Para cada remuestreo $\mathcal{D}^{(i)}$, $i = 1, ..., m$ de los datos **hacer**:
   - 1.1. Particionar la muestra en un conjunto de entrenamiento $\mathcal{D}^{(i)}_{train}$ y test $\mathcal{D}^{(i)}_{test}$.
   - 1.2. Construir modelo ϕ_0 con $\mathcal{D}^{(i)}_{train}$ usando todas las predictoras.
   - 1.3. Estimar rendimiento de ϕ_0 en $\mathcal{D}^{(i)}_{test}$ con $M^{(i)}_0 = M(\phi_0, \mathcal{D}^{(i)}_{test})$.
   - 1.4. Estimar la importancia de cada predictora X_k como $I^{(i)}_{0,k} = I(\phi_0, X_k)$, $k = 1, ..., n$
   - 1.5. Para tamaños $S_j \in \boldsymbol{S} = \{S_1, S_2, ..., S_N\}$ de subconjuntos de predictoras **hacer**:
     - 1.5.1. Escoger el subconjunto $\boldsymbol{F}_j$ de las S_j predictoras con mayor valor de $I^{(i)}_{j-1,k} = I(\phi_{j-1}, X_k)$, $k = 1, ..., n$.
     - 1.5.2. Construir modelo ϕ_j con $\mathcal{D}^{(i)}_{train}$ usando $\boldsymbol{F}_j$.
     - 1.5.3. Estimar rendimiento de ϕ_j en $\mathcal{D}^{(i)}_{test}$ con $M^{(i)}_j = M(\phi_j, \mathcal{D}^{(i)}_{test})$.
     - 1.5.4. Estimar la importancia de cada predictora X_k como $I^{(i)}_{j,k} = I(\phi_j, X_k)$, $k = 1, ..., n$. Otra opción es utilizar la estimación inicial: $I^{(i)}_{j,k} \leftarrow I^{(i)}_{0,k}$.
2. Con $M_j = \sum_{i=1}^{m} M^{(i)}_j / m$, $j = 1, ..., N$, estimar el tamaño S_j apropiado.
3. Con $I_{j,k} = \sum_{i=1}^{m} I^{(i)}_{j,k} / m$, $j = 1, ..., N$, $k = 1, ..., n$, estimar $\boldsymbol{F}$ como las S_j predictoras con mayor valor de $I_{j,k}$, $k = 1, ..., n$.
4. **Devolver** $\boldsymbol{F}$ como selección de variables.

El algoritmo determina para un problema de clasificación de una variable Y con unas predictoras $X_1, ..., X_n$, el subconjunto óptimo $\boldsymbol{F} \subseteq \{X_1, ..., X_n\}$ de predictoras para la clasificación de Y. Necesita para ello un medidor de importancia de variables I, tal que $I(X_i) = I_i$ es un número real que estima la importancia de la variable para la clasificación. Normalmente I es una función que extrae la importancia de las variables de un clasificador ϕ con un algoritmo específico para este, que se notaría

como $I(\phi, X_i)$.

De esta forma, el proceso comienza construyendo un clasificador ϕ_0 utilizando todas las predictoras $\boldsymbol{F}_0 = \{X_1, ..., X_n\}$, y se estima la importancia de cada una con un algoritmo I, así como el rendimiento de ϕ_0 en un conjunto de validación utilizando alguna métrica M, como por ejemplo la tasa de aciertos, usando dichas predictoras. Después, se repite el proceso para distintos conjuntos de variables $\boldsymbol{F}_j \subset \boldsymbol{F}_{j-1}, j = 1, ...N$, donde cada $\boldsymbol{F}_j$ se corresponde con las S_j variables más importantes de acuerdo con la estimación $I(\phi_{j-1}, X_k)$, $k = 1, ..., n$. $\boldsymbol{S} = \{S_1, ..., S_N\}$ es un hiperparámetro del algoritmo RFE, que denota los distintos tamaños de los subconjuntos $\boldsymbol{F}_j$ a evaluar. Todo este proceso se repite varias veces, siguiendo algún esquema de remuestreo como *bootstrap* para estimar con menor sesgo el valor de la métrica M, así como la importancia de cada variable para cada S_j, mediante el promedio.

Por ejemplo, en este caso, podría utilizarse como ϕ una regresión logística con el conjunto total de variables, y la función I proporcionaría el valor absoluto del peso de cada predictora X_i en la regresión como estimación de su importancia. Podría establecerse $\boldsymbol{S} = \{30, 15, 10\}$ para probar los subconjuntos de 30, 15 y 10 variables con mayor importancia dada la estimación anterior, y seleccionarse el subconjunto con mejor rendimiento en cuanto a M.

Basándose en el concepto de importancia aquí explicado, en la siguiente sección se explica su uso para construir un *dashboard* interactivo con la importancia de las distintas variables para la predicción de la mortalidad, así como el resultado de un proceso de RFE usando dichas medidas de importancia en nuestro conjunto de datos.

3.2.2. Visualización de la importancia y selección de variables con un *dashboard* interactivo

Para evaluar la evolución del impacto de las variables a lo largo de las distintas agrupaciones de olas de la pandemia (por fenotipo: 1 (Wuhan), 2 (Alfa), 3, 4 y 5 (Delta) y 6 (Ómicron)), se decidió que una buena forma era construir un *dashboard* interactivo que permitiera visualizar fácilmente la importancia de las distintas variables en cada ola de la pandemia. También se determinó que sería interesante poder ajustarlo en función de si la muestra era más temprana o más tardía; en este caso trabajando con la primera muestra conocida de los pacientes dentro de la primera semana del ingreso, con muestras de cualquier momento de la semana (de cada paciente, utilizar una muestra aleatoria de cualquiera de las de su primera semana ingresado), o con las últimas muestras conocidas de la semana. Esto permitiría estudiar la evolución de la importancia de las variables a medida que avanza la estancia del paciente.

Finalmente, como para algunos de los estudios posteriores se iba a requerir el uso de una discretización de las variables con intervalos de referencia proporcionados

por el equipo médico, se aprovechó para incluir en el *dashboard* una estimación de la importancia de las variables en función de si se utilizaban los datos originales o los discretizados, pudiendo así comprobar la pérdida de información asociada a la discretización y cómo cambia la prioridad de cada variable para la clasificación. La discretización se corresponde con los intervalos de referencia de la Tabla A.1. La edad se discretizó con los cortes $< 65, < 75, >= 75$ por sugerencia de un miembro del equipo médico, y el índice de masa corporal con $< 18.5, < 25, < 30, >= 30$, intervalos correspondientes a la delgadez, normalidad, sobrepeso y obesidad, de acuerdo con la *World Health Organization*[3].

Para poder construir un *dashboard* de este tipo, se necesita información de la importancia de las variables de cada posible combinación, e.g. importancia de las variables para las olas 3,4,5, primera muestra conocida, y uso de los datos discretizados. Esta estimación se realizó con 3 metodologías diferentes para lograr mayor robustez, todas implementadas en el paquete *caret*.

La primera metodología fue con *random forests* (RF) (Breiman, 2001), un clasificador de alto rendimiento en clasificación basado en conjuntos de árboles de decisión, que permite estimar la importancia de cada variable a partir de la reducción de impureza asociada a dicha variable en los distintos árboles que conforman el clasificador. La segunda metodología fue con *Generalized Boosting Models* (GBM) (Friedman, 2001), otro clasificador basado en grupos de árboles. Este emplea descenso por gradiente para reducir el error de clasificación, y permite promediar la importancia de cada variable como la reducción de error que le es atribuible en cada árbol en el proceso de entrenamiento.

Las metodologías anteriores estiman la importancia de cada variable de forma relativa al resto, puesto que el clasificador tiene en cuenta relaciones entre ellas. Se añadió una tercera metodología más simple que solo tiene en cuenta la importancia individual de cada variable X_i. El funcionamiento consiste en construir dados varios posibles valores x_i del dominio de X_i, un clasificador que estima la clase positiva para cualquier $x_i' > x_i$, y la clase negativa en caso contrario. En el caso de las variables categóricas, puede hacerse la predicción de la clase positiva cuando $x_i' = x_i$ y de la negativa en caso contrario. Utilizando estos distintos puntos x_i, y la sensibilidad y especificidad del clasificador asociado a cada punto, se construye una curva ROC, y el área bajo la misma es la estimación univariante de la importancia de la variable.

Para cada metodología y muestra de los datos (discretizada o no), se estimó la importancia de cada variable. Los hiperparámetros de RF y GBM se estimaron utilizando una búsqueda con muestreo de cuadrícula de la mejor configuración en cuanto a AUC en la predicción de mortalidad en una validación cruzada repetida con 3 re-

[3] https://www.who.int/europe/news-room/fact-sheets/item/a-healthy-lifestyle---who-recommendations

peticiones y 10 particiones. Los resultados se almacenaron en un fichero CSV para utilizarse posteriormente en la generación del *dashboard*.

Aprovechando que se disponía de estos tres algoritmos para la estimación de la importancia de variables, de los cuales RF y GBM se corresponden con un tipo de clasificador real, se hizo también un proceso de RFE para cada ola y tipo de datos (discretizados / no discretizados) para la construcción posterior de modelos y también para intentar comprender mejor qué variables pueden llegar a considerarse importantes (si son elegidas por ambos modelos por ejemplo). Se evaluaron subconjuntos de entre 10 y 30 variables usando la primera muestra conocida de los pacientes (población que se utilizaría en los estudios posteriores), y se seleccionó el subconjunto con mejor resultado promedio en la métrica AUC (sin considerar el conjunto de todas las variables por ser demasiado grande) en una validación cruzada de 10 particiones y 3 repeticiones. Véase el Apéndice B para consultar los resultados de este proceso.

Ayudándose del paquete *flexdashboard* (Iannone y col., 2018) de R, se desarrolló el *dashboard* con toda la información descrita en los párrafos anteriores. El *dashboard* consta de dos páginas, una para visualizar la importancia de variables, y la otra para consultar la evolución y resultados del proceso de RFE.

En la página para visualizar la importancia de las variables, se tiene un panel lateral (Figura 3.6a) que actúa como formulario para generar un gráfico bajo distintas condiciones de los datos. El formulario incluye:

1. Un selector del número de variables a mostrar, pues normalmente no se van a querer consultar todas las variables simultáneamente, sino por ejemplo, el top 20.

2. Un *checkbox* para indicar si se quiere utilizar la versión discretizada de los datos o no.

3. Unos *checkboxes* para indicar los grupos de olas a considerar en la generación del gráfico.

4. Otros *checkboxes* para indicar cuáles de las metodologías utilizadas para estimar la importancia de cada variable se tendrán en cuenta en la generación del gráfico.

5. Un último conjunto de *checkboxes* para establecer si se consideran las primeras muestras de cada paciente, las últimas, o una muestra aleatoria de cualquier momento, todo en la primera semana del ingreso (admitiendo también muestras de hasta 2 días anteriores).

6. Finalmente, se elige el tipo de formulario a generar de acuerdo con las opciones escogidas. Todos fundamentalmente son histogramas apilados, donde el tama-

ño de cada barra se corresponde con la importancia calculada correspondiente según las condiciones especificadas en el formulario, y dentro de cada barra se muestra el *ranking* correspondiente a dicho valor de importancia. Hay tres tipos:

6.1. Formulario comparativo por ola. Se muestra una comparación por olas de la importancia de las distintas variables. En caso de haber varios valores seleccionados en el apartado de metodología y tiempo de la muestra, se agregan los valores de la importancia correspondientes mediante el promedio. Un ejemplo de este tipo de gráfico se muestra en la Figura 3.7.

6.2. Formulario comparativo por metodología. Se muestra una comparación para las metodologías escogidas de la importancia de las variables. Si hay valores múltiples de otras entradas del formulario, como por ejemplo, una selección de varias olas, se agregan los valores de la importancia correspondientes mediante el promedio. En la Figura 3.8 se muestra un ejemplo de este tipo de gráfico.

6.3. Formulario comparativo por muestra del paciente. Este gráfico es el análogo a los dos anteriores, pero haciendo una comparativa de la importancia de las variables cuando se usan las primeras, últimas o una selección aleatoria de las muestras de cada paciente. Se muestra un ejemplo de este tipo de gráfico en la Figura 3.9.

En la página para visualizar el proceso de RFE, se tiene un panel lateral (Figura 3.6b) que actúa como formulario para generar un gráfico bajo distintas condiciones de los datos. El formulario incluye:

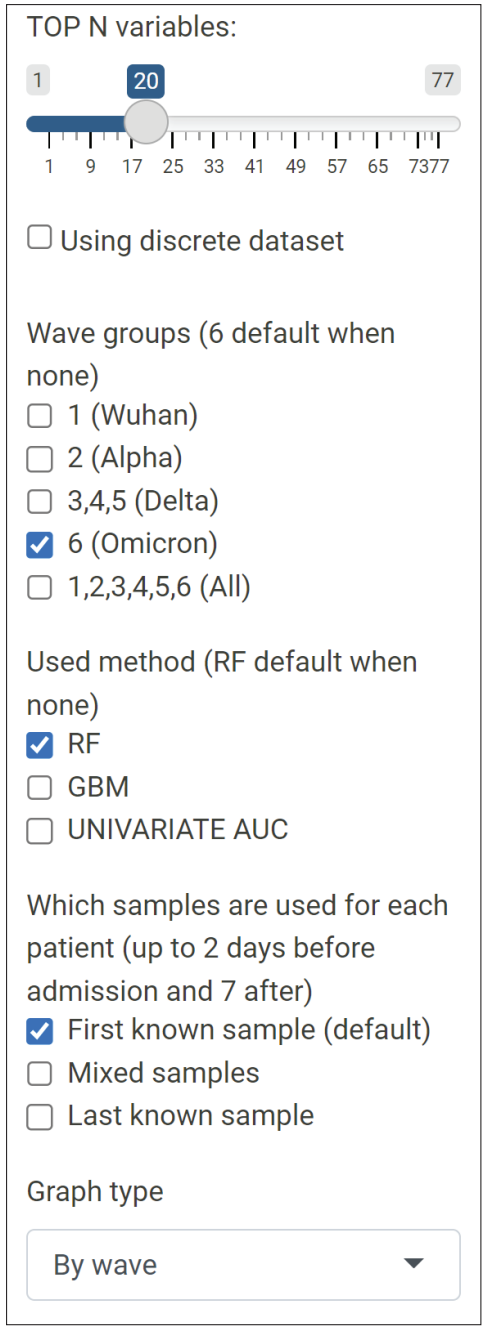
1. Un selector de grupo de ola, permitiendo elegir entre la ola "1", "2", "3, 4 y 5" o "6", asociadas a las distintas variantes.

2. Un grupo de *checkboxes* para indicar los clasificadores utilizados en el proceso de RFE a considerar, en este caso, RF o GBM.

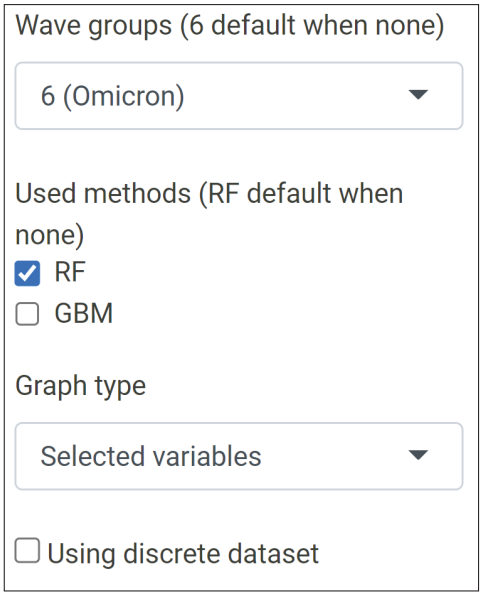
3. Un selector para indicar el tipo de gráfico a generar. Hay dos tipos:

   3.1. Tabla de selección. Se muestra una tabla con la unión y la intersección del conjunto de variables seleccionadas para los clasificadores indicados. Esto permite ver la selección individual de RF y GBM, pero también la unión e intersección de ambas al seleccionar ambas opciones. Un ejemplo de este gráfico se muestra en la Figura 3.10a.

   3.2. Gráfico de evolución. Se muestra una gráfica con la evaluación del AUC para los clasificadores seleccionados (curvas diferentes) en los distintos tamaños de subconjuntos de variables probados. Se indica con una flecha el tamaño final del subconjunto de variables escogido, que es el que tuvo



(a) Formulario para página de importancia de variables.



(b) Formulario para página de RFE.

Figura 3.6: Formularios para la generación de gráficos del *dashboard* interactivo.

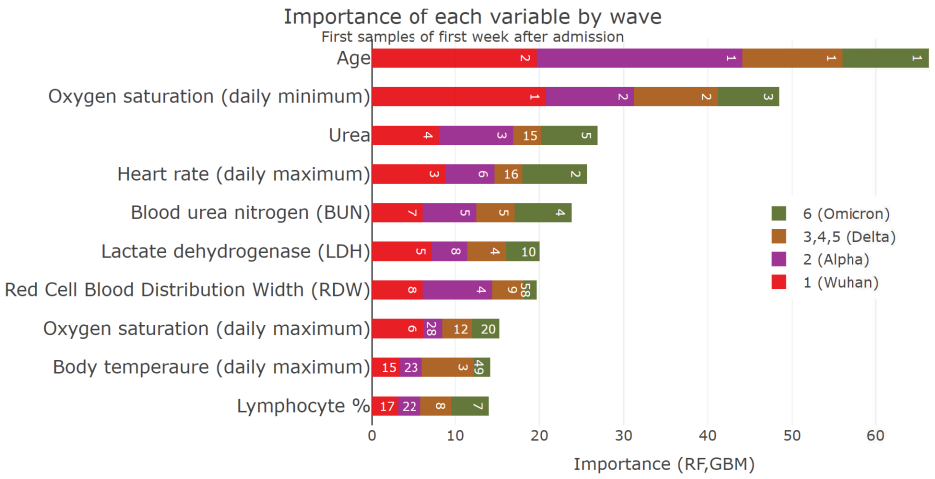


Figura 3.7: Gráfico comparativo de la importancia de las variables (top 10) de todos los grupos de olas usando RF y GBM con la primera muestra de cada paciente.

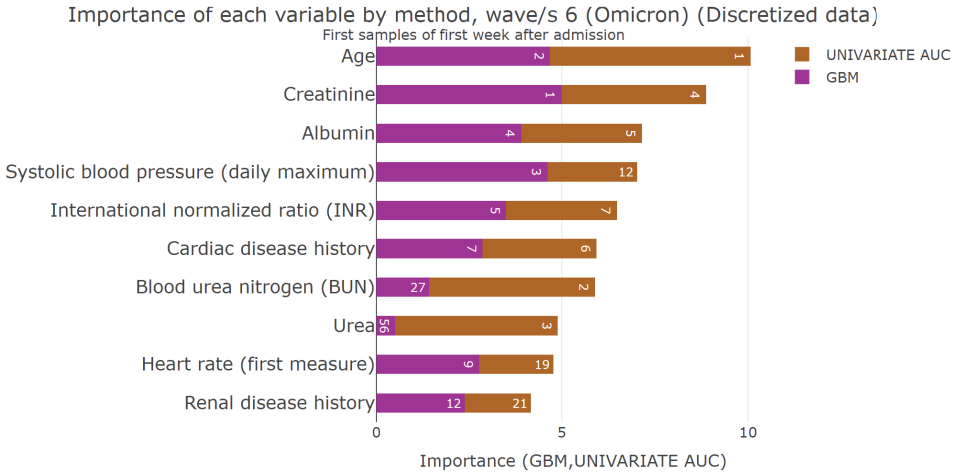


Figura 3.8: Gráfico comparativo de la importancia de las distintas variables según GBM y el método con AUC para la ola 6, con la primera muestra de los pacientes en la versión discretizada de los datos.

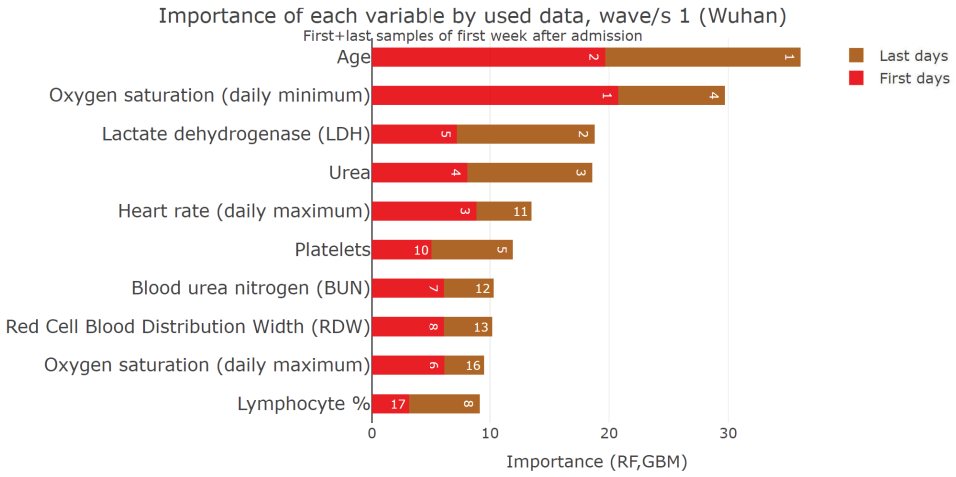


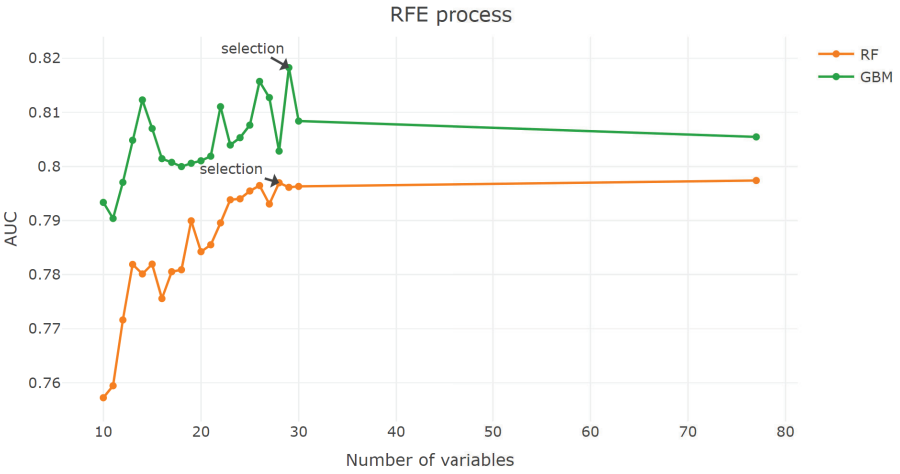
Figura 3.9: Gráfico comparativo de la importancia de variables usando RF y GBM para la ola 1 cuando se utiliza la primera y la última muestra de cada paciente.

mejor AUC dentro de los tamaños candidatos. En la Figura 3.10b se ve un gráfico de este tipo.

4. Un *checkbox* para indicar si se quiere trabajar con la selección realizada sobre una versión discretizada de los datos.

Selection (41). Union	Selection (15). Intersection
Age	Age
Systolic blood pressure (daily maximum)	Systolic blood pressure (daily maximum)
Oxygen saturation (daily maximum)	Creatinine
Segmented neutrophils %	International normalized ratio (INR)
Creatinine	Monocytes %
International normalized ratio (INR)	Heart rate (first measure)
Monocytes %	Lymphocyte count
Heart rate (first measure)	Red Cell Blood Distribution Width (RDW)
Lymphocyte count	Blood urea nitrogen (BUN)
Estimated glomerular filtration rate (eGFR) ckd-epi	Albumin
Eosinophil %	Cardiac disease history
Heart rate (daily maximum)	Body temperaure (daily maximum)
Red Cell Blood Distribution Width (RDW)	Oxygen saturation (first measure)
Oxygen saturation (daily minimum)	Mean corpuscular hemoglobin concentration (MCHC)
Blood urea nitrogen (BUN)	
Prothrombin time (PT)	
Prothrombin Time (Quick)	

(a) Fragmento de la tabla generada para la ola 6 al utilizar la versión discretizada de los datos.



(b) Evolución del proceso de RFE con RF y GBM según la métrica AUC.

Figura 3.10: Formularios para la generación de gráficos del *dashboard* interactivo.

3.3. Redes Bayesianas discretas para variantes Delta y Ómicron

Con el propósito de poder hacer un estudio de la interacción entre variables para las variantes Ómicron y Delta, se decidió construir una red Bayesiana discreta para cada una de ellas usando para la discretización los intervalos de referencia proporcionados por el equipo médico. En la Sección 2.2.1 se encuentra una explicación del

funcionamiento general de este modelo.

Para aprender una red Bayesiana $(G, \boldsymbol{\theta})$, primero hay que aprender la estructura G. Existen para ello numerosos algoritmos, normalmente clasificados como los basados en pruebas de independencia condicional, los de métrica+búsqueda, y una mezcla de ambos tipos, es decir, los híbridos. Los del primer tipo estiman la estructura algorítmicamente a partir de pruebas de independencia condicional. Una prueba de independencia condicional $I(X, Y|Z)$ indica si la variable aleatoria X es independiente de la variable Y cuando se conoce Z, donde X, Y y Z podrían ser vectores de variables aleatorias. En cambio, los de métrica+búsqueda utilizan como entrada una función de puntuación de la estructura $\mathcal{S}(G)$, que se optimiza realizando una búsqueda sobre el espacio de posibles grafos acíclicos dirigidos (posiblemente restringido) utilizando algoritmos como la búsqueda voraz, búsqueda tabú (Glover, 1986), etc.

Una vez aprendida la estructura G, se aprenden los parámetros de la red $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_n)$, donde los parámetros $\boldsymbol{\theta}_i$ correspondientes a cada variable X_i del modelo modelan la distribución de X_i dados sus padres $P(X_i|\boldsymbol{Pa}_i)$. En el caso de una red discreta, se representarían como una tabla de probabilidad condicional, donde cada posible valor de $X_i = x_i$ y configuración de valores de sus padres $\boldsymbol{Pa}_{i1} = \boldsymbol{pa}_{i1}, ..., \boldsymbol{Pa}_{im} = \boldsymbol{pa}_{im}$, dan lugar a cada probabilidad condicional $P(X_i = x_i|\boldsymbol{Pa}_{i1} = \boldsymbol{pa}_{i1}, ..., \boldsymbol{Pa}_{im} = \boldsymbol{pa}_{im})$. Utilizando los datos, estos parámetros se pueden estimar con el método de máxima verosimilitud, o bien añadiendo muestras imaginarias para evitar probabilidades condicionales nulas dadas configuraciones de los padres no vistas en los datos, que se conoce como estimación Bayesiana de los parámetros.

Para ambas olas, se incluyó una serie de variables de interés general para los médicos, ya sea por definir perfil de los pacientes (sexo, edad, historial médico), o por ser útiles en determinar el bienestar del paciente (vitales), así como distintas analíticas de laboratorio. Después, con el ánimo de descubrir el impacto en la mortalidad de otras variables no tenidas en cuenta en el listado superior, se incluyeron las 10 variables más relevantes según el *ranking* de importancia producido por el método *filter* con AUC descrito en la Sección 3.2. Adicionalmente, además de la mortalidad, se incluyó como variable clase la duración del ingreso del paciente, por la que también mostró interés el equipo médico; esta se discretizó con 5 intervalos de igual frecuencia. En la Tabla 3.3 se muestra la selección de variables descrita.

Como algoritmo de aprendizaje se empleó uno basado en métrica+búsqueda utilizando la implementación del paquete *bnlearn* en R (Scutari, 2009) de búsqueda tabú. Aunque el propósito de estas redes es comprobar la interacción entre las variables, se consideró que la variable de mortalidad tenía mayor interés. Por ello, la selección de hiperparámetros de la búsqueda se realizó teniendo en cuenta la capacidad predictiva de la red en cuanto al área bajo la curva ROC (AUC) con respecto a dicha variable de mortalidad. El método utilizado para encontrar los mejores hiperparámetros fue

Variables for Delta and Omicron variants	
Name	**Category**
Stay duration	**Outcome**
Death	
Vaccinated (>=1 dose)	**Profile**
BMI	
Age	
Sex	
Smoker	**Comorbilities**
Cardiac disease history	
Lung disease history	
Diabetic	
Arterial hypertension disease history	
Systolic blood pressure (daily maximum)	**Vitals**
Diastolic blood pressure (daily maximum)	
Body temperaure (daily maximum)	
Oxygen saturation (daily minimum)	
Heart rate (daily maximum)	
Ferritin	**Laboratory tests**
C-reactive protein	
D-Dimer	
Lymphocyte count	
Lymphocyte %	
Lactate dehydrogenase (LDH)	
Partial pressure of oxygen (Blood gas test)	
Partial pressure of CO2 (Blood gas test)	

(a) Selección común para variantes Delta y Ómicron.

Delta variant additional variables	
Name	**Category**
Urea	**Laboratory tests**
Hemoglobin	
Hematocrit	
Albumin	
Red blood cells	
Red Cell Blood Distribution Width (RDW)	
Blood urea nitrogen (BUN)	
Omicron variant additional variables	
Name	**Category**
Creatinine	**Laboratory tests**
Urea	
Blood urea nitrogen (BUN)	
Albumin	
International normalized ratio (INR)	
Red Cell Blood Distribution Width (RDW)	
Prothrombin time (PT)	

(b) Selección específica según variante.

Tabla 3.3: Selección de variables para redes Bayesianas discretas.

una búsqueda con muestreo de cuadrícula, que probó cada una de las combinatorias de los siguientes valores de hiperparámetros:

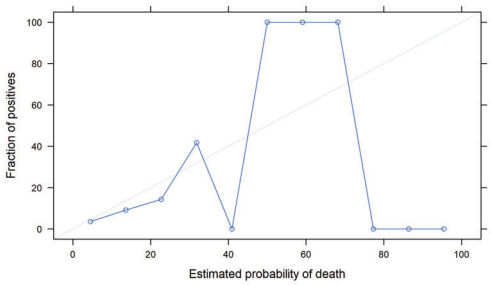
- Como *scores* de la estructura de la red, se consideraron *"modified Bayesian Dirichlet equivalent"* (mBDe) de Cooper y col. (2013) y *"factorized normalized maximum likelihood"* (FNML) de Silander y col. (2008).
- Valores de tamaño de la lista tabú de 5, 50 y 100.
- Utilizar o no un orden parcial de los nodos (que condiciona la dirección de los arcos), donde se tienen primero el sexo y la edad del paciente, en segundo lugar el historial médico del paciente, después las analíticas de laboratorio, seguido de las vitales, y finalmente las posibles variables clase.

No se estableció un límite en el número de padres, y se determinó como condición de parada el transcurso de 10 iteraciones del algoritmo sin mejorar el *score*.

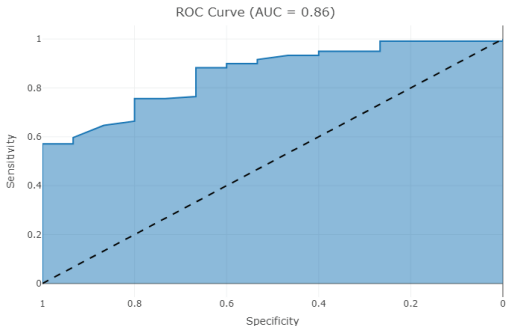
La puntuación dada a cada configuración de hiperparámetros se estimó con una validación cruzada repetida de 7 particiones y 4 repeticiones del AUC en predicción de la mortalidad; se prefería un múltiplo de 7 para facilitar la computación paralela utilizando un procesador de 7 núcleos. Estas particiones se realizaron sobre la combinación del conjunto de validación y entrenamiento que se mencionó en la Sección 3.1.6, utilizando los primeros datos conocidos del paciente. Después se usó el conjunto de test para comprobar el rendimiento final con la selección de hiperparámetros

estimada.

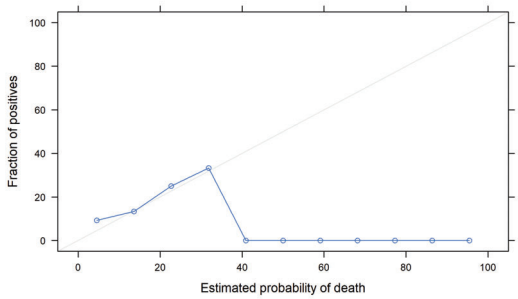
La mejor configuración se obtuvo en ambos casos sin emplear el orden parcial de los nodos, *score* mBDe y tamaño de lista tabú 100. Se estimó en la validación cruzada un AUC de 0.78 con desviación estándar 0.05 para la variante Delta y un AUC de 0.71 con desviación estándar 0.05 para la variante Ómicron. Sobre el conjunto de test, se obtuvo respectivamente un AUC de 0.86 y 0.69 para las variantes Delta y Ómicron. La Figura 3.11 muestra la curva ROC y calibración de ambas redes sobre el mencionado conjunto de test. Las Tablas 3.4 y 3.5 muestran los resultados de cada configuración de hiperparámetros probada en la validación cruzada sobre el conjunto de entrenamiento en cuanto a AUC.



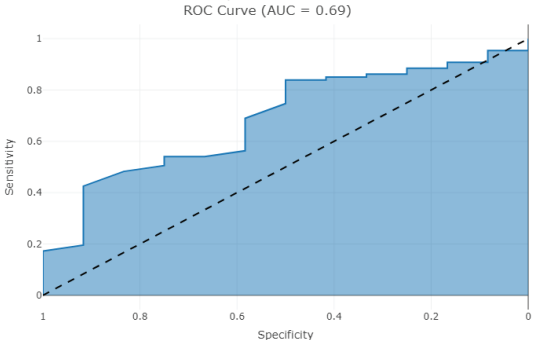
(a) Curva de calibración en conjunto de test (variante Delta).



(b) Curva ROC en conjunto de test (variante Delta).



(c) Curva de calibración en conjunto de test (variante Ómicron).



(d) Curva ROC en conjunto de test (variante Ómicron).

Figura 3.11: Capacidad predictiva de la mortalidad con redes Bayesianas discretas para variantes Delta y Ómicron.

Partial node ordering	Score	Tabu list size	Max iterations w/o improvement	AUC	AUC std. dev.
No	mBDe	50	10.00	0.78	0.04
No	mBDe	100	10.00	0.78	0.04
No	FNML	50	10.00	0.77	0.04
No	FNML	100	10.00	0.77	0.04
Yes	mBDe	50	10.00	0.74	0.04
Yes	mBDe	100	10.00	0.74	0.04
Yes	FNML	50	10.00	0.75	0.04
Yes	FNML	100	10.00	0.75	0.04

Tabla 3.4: Búsqueda de hiperparámetros para red de Delta en datos de entrenamiento y validación, con la media y desviación típica del AUC en predicción de mortalidad en una validación cruzada repetida.

Las Figuras C.1 y C.2 en el Apéndice C muestran la estructura completa obtenida

Partial node ordering	Score	Tabu list size	Max iterations w/o improvement	AUC	AUC std. dev.
No	mBDe	50	10.00	0.71	0.05
No	mBDe	100	10.00	0.71	0.05
No	FNML	50	10.00	0.68	0.05
No	FNML	100	10.00	0.69	0.05
Yes	mBDe	50	10.00	0.68	0.05
Yes	mBDe	100	10.00	0.68	0.05
Yes	FNML	50	10.00	0.69	0.06
Yes	FNML	100	10.00	0.69	0.06

Tabla 3.5: Búsqueda de hiperparámetros para red de Ómicron en datos de entrenamiento y validación, con la media y desviación típica del AUC en predicción de mortalidad en una validación cruzada repetida.

para las variantes Delta y Ómicron respectivamente, con las probabilidades a priori de cada variable. Para visualizar y realizar inferencia de forma interactiva con las redes Bayesianas discretas se ha utilizado la herramienta *software* GeNIe (Druzdzel, 1999).

Inmediatamente se pudieron observar algunos detalles en el cambio de distribución a priori de algunas variables. Por ejemplo, en cuanto al perfil de los pacientes, en Delta el $56\,\%$ de los pacientes era de sexo masculino y el $58\,\%$ de edad superior a 65 años, mientras que en Ómicron estos porcentajes son del $50\,\%$ y $65\,\%$ respectivamente, es decir, menos hombres y una población de mayor edad. También observamos que la probabilidad de tener valores de inflamación por encima de lo normal en el Dímero-D aumentó del $65\,\%$ al $89\,\%$ de una ola a la siguiente. Algo similar ocurre con los linfocitos, cuyo valor relativo toma valores por debajo de lo normal (linfopenia) con una probabilidad del $65\,\%$ en Delta y con una probabilidad del $80\,\%$ en Ómicron. Cabe mencionar que estas dos últimas variables se utilizan para evaluar el nivel de inflamación.

En cuanto a la mortalidad, en las Figuras 3.12 y 3.13 se muestra el manto Markov (variables tales que, conocidos sus valores, la mortalidad es independiente del valor del resto de variables, como se explicó en la sección 2.2.1) de las variantes Delta y Ómicron respectivamente. Cabe destacar que excepto por la edad, el manto de cada variante no tiene más variables en común.

Al buscar una relación entre las comorbilidades y la mortalidad, en Delta se observa que todas las incluidas (fumador, enfermedad pulmonar, historial de enfermedades cardiovasculares, diabetes, hipertensión), todas incluyen la edad como último nodo del camino más corto hacia la mortalidad (siguiendo los arcos sin importar la orientación). En Ómicron observamos que todas las comorbilidades menos el historial de enfermedades cardiovasculares tienen también un camino más corto a la mortalidad que finaliza en la edad. El historial de enfermedades cardiovasculares en cambio, tiene una conexión directa a la mortalidad.

Tras observar indicios de la posible importancia del historial de enfermedades cardiovasculares para la mortalidad, observamos que en Delta, tener presente esta co-

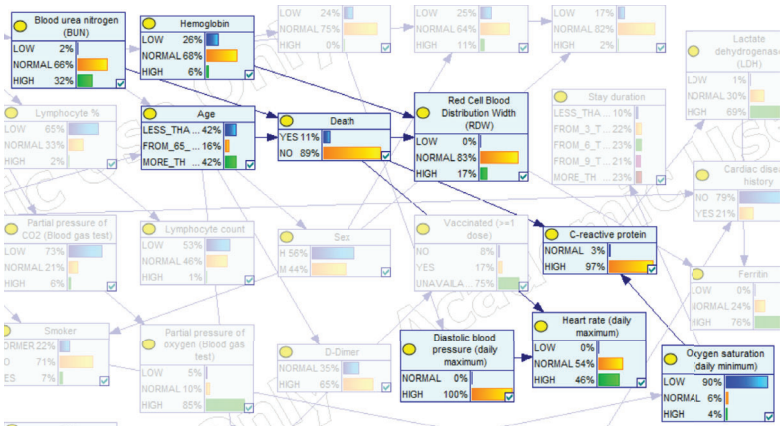
morbilidad incrementa la probabilidad de muerte al 15 %, frente al 11 % en ausencia de la misma. En la red se observa que el impacto es a través de la edad, donde la probabilidad de tener edades superiores a 65 años se incrementa al 74 %, frente al 58 % a priori. En Ómicron en cambio, la presencia de la comorbilidad incrementa la probabilidad de muerte al 18 %, frente al 11 % en ausencia de la misma, y la probabilidad de tener una edad superior a 65 años se incrementa al 94 %, frente al 65 % a priori. Si fijamos en Ómicron el perfil de edades a pacientes mayores de 65 años, la probabilidad de fallecer es del 20 %; si en este perfil establecemos la evidencia de la comorbilidad, la probabilidad de fallecer no llega a aumentar un 1 %, y tampoco hay cambios significativos en ausencia de la comorbilidad. Es decir, incluso si esta comorbilidad forma parte del manto de Markov en Ómicron, realmente no tiene mucha relevancia dentro del mismo perfil de riesgo de edades.

Este ejercicio lo podemos repetir con las distintas variables del manto de Markov de Ómicron. Observamos que bajo el perfil de edades descrito anteriormente, una cuenta de linfocitos por debajo de lo normal incrementa la probabilidad de muerte al 24 %, y una demasiado alta, al 29 %, mientras que se reduciría al 12 % en rangos normales de la variable. Los valores altos de la temperatura máxima diaria incrementan la probabilidad de muerte al 33 %, mientras que los normales la reducen al 16 %. Finalmente, los valores altos del INR, incrementan la probabilidad de muerte al 29 %, mientras que los normales la reducen al 15 %. De esto se extrae que para un mismo perfil de riesgo de edades (con edad mayor a 65 años), la cuenta de linfocitos normal es la configuración que más reduce la probabilidad de fallecer, mientras que la fiebre es la que más la puede incrementar. Tener valores normales de ambas, reduce la probabilidad de fallecer al 10 %, mientras que los valores de mayor riesgo de ambas, incrementan la probabilidad de fallecer al 45 %. El riesgo de fallecer podría incrementarse todavía más al 58 %, con niveles altos del INR.

Esta exploración para el mismo perfil de edades, también se puede repetir en Delta, en su correspondiente manto de Markov, donde la probabilidad de base de fallecer sería también del 20 %, al igual que en Ómicron. Una vez más, hay variables que no tienen impacto apreciable, como la tensión arterial diastólica y la saturación de oxígeno. Un ritmo cardíaco normal en cambio, reduce la probabilidad de fallecer al 14 %, y uno alto, la aumenta al 27 %. Valores normales de la proteína C-reactiva, reducen la probabilidad de fallecer al 2 %, mientras que los altos la incrementan al 21 %. Con el RDW, los valores normales y altos se corresponden respectivamente con una reducción al 16 % y un incremento al 35 % de la probabilidad de fallecer. Con la hemoglobina, los valores bajos incrementan al 24 % la probabilidad de fallecer, mientras que los normales la reducen al 18 %; los valores altos no tienen un impacto apreciable. Finalmente, para el BUN los valores normales reducen la probabilidad de fallecer al 12 %, mientras que los altos, la incrementan al 28 %. En este caso, los valores de la proteína C-reactiva normales dan la menor probabilidad de fallecer,

mientras que los altos del RDW, la maximizan.

También se ha analizado el impacto de la vacunación en la mortalidad, donde se ha visto que no estar vacunado puede reducir en un 1% la probabilidad de muerte en ambas redes, aunque en ambos casos porque el perfil del paciente se estima como más joven para pacientes no vacunados, donde los pacientes más jóvenes tienen menor probabilidad de fallecer. Algo similar ocurre con los fumadores, que tienen mayor probabilidad de tener menos de 65 años, con lo que establecer la evidencia de que el paciente es fumador reduce la probabilidad de muerte (véase Figura 3.14). Este problema se conoce como sesgo de selección, donde analizar de forma individual una variable puede llevar a conclusiones erróneas (véase la explicación de la Sección 2.2.1).

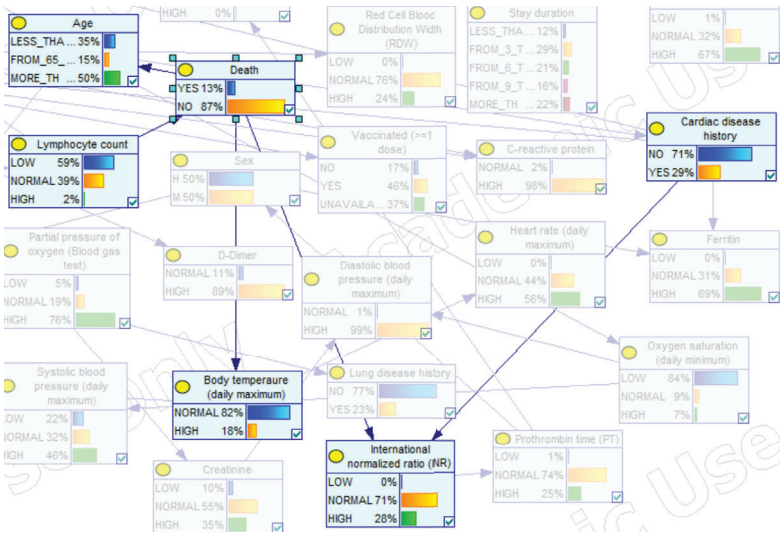


(a) Resaltado en la estructura.

Variable
Diastolic blood pressure (daily minimum)
Heart rate (daily minimum)
Oxygen saturation (daily minimum)
Hemoglobin
Red Cell Blood Distribution Width
Blood urea nitrogen (BUN)
Age
C-reactive protein

(b) Lista de variables del manto.

Figura 3.12: Manto de Markov para la red de Delta.



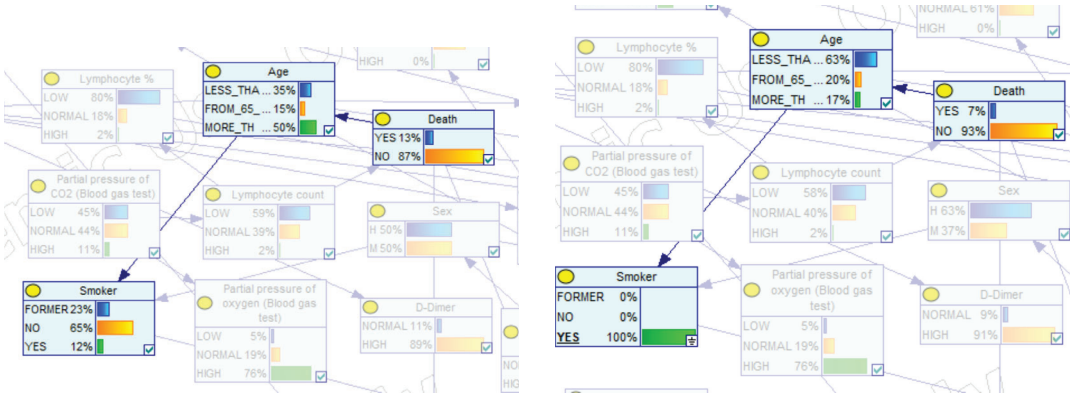
(a) Resaltado en la estructura.

Variable
Cardiac disease history
International normalized ratio (INR)
Age
Body temperature (daily maximum)
Lymphocyte count

(b) Lista de variables del manto.

Figura 3.13: Manto de Markov para la red de Ómicron.

Desafortunadamente, para el tiempo de ingreso solamente se ha encontrado una



(a) Valores de probabilidad a priori de fumador, edad, y muerte.

(b) Valores a posteriori de la edad y muerte tras fijar la evidencia de paciente fumador.

Figura 3.14: Problema del sesgo de selección en fumadores, visualizado con la red.

relación en la red para Delta, y únicamente con el valor de la saturación mínima de oxígeno. De esta relación se destaca que se ha calculado con la red que los pacientes con valores de esta variable en rangos normales, tienen una probabilidad del 79 % de tener un ingreso de duración inferior a 6 días, mientras que la probabilidad sería del 30 % para valores bajos de la variable.

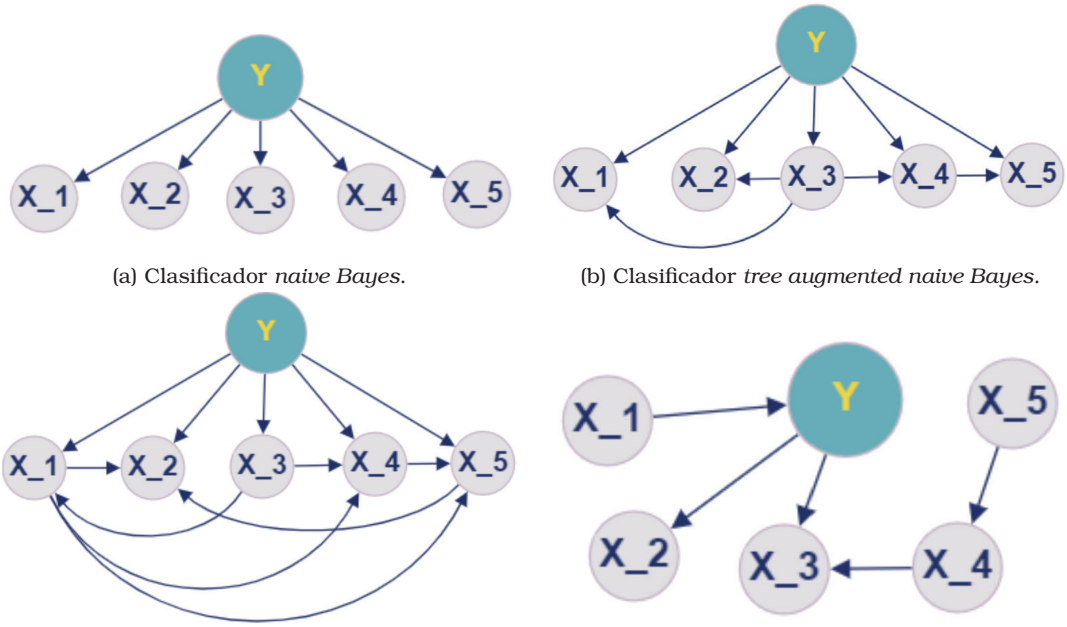
3.4. Diseño de clasificadores Bayesianos Híbridos semiparamétricos

Una de las contribuciones de este trabajo es la ampliación del paquete *PyBNesian* de Atienza y col. (2022), que implementa redes Bayesianas semiparamétricas híbridas, para la construcción de clasificadores Bayesianos Híbridos semiparamétricos. Las redes de este tipo se caracterizan por permitir modelar la distribución de variables continuas y discretas, motivo de que se les llame híbridas, donde se puede modelar la distribución de las variables continuas dados sus padres como *conditional linear Gaussian* (CLG) o como una *conditional kernel density estimation* (CKDE). Las variables continuas puede tener padres continuos o discretos, pero las variables discretas únicamente admiten padres discretos.

Internamente, el paquete está implementado en el lenguaje de programación C++ con incorporación de OpenCL por motivos de rendimiento; sin embargo, la API del paquete está desarrollada en el lenguaje Python para facilitar su uso.

El objetivo de la contribución realizada es adaptar algunos de los trabajos más relevantes del estado del arte en *Bayesian network classifiers* para funcionar con este tipo de red. Concretamente, serían el clasificador *naive Bayes* (NB) (Minsky, 1961), *tree augmented naive Bayes* (TAN) (Friedman y col., 1997) y k-DB (Sahami, 1996) y los *unrestricted Bayesian network classifiers* (UBNC).

El clasificador NB (Figura 3.15a) tiene una estructura predefinida donde la variable clase Y es padre de cada predictora X_i, y no se admiten arcos entre predictoras. El clasificador TAN (Figura 3.15b) extiende NB para que en el subgrafo que no contiene la variable clase, llamado subgrafo de predictoras, tenga una estructura de árbol; con lo que todas las predictoras menos una (raíz del árbol), tendrían exactamente un padre además de la variable clase. El clasificador k-DB (Figura 3.15c) es una generalización del TAN donde las predictoras tienen como mucho k padres además de la clase. Los UBNC (Figura 3.15d) admiten cualquier tipo de estructura de red Bayesiana, con la única restricción en este caso de que las variables discretas no pueden tener padres continuos.



(a) Clasificador *naive Bayes*.

(b) Clasificador *tree augmented naive Bayes*.

(c) Clasificador k-DB con $k = 2$.

(d) *Unconstrained Bayesian network classifier*.

Figura 3.15: Ejemplos de estructuras de distintos tipos de clasificadores Bayesianos.

En el *software* para construcción de BNC discretos *bnclassify* (Mihaljevič y col., 2019), la metodología general para encontrar la estructura de estos clasificadores es *wrapper*, donde se hace una búsqueda codiciosa para encontrar la estructura admitida por el tipo de clasificador que maximice la tasa de aciertos. Aquí en cambio, se ha decidido tomar un enfoque *filter* mediante la información mutua para estimar la estructura de los TAN y k-DB.

En el caso de k-DB, el trabajo original de Sahami (1996) ya utilizaba la información mutua condicional entre predictoras dada la clase, y la información mutua entre las predictoras y la clase, para determinar la estructura del clasificador. Para la construcción de un TAN, Friedman y col. (1997) sugirieron el uso de una extensión del algoritmo de Chow y col. (1968) para estimar la estructura del clasificador. Partiendo de un grafo no dirigido completamente conectado cuyos nodos son las predictoras, siendo el peso de cada arista entre dos nodos la información mutua condicional entre

dichas predictoras dada la clase, se obtiene un árbol recubridor máximo usando el mencionado algoritmo de Chow-Liu.

En este trabajo, al usarse mixturas de variables categóricas y continuas con un enfoque semiparamétrico, se emplea una estimación de la información mutua $MI(X_i, X_j)$ apropiada para variables de distintos tipos y no paramétrica. Más concretamente, se utiliza la implementación de *scikit-learn* de la definición no paramétrica de información mutua de Ross (2014) para mixturas de variables continuas y discretas, que se denominará $MI_{kNN}(X_i, X_j)$, pues está basada en k-vecinos próximos. Sin embargo, para los métodos descritos de estimación de TAN y k-DB se necesita poder calcular la información mutua condicionada a la clase $CMI(X_i, X_j|Y)$. Como la variable clase es categórica, se puede utilizar la definición modular de la información mutua condicional cuando las variables del conjunto condicionante (en este caso, la clase) son categóricas, tal y como se ve en la Ecuación 3.1:

$$CMI(X_i, X_j|Y) = \sum_{y \in Dom(Y)} P(Y = y) \cdot MI(X_i, X_j|Y = y) \tag{3.1}$$

con $Dom(Y)$ el dominio de Y.

Para estimar $MI(X_i, X_j|Y = y)$, se usan los datos donde $Y = y$ para aplicar la ya mencionada $MI_{kNN}(X_i, X_j)$, y $P(Y = y)$ simplemente se estima por el método de máxima verosimilitud. Cabe mencionar que al estar basada en k vecinos más cercanos, un parámetro más a configurar al utilizar está definición de la información mutua es el propio valor de k.

Para la construcción de clasificadores TAN, cabe mencionar que el algoritmo de Chow-Liu produce grafos no dirigidos, que Friedman y col. (1997) proponían orientar tomando un nodo como raíz y dirigiendo los nodos hacia fuera recursivamente. Aquí, como se tiene la restricción de que las variables categóricas no pueden tener padres continuos, se utiliza la ampliación del algoritmo de Edmonds (1967) que permite partir de un grafo dirigido completamente conectado, y producir un árbol recubridor máximo dirigido. De esta forma, se puede no tener en cuenta los arcos de variables continuas a variables discretas al aplicar el algoritmo. Para la construcción de estructuras k-DB, simplemente no se tienen en cuenta como candidatos los arcos de variables discretas a variables continuas al aplicar el algoritmo del trabajo original.

Adicionalmente, en el diseño realizado, se permite utilizar una versión selectiva *filter* de los clasificadores NB, TAN y k-DB utilizando la información mutua entre las predictoras y las clases, que no tiene en cuenta las variables tales que $MI_{kNN}(X_i, Y) < \alpha$, con α un hiperparámetro a configurar. Esto es similar al método de Pazzani y col. (1997), con la diferencia de que en este trabajo se probaba a seleccionar las m variables con mayor información mutua.

Para la construcción de los *unconstrained* BNC, se usa un enfoque *wrapper*, pero que maximiza un *score* descomponible de la red. En el caso de *PyBNesian*, se utiliza el logaritmo de la verosimilitud (*log-likelihood*) de la estructura, pero empleando diferentes estrategias de particionado de los datos para evitar el sobreajuste de la red. En concreto, bien se separa un conjunto de validación (*holdout*), o se realiza una validación cruzada (*cross validation*) y al evaluar la verosimilitud de la estructura, se estiman los parámetros en el conjunto de entrenamiento correspondiente a la partición actual, y se calcula la verosimilitud sobre el conjunto de validación. También se puede utilizar una combinación de ambas en el proceso de búsqueda, usando una partición de validación para determinar el criterio de parada, y una validación cruzada en la partición entrenamiento para estimar la verosimilitud de la red en muestras no vistas. Véase Atienza y col. (2022) para una explicación en mayor profundidad.

Teniendo como base las redes Bayesianas semiparamétricas, parte del proceso de construcción de la red, es determinar el tipo de distribución asignada a cada variable continua, que puede ser CLG, o CKDE. En el caso de las *unconstrained BNC*, el propio proceso de búsqueda de la estructura de la red, también busca el tipo de distribución de las variables continuas que maximiza el *score* correspondiente. Para NB, TAN y k-DB, se realiza también un proceso de búsqueda sobre la estructura obtenida con el algoritmo correspondiente, pero solamente en el espacio de tipos de distribución de las variables continuas; es decir, la estructura permanece igual.

Una vez obtenida la estructura G y parámetros $\boldsymbol{\theta}$ de los clasificadores, se puede calcular la probabilidad a posteriori de la clase Y dadas las predictoras a partir de la distribución conjunta de la red en los distintos valores que puede tomar Y, tal y como se muestra en la Ecuación 3.2:

$$P(Y = y|X_1 = x_1, ..., X_n = x_n) \propto P(Y = y) \cdot P(X_1 = x_1, ..., X_n = x_n|Y = y) \tag{3.2}$$

Donde el segundo término $P(X_1 = x_1, ..., X_n = x_n|Y = y)$ se factoriza según la estructura del clasificador considerado.

La implementación[4] de estos clasificadores se ha realizado en Python, ajustándose la API de estimadores *scikit-learn*. Para cada uno de los cuatro tipos de clasificador mencionados, se incluye la variante semiparamétrica y la variante más tradicional CLG, donde todos los nodos continuos son de este tipo; es decir, en los procesos de búsqueda no se plantea realizar cambios de tipo a las variables continuas. También se ha añadido la correspondiente variante original discreta, aunque admitiéndose solamente cuando todos los nodos son, o pueden ser, de tipo discreto. Es decir, se han implementado un total de 12 clasificadores, o tres variantes de cuatro formas

[4]Implementación de clasificadores Bayesianos híbridos semiparamétricos: `https://github.com/johacks/anexoTFM/blob/main/spbn_classifier.py`

de obtener la estructura de la red. Véase en la Tabla 3.6 los 12 tipos de clasificador Bayesiano implementados con el nombre de la clase Python asociada en la implementación.

Structure \ Network type	CLG	Semiparametric	Discrete
Naive Bayes (k-DB with $k = 0$)	KDBBNClassifierCLG	KDBBNClassifierSP	KDBBNClassifierDisc
Chow-Liu TAN	CLTANBNClassifierCLG	CLTANBNClassifierSP	CLTANBNClassifierDisc
k-DB	KDBBNClassifierCLG	KDBBNClassifierSP	KDBBNClassifierDisc
UBNC	HCClassifierCLG	HCClassifierSP	HCClassifierDisc

Tabla 3.6: Nombre de la clase en Python de los 12 clasificadores Bayesianos implementados por tipo de estructura y tipo de red Bayesiana.

3.5. Construcción de modelos predictivos por olas

3.5.1. Descripción del procedimiento

Como parte del análisis por olas de la mortalidad, se decidió probar a construir distintos modelos predictivos para cada una, con los siguientes objetivos:

- Comparar la fiabilidad y robustez de distintos tipos de modelos a lo largo de las distintas olas.
- Obtener una estimación de la evolución del error mínimo en la predicción de la mortalidad a lo largo de diferentes olas.
- Poner a prueba los clasificadores Bayesianos Híbridos (HBNC) descritos en la Sección 3.4.

Para todos los clasificadores se utilizó una implementación del paquete *scikit-learn*, o en el caso de los HBNC, la implementación realizada con la API seguida por *scikit-learn*. La selección de variables empleada para cada ola fue la producida por el proceso de *recursive feature elimination* (RFE) descrito en la Sección 3.2 con el modelo GBM (véase Apéndice B para un listado). Un primer motivo para usar la selección de GBM es que obtuvo mejores resultados en cuanto al AUC estimado en el proceso de RFE. El segundo motivo es que para todas las olas produjo una selección conformada únicamente por variables continuas, con lo que se podrían utilizar los modelos de *scikit-learn* sin tener que recurrir a estrategias de codificación de las variables categóricas a indicadores.

El procedimiento seguido para cada clasificador fue el mismo: se estimaron los distintos hiperparámetros utilizando optimización Bayesiana (Snoek y col., 2012) con la implementación del paquete Python *scikit-optimize*. El algoritmo trata los valores de la métrica M de clasificación a optimizar de un modelo como una variable aleatoria $\boldsymbol{X}$, que se modela normalmente como una Gaussiana multivariante $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, donde su covarianza viene dada por el kérnel de un proceso Gaussiano entre los hiper-

parámetros a configurar. Iterativamente, se determina una siguiente configuración de hiperparámetros a evaluar $\boldsymbol{\lambda}$, que maximice una función de adquisición sobre la priori de dicha distribución, donde la función de adquisición normalmente se define para que haya un balance entre exploración y explotación en la búsqueda. Tras evaluar M con el modelo construido con $\boldsymbol{\lambda}$, se actualizan los parámetros de $\mathcal{N}$ con dicha evaluación y se repite el proceso hasta alcanzar un criterio de parada.

El algoritmo buscó la configuración de hiperparámetros de cada modelo que maximizaba su AUC en datos no vistos en el entrenamiento. Dicha métrica se estimó en cada iteración mediante una validación cruzada con 5 particiones. Este proceso se realizó sobre la partición de entrenamiento de la división entrenamiento-validación-test mencionada en la Sección 3.1.6, empleando los primeros datos conocidos de cada paciente. Con la partición de validación, una vez estimada la mejor configuración de hiperparámetros de cada clasificador, se dibujó la curva de calibración, curva ROC y se calculó el AUC, *Brier score*, tasa de aciertos, especificidad y sensibilidad.

Finalmente, se seleccionó para cada ola el clasificador con mejor métrica AUC en el conjunto de validación, y tras volver a entrenar el modelo con el conjunto combinado de entrenamiento y validación, se repitió la estimación de las distintas métricas sobre el conjunto de test.

Los modelos probados fueron una regresión logística regularizada, árboles de decisión, *Gradient boosting decision trees* (GBDT), y las distintas versiones de los HBNC desarrollados para variables continuas y discretas, es decir, con *conditional linear Gaussian* (CLG) y con redes híbridas semiparamétricas (SP). Como se tiene una gran variedad de clasificadores y resultados, los detalles del resultado de cada uno en el conjunto de validación se anexan en *GitHub* en el formato HTML[5]. De esta forma, solamente se subrayará lo más relevante del procedimiento.

El primer modelo probado fue una regresión logística regularizada. En un problema de clasificación binaria como este, el modelo estima la probabilidad de la clase positiva como el resultado de aplicar la función sigmoide sobre una combinación lineal de las predictoras. La regularización busca contraer el valor de los pesos de dicha combinación lineal. En este caso se utilizó la regularización *elasticnet*, que combina la penalización Lasso y Ridge mediante una ponderación de ambas. Como optimizador de los pesos se empleó el algoritmo SAGA (Defazio y col., 2014). La función de pérdida que minimiza SAGA viene dada por la ecuación 3.3:

$$\frac{-log\mathcal{L}(\boldsymbol{\beta}, \mathcal{D})\mathrm{pen}(\boldsymbol{\beta})}{C} \tag{3.3}$$

[5]Detalle de resultados para cada modelo y ola: `https://johacks.github.io/anexoTFM/modelos.html`

donde $\mathcal{L}(\boldsymbol{\beta}, \mathcal{D})$ es la verosimilitud de los pesos $\boldsymbol{\beta}$ para un conjunto de datos $\mathcal{D}$ y $\texttt{pen}(\boldsymbol{\beta}) = \alpha \cdot l_1(\boldsymbol{\beta}) + (1-\alpha) \cdot l_2(\boldsymbol{\beta})$ es la mencionada penalización *elasticnet*, que pondera la penalización l_1 (Lasso) y l_2 (Ridge). C es el inverso de la fuerza regularización, y como se puede deducir, a menores valores, mayor es la regularización.

Los hiperparámetros considerados en la búsqueda fueron los siguientes:

- C: inverso de la fuerza de regularización. Valores entre 0.1 y 100.
- α: peso de la regularización Lasso, complementario al peso de la regularización Ridge; al optimizar este hiperparámetro, se busca alcanzar un equilibrio óptimo entre ambos tipos de regularización. Valores entre 0 y 1.
- Balance de clases: asignar pesos a las clases inversamente proporcionales a su frecuencia de aparición en los datos. Se considera aplicar este balance o no.

El segundo modelo que se probó fueron los árboles de decisión. Este clasificador se basa en agrupar por intervalos y de forma jerárquica los valores de las predictoras, y después realizar predicciones de instancias nuevas según la clase mayoritaria del nodo hoja al que pertenece. La implementación de *scikit-learn* está basada en el trabajo de Breiman y col. (2017) y admite una gran variedad de hiperparámetros. Se seleccionaron los siguientes para su optimización:

- Criterio de separación: métrica utilizada para evaluar la mejora del árbol tras hacer una división. Criterio elegido entre Gini, entropía, y *log loss*.
- Profundidad máxima del árbol. Valores entre 1 y 8.
- Mínimo de muestras para separar. Valores entre 2 y 15.
- Mínimo de muestras por hoja del árbol. Valores entre 1 y 10.
- Balance de clases: asignar pesos a las clases inversamente proporcionales a su frecuencia de aparición en los datos. Se considera aplicar este balance o no.
- Mínimo de reducción de impureza. Valores entre 0 y 0.1.
- Número máximo de variables a considerar para realizar cortes. Se consideró utilizar el número total de variables n, $\sqrt{n}$, y $log_2(n)$.
- α: parámetro de complejidad utilizado en *Minimal Cost-Complexity Pruning*, un proceso de poda tras la construcción del árbol que ayuda a evitar el sobreajuste. Los valores más altos de α imponen mayores restricciones a la complejidad del árbol, determinada en base al número de nodos terminales y grado de impureza de cada uno. Valores entre 0 y 0.035.

El tercer modelo utilizado fue GBDT, un modelo basado en conjuntos de árboles, donde los sucesivos árboles se especializan en resolver los fallos de los anteriores. La

implementación de *scikit-learn* se basa en el trabajo de Friedman (2001). Se optimizaron los siguientes hiperparámetros del algoritmo:

- Número de árboles. Valores entre 5 y 150.
- Tasa de aprendizaje: parámetro que regula la reducción de la importancia de los árboles nuevos que se van integrando al modelo para evitar el sobreajuste. Valores entre 0.05 y 0.2.
- Número de variables consideradas en los árboles del modelo al realizar cortes. Considerado el total del número de variables n y $log_2(n)$.
- Número de iteraciones sin mejora de la función de pérdida del modelo. Valores entre 1 y 10.

Finalmente, se probaron los HBNC implementados. Para *naive Bayes*, TAN y k-DB, se optimizaron los siguientes hiperparámetros:

- Número de vecinos: número de vecinos empleados en la estimación de la información mutua. Probados los valores de 5, 10, 50 y 100.
- Umbral de selección de variables α: valor que tiene que superar o igualar cada predictora en cuanto a información mutua con respecto a la clase para ser incluida en el modelo. Probados los valores de 0 y 0.1

Para el k-DB por restricciones de tiempo, se decidió fijar $k = 2$, puesto que se esperaría que $k = 1$ ya produjera resultados similares al TAN con el algoritmo de Chow-Liu, y $k = 0$ es *naive Bayes*. Adicionalmente, para los clasificadores mencionados en la versión semiparamétrica, se optimizó el mecanismo de selección de *bandwidth* del kernel, eligiendo entre aplicar la regla de Scott (Scott, 2015), y la referencia normal (detallada por Atienza y col. (2022)).

Con los *unconstrained* BNC, para la versión semiparamétrica se fijó el *score* empleando *holdout* y la selección de *bandwidth* utilizando la regla de Scott; y para la versión CLG, se utilizó el *score* BIC. Como hiperparámetros, se optimizaron:

- ϵ: incremento mínimo del *score* para continuar la búsqueda. Se consideraron valores entre 0.000001 y 1.
- Máximo número de padres de cada nodo. Considerados valores en $\{1, ..., 5, \infty\}$ o para las CLG, y en $\{1, .., 3\}$ para las semiparamétricas (por motivos de tiempos de ejecución).

3.5.2. Comparativa de los resultados

Tras estimar los parámetros y calcular métricas sobre el conjunto de validación correspondiente a cada grupo de olas, se generó la tabla que se muestra en la Tabla

3.7, donde los colores azul a rojo se corresponden respectivamente con valores más bajos a más altos de forma relativa al rango de valores de la columna. De la tabla, realmente las únicas métricas importantes son el AUC/ROC y el *Brier score*, pues tienen en cuenta las probabilidades producidas por el modelo, mientras que el resto se basan en predicciones máximas a posteriori, que normalmente no son óptimas cuando hay un desbalance de clases.

	AUC				BRIER SCORE				SENSITIVITY				SPECIFICITY				ACCURACY			
	W 1	W 2	W 3,4,5	W 6	W 1	W 2	W 3,4,5	W 6	W 1	W 2	W 3,4,5	W 6	W 1	W 2	W 3,4,5	W 6	W 1	W 2	W 3,4,5	W 6
LogisticRegression	0.87	0.88	0.82	0.83	0.87	0.91	0.83	0.90	0.60	0.35	0.74	0.14	0.90	0.97	0.73	0.96	0.81	0.87	0.73	0.86
DecisionTreeClassifier	0.80	0.80	0.80	0.78	0.83	0.83	0.81	0.81	0.72	0.89	1.00	0.73	0.82	0.57	0.51	0.79	0.79	0.63	0.56	0.79
GradientBoostingClassifier	0.88	0.88	0.89	0.82	0.88	0.91	0.93	0.90	0.61	0.28	0.15	0.09	0.92	0.98	0.98	0.99	0.82	0.86	0.89	0.88
Naive-Bayes-CLG	0.84	0.85	0.82	0.75	0.80	0.85	0.88	0.83	0.55	0.49	0.37	0.23	0.88	0.90	0.93	0.88	0.78	0.83	0.86	0.80
Naive-Bayes-SP	0.86	0.86	0.82	0.80	0.83	0.87	0.90	0.88	0.59	0.49	0.22	0.27	0.89	0.90	0.94	0.95	0.79	0.83	0.86	0.87
Chow-Liu-TAN-CLG	0.84	0.85	0.80	0.73	0.80	0.85	0.87	0.81	0.55	0.51	0.33	0.23	0.88	0.89	0.92	0.86	0.78	0.82	0.86	0.78
Chow-Liu-TAN-SP	0.86	0.84	0.78	0.74	0.83	0.84	0.89	0.85	0.56	0.46	0.22	0.23	0.90	0.89	0.96	0.92	0.80	0.82	0.88	0.83
2DB-BNC-CLG	0.84	0.84	0.81	0.76	0.80	0.86	0.87	0.85	0.52	0.47	0.33	0.18	0.89	0.90	0.93	0.92	0.77	0.83	0.86	0.83
2DB-BNC-SP	0.83	0.85	0.82	0.75	0.80	0.88	0.89	0.87	0.53	0.37	0.22	0.18	0.89	0.95	0.95	0.93	0.78	0.85	0.88	0.84
Unconstrained-BNC-CLG	0.84	0.82	0.75	0.66	0.81	0.84	0.87	0.84	0.58	0.40	0.26	0.09	0.87	0.90	0.94	0.92	0.78	0.82	0.86	0.82
Unconstrained-BNC-SP	0.85	0.86	0.77	0.73	0.84	0.90	0.91	0.89	0.43	0.37	0.07	0.23	0.94	0.96	0.98	0.95	0.78	0.86	0.88	0.86

Tabla 3.7: Métricas por modelo y ola en el conjunto de validación correspondiente.

De los modelos probados se observó que la regresión logística tenía siempre buenos resultados de calibración (según el *Brier score*) y AUC, aunque excepto por la ola 6, correspondiente a Ómicron, siempre era superada por el algoritmo GBDT. Los árboles de decisión en ningún caso funcionaron comparativamente bien con respecto al resto de modelos, salvo quizás por las olas 3, 4 y 5, correspondientes a la variante Delta, donde sin llegar a tener un AUC mucho peor que el resto de modelos, alcanzó una muy buena sensibilidad, que es complicado de lograr, teniendo en cuenta que la clase minoritaria es el fallecimiento del paciente (caso positivo de mortalidad).

De los clasificadores Bayesianos, los mejores resultados en AUC siempre los proporcionó el *naive Bayes* en su versión semiparamétrica. Si nos fijamos solamente en la rama de las CLG, también la rama con *naive Bayes* es la que da mejores resultados según este criterio. El buen funcionamiento de esta estructura encaja con la efectividad de la regresión logística, que también es un modelo lineal. La excepción a la regla se da en la variante Delta, donde los modelos de *naive Bayes* y regresión logística funcionaron significativamente peor que los GBDT. Del resto de clasificadores, cabe mencionar que la versión semiparamétrica del *unconstrained* BNC funcionó siempre mejor que la versión CLG en cuanto a AUC. También que el *unconstrained* BNC tuvo mejores resultados en las primeras dos olas, pero más adelante mermó significativamente su rendimiento.

Tras observar estos resultados, se repitió la estimación en el ya mencionado conjunto

de test con el mejor modelo para cada variante en cuanto a AUC en el conjunto de validación. En este caso sería GBDT para las tres primeras variantes, y la regresión logística para Ómicron. El AUC estimado fue de 0.89, 0.88, 0.87 y 0.75 en las respectivas variantes Wuhan, Alfa, Delta y Ómicron. Con estos resultados, y los del conjunto de validación, queda patente que el error mínimo ha aumentado significativamente para la variante Ómicron.

3.6. Adaptación de un algoritmo de explicaciones contrafactuales

En los apartados anteriores se ha visto que los modelos opacos pueden ser los más eficaces en cuanto a las predicciones. Sin embargo, su falta de transparencia es limitante para la implantación de este tipo de modelos en el ámbito médico. En la Sección 2.2.3 se discutió que una de las formas de mejorar la interpretabilidad de los modelos de aprendizaje automático es el uso de explicaciones contrafactuales.

Siguiendo la notación de la sección referenciada, idealmente, se busca dada una entrada al modelo $\mathbf{x} = (x_1, ..., x_n)$, una o varias explicaciones contrafactuales $\boldsymbol{P} = \{\mathbf{x'}^{(1)}, ..., \mathbf{x'}^{(P)}\}$ tal que la calidad de cada una de ellas $\mathbf{x'} \in \boldsymbol{P}, \mathbf{x'} = (x'_1, ..., x'_n)$, se mide por su plausibilidad, similitud a la entrada original $\mathbf{x}$, y similitud de la salida del modelo $\phi(\mathbf{x'}) = y'$ a la de la salida contrafactual esperada y_c. Por ejemplo, para un paciente al que se pronostica el fallecimiento $Y = y$ por un modelo ϕ dada una serie de medidas de laboratorio $\mathbf{x}$, es decir $\phi(\mathbf{x}) = y$, podrían buscarse valores alternativos similares $\boldsymbol{P}$ tales que el modelo pronosticaría la supervivencia y_c, para entender así mejor el funcionamiento seguido por el modelo para la clasificación.

Una de las aportaciones de este trabajo es la adaptación del algoritmo de búsqueda de explicaciones contrafactuales de Dandl y col. (2020) mediante el uso de redes Bayesianas.

3.6.1. Diseño original y modificaciones

En el trabajo de Dandl y col. (2020), se usa NSGA-II (Deb y col., 2002), un algoritmo evolutivo de optimización multiobjetivo, para los tres objetivos de plausibilidad, similitud a la entrada original, y similitud a la salida deseada. Al utilizarse un algoritmo multiobjetivo, se busca obtener soluciones distribuidas a lo largo de la frontera Pareto óptima, que es el conjunto de soluciones alcanzables tal que ninguna es mejor que estas en cuanto a todos los objetivos a la vez. Naturalmente, hay que cuantificar de alguna forma el grado de cumplimiento de cada objetivo; para ello, en el trabajo original se utilizan los siguientes criterios:

- Similitud de $\mathbf{x}$ a $\mathbf{x'}$, cuya maximización se expresa como la minimización de

su distancia. Se usó la distancia de Gower, que es el promedio de la distancia entre cada componte de los vectores $\sum_{i=1}^{n} d(x_i, x'_i)/n$. Cuando X_i es una variable aleatoria continua, se usa la distancia Manhattan normalizada al rango $[0, 1]$ dividiendo entre $\text{máx}(X_i) - \text{mín}(X_i)$: $d(x_i, x'_i) = |x'_i - x_i|/(\text{máx}(X_i) - \text{mín}(X_i))$. Si X_i es una variable aleatoria categórica, la distancia es $d(x_i, x'_i) = \mathbb{1}^{\mathbb{C}}(x_i, x'_i)$, con $\mathbb{1}^{\mathbb{C}}(x_i, x'_i) = 0$ cuando $x_i = x'_i$, y 1 en otro caso. En el trabajo original, se añade un segundo objetivo a minimizar: el número de variables cambiadas, para obtener explicaciones más interpretables, donde sea fácil localizar los cambios respecto a la entrada original.

- Similitud de y' a y_c. Para un problema de clasificación binaria como este, se podría tomar la salida y' como la probabilidad de la clase positiva, e $y_c = 1$ cuando se buscan predicciones de la clase positiva, y 0 en caso contrario. Para problemas no binarios, se haría lo análogo, usando la clase y_c como referencia. Se expresa como la minimización de la distancia Manhattan entre y' e y_c.

- Plausibilidad de **x'**. Es preferible que las explicaciones contrafactuales no sean inverosímiles. En el trabajo original se cuantifica la verosimilitud como la mínima distancia de Gower entre **x'** y cualquier entrada del conjunto de entrenamiento original $\mathcal{D}$. Cabe mencionar que no establecieron ningún umbral a partir del cual **x'** tendría suficiente verosimilitud.

La distancia de Gower tiene el problema de que queda sesgada cuando las variables continuas tienen valores extremos, que provocan un trato potencialmente muy diferente entre unas variables continuas y otras. Todavía hay más sesgo al comparar las distancias de variables continuas con las de las variables categóricas, donde el simple cambio de la variable genera una distancia de 1 en las categóricas, mientras que la distancia entre las continuas muy difícilmente puede valer cerca de 1, pues implicaría que ambos puntos se encuentren en extremos contrarios de la distribución.

En este trabajo se propone modificar la distancia entre variables continuas para usar una versión truncada de la distancia empleada por Wachter y col. (2017) en su trabajo de contrafactuales. Esta distancia simplemente cambia la normalización aplicada a las variables continuas, normalizando la distancia Manhattan con la desviación mediana absoluta $d(x_i, x'_i) = |x'_i - x_i|/(DMA(X_i))$ en vez de con el rango de la variable, haciéndola más robusta a los valores extremos. Sin embargo, esta distancia no produce valores en el rango $[0, 1]$, lo que impide que sea comparable con la distancia entre variables categóricas.

Para solventar este problema, se toma la tangente hiperbólica de la distancia, que para entradas positivas produce valores en el rango $[0, 1]$. En la Figura 3.16 se puede consultar el truncado que hace la tangente hiperbólica de valores positivos: para una entrada de valor 1, es decir, cuando dos valores están separados por la desviación

mediana absoluta, la distancia truncada sería de $\tanh(1) \approx 0.75$, y para dos veces la desviación mediana absoluta, sería de $\tanh(2) \approx 0.96$. Es decir, se hace comparable el cambio de una variable categórica al de una variable continua en dos veces la desviación mediana absoluta. Otra propiedad es que, para entradas inferiores a 1, la tangente hiperbólica produce resultados casi directamente proporcionales a la entrada.

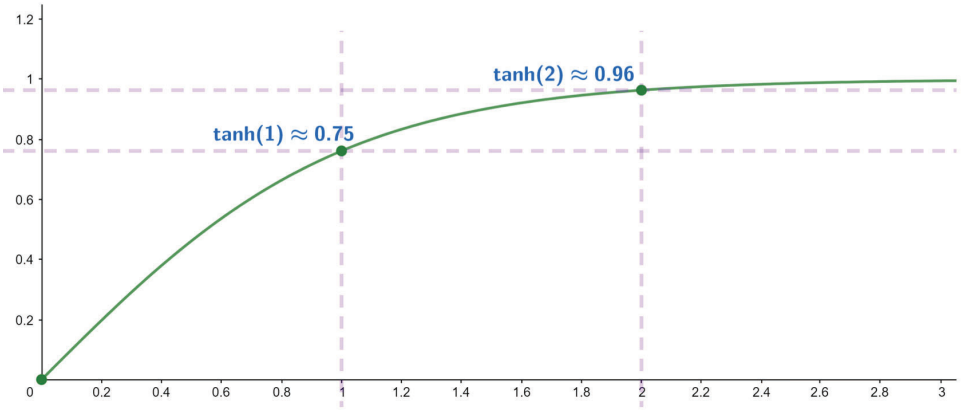


Figura 3.16: Truncado de una función de recorrido positivo al rango $[0, 1]$ mediante la tangente hiperbólica.

Con todo, la distancia D descrita para dos vectores de variables continuas y discretas queda reflejada en la Ecuación 3.4 y 3.5.

$$D(\mathbf{x}, \mathbf{x'}) = \sum_{i=1}^{n} \frac{d(x_i, x'_i)}{n} \tag{3.4}$$

$$\text{donde } d(x_i, x'_i) = \begin{cases} \tanh\left(\frac{|x_i - x'_i|}{DMA(X_i)}\right) & \text{si } X_i \text{ es continua} \\ \mathbb{1}^{\complement}(x_i, x'_i) & \text{en otro caso.} \end{cases} \tag{3.5}$$

También, para simplificar el proceso de búsqueda, se elimina el objetivo del trabajo original de cambiar la menor cantidad de variables posible. En su lugar, se introduce una operación de reparación aplicada en cada iteración del algoritmo NSGA-II tal que si $d(x_i, x'_i) < \delta$, se asigna $x'_i \leftarrow x_i$, con δ un parámetro a establecer. De esta forma, se evitan explicaciones **x'** con cambios en muchas variables continuas x'_i, pero que apenas difieren del valor original x_i de la variable. Empíricamente, se ha visto que $\delta \in [0.05, 0.2]$ da buenos resultados. Cabe mencionar que por la Ecuación 3.5, para $\delta = 0.2$ la distancia mínima $d_i(x_i, x'_i) = \delta = 0.2$ tal que se acepta el cambio de x_i a x'_i al aplicar el operador de reparación, implica que $\frac{|x_i - x'_i|}{DMA(X_i)} = \tan(0.2) \approxeq 0.2$. Es decir, x'_i tiene que variar con respecto a x_i en valor absoluto un mínimo del $20\,\%$ de la desviación mediana absoluta de la variable $DMA(X_i)$. Lo análogo aplica para $\delta = 0.05$, pues para valores bajos, $\tan(x) \approxeq x$.

Para la similitud de y' a y_c, se ha añadido la restricción opcional a las soluciones candidatas de que la salida y' debería ser la explicación más probable en el caso de variables categóricas.

Finalmente, como aspecto más novedoso, se emplea una red Bayesiana semiparamétrica híbrida (Atienza y col., 2022) para estimar la plausibilidad de las soluciones candidatas. La estructura de la red se puede aprender mediante una búsqueda voraz, y los parámetros mediante máxima verosimilitud para este caso. La plausibilidad de una solución **x'** para una etiqueta y_c en la variable clase, se estimaría como la probabilidad conjunta de darse dicha instancia, es decir, $P(X_1 = x'_1, ..., X_n = x'_n, Y = y_c)$, que puede calcularse mediante la factorización de la red. Para facilitar el proceso de búsqueda, se decidió tratar este objetivo de plausibilidad como una restricción a las soluciones candidatas, es decir, exigir un valor mínimo de $P(\mathbf{x'}, Y = y_c)$. La problemática reside en fijar ese valor para distintos valores del número de variables n, donde valores más altos de n implican valores más bajos en la conjunta.

Se observó en una red semiparamétrica híbrida construida con los datos $\mathcal{D}$ de la variante Ómicron, que el logaritmo de la probabilidad conjunta de las distintas instancias $(\mathbf{x}^{(k)}, y^{(k)}) \in \mathcal{D}$ etiquetadas con $y^{(k)} = y_c$ para la variable clase Y de la mortalidad (con las dos posibles definiciones de y_c, es decir, supervivencia o muerte), expresado como $log(P(X_1 = x_1^{(k)}, ..., X_n = x_n^{(k)}, Y = y^{(k)} = y_c))$, seguía una distribución similar a una normal (véase Figura 3.17). Por tanto, asumiendo $log(P(X_1 = x_1^{(k)}, ..., X_n = x_n^{(k)}, Y = y^{(k)} = y_c)) \sim \mathcal{N}(\mu, \sigma)$, se definió para cada explicación candidata **x'**, la restricción $log(P(X_1 = x'_1, ..., X_n = x'_n, Y = y_c)) > \beta$, con el valor por defecto $\beta = \mu - \sigma$. Es decir, estimamos que la solución candidata **x'** debería estar en un percentil ≈ 16 o superior en cuanto a la métrica de la plausibilidad propuesta en las observaciones previas etiquetadas con y_c, pues es el valor del percentil asociado a $\mu - \sigma$ en una normal $\mathcal{N}(\mu, \sigma)$.

Como método adicional para aumentar la plausibilidad de las soluciones candidatas, se añadió un operador de reparación que mantenía las variables continuas en los rangos de valores observados en los datos.

En cuanto al optimizador, NSGA-II, se encargará de optimizar los dos objetivos mencionados: probabilidad a posteriori de la clase esperada y similitud a la instancia original; bajo las restricciones de plausibilidad mínima y que la clase esperada sea la explicación más probable. Para ello, seguirá un proceso iterativo que irá modificando y reparando el conjunto (población) de soluciones candidatas (individuos). Los operadores de reparación ya se ha mencionado que sirven para evitar cambios demasiado sutiles en las variables continuas, así como para evitar que tomen valores fuera de su rango observado previamente.

La modificación de la población se hace mediante operaciones de cruce y mutación.

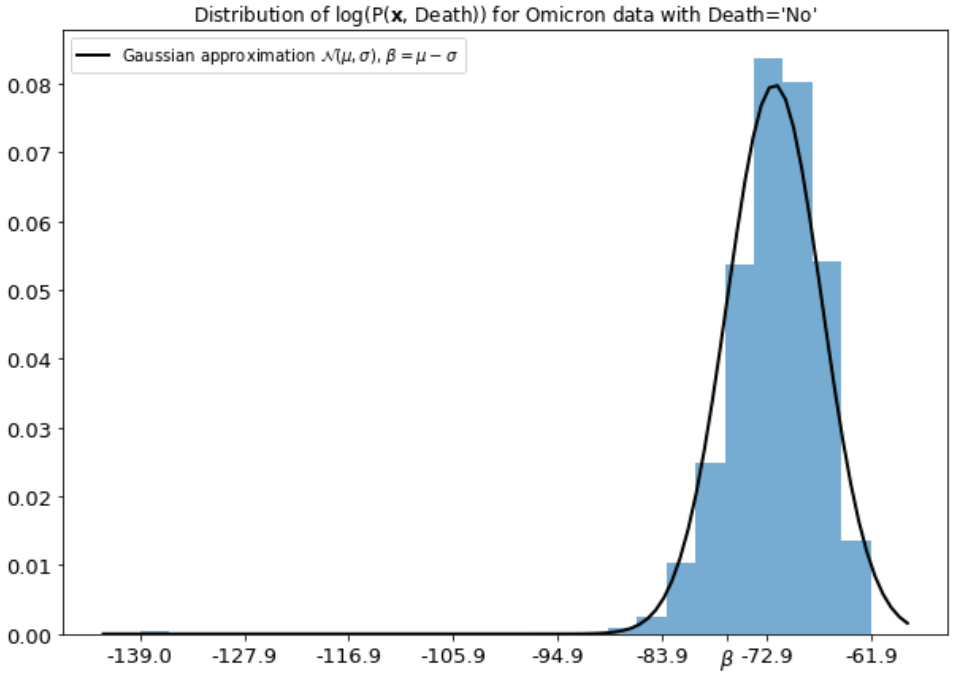


Figura 3.17: Distribución de $log(P(\mathbf{x}^{(k)}, y^{(k)}))$ en muestras $(\mathbf{x}^{(k)}, y^{(k)})$ con $y^{(k)} = y_c$ de Ómicron, con y_c la supervivencia del paciente.

El operador de cruce se encarga de generar a partir de dos individuos $\mathbf{x'}^{(i)}, \mathbf{x'}^{(j)}$ (padres) otras dos soluciones nuevas (descendencia) con características de los padres. El operador de mutación modifica de forma separada cada individuo **x'**, realizando pequeños cambios en el mismo.

En el trabajo original, la población $\boldsymbol{P}$ se inicializa con muestras aleatorias del dominio de cada variable, y después se establecen también aleatoriamente algunos valores de la muestra original $x'_i \leftarrow x_i$ para cada $\mathbf{x'} \in \boldsymbol{P}$. Como operadores de cruce, se propone usar los propuestos por Syswerda (1989) y por Deb y col. (1995) para variables categóricas y numéricas respectivamente. Para mutar las variables, se hace uso de *transformation forests* (Hothorn y col., 2017), que permiten realizar pequeñas modificaciones verosímiles en **x'**.

Aquí la aproximación tomada es más simple para las tres operaciones mencionadas. En la inicialización de la población $\boldsymbol{P}$, se toma como población inicial el resultado de mutar varias copias de la muestra original **x**. Para la operación de cruce se intercambia el valor de una de las variables entre los padres para generar la descendencia. El operador de mutación se modifica empleando la propia red Bayesiana para muestrear de la distribución $P(X_i|\boldsymbol{Pa}_i = \boldsymbol{pa'}_i)$ un nuevo valor x'_i de cada variable mutada X_i en **x'**, donde $\boldsymbol{pa'}_i$ sería la configuración de los padres de X_i en **x'**. En cuánto a qué variables son mutadas (en orden topológico), estas se escogen al azar y sin reemplazo. La cantidad de variables seleccionadas para la mutación de cada **x'**, se obtienen de una distribución de Boltzmann con $\lambda = 2$ y $N = 3$ por defecto, que se ve en la Figura

3.18. La idea es que en general no se mute para una solución candidata más de 1 variable a la vez, pero con una baja probabilidad se admita el cambio de dos o tres variables, permitiendo así escapar de óptimos locales. Tanto para el cruce como para la mutación, se puede especificar qué variables pueden ser cambiadas en el proceso de búsqueda.

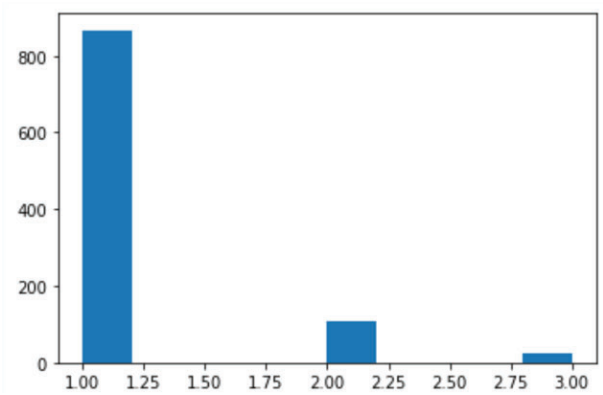


Figura 3.18: Distribución de Boltzmann con $\lambda = 2$ y $N = 3$

Como criterio de parada del algoritmo, se utiliza la convergencia del hipervolumen (Hutchinson, 1957). Para problemas de dos objetivos como este, el hipervolumen mide el área máxima que se puede formar en el espacio de los objetivos a partir de las soluciones existentes y un punto de referencia arbitrario. La Figura 3.19 ilustra el cálculo del hipervolumen, donde r es el punto de referencia y los puntos p_i representan el valor en el espacio objetivo de los individuos de la población. Normalmente, el punto de referencia r es una estimación del peor valor que puede tomar un individuo en el espacio objetivo; en el caso de la Figura 3.19, se busca minimizar los objetivos f_1 y f_2; con lo que r se toma con valores altos en ambos objetivos, y cuanto más distribuidos y pequeños sean los valores de f_1 y f_2 para la población, mayor será el hipervolumen. Para estimar la convergencia del hipervolumen, un parámetro del algoritmo es la paciencia p, que mide el número de generaciones sin que el valor del hipervolumen se incremente en un valor h.

El Algoritmo 3.2 resume todos los pasos explicados, además de indicar cómo se tienen en cuenta las restricciones mencionadas de plausibilidad, y de que y_c sea la explicación más probable según el modelo ϕ para **x'**.

3.6.2. Ejemplo de uso

La implementación[6] de este algoritmo se realizó en Python, con ayuda del paquete *pymoo* (Blank y col., 2020). Para probar este algoritmo, se usaron los datos correspondientes a la variante Ómicron para construir una red Bayesiana semiparamétrica híbrida, utilizando la selección de variables correspondiente al algoritmo RF de la

[6] https://github.com/johacks/anexoTFM/blob/main/counterfactuals.ipynb

Algoritmo 3.2 Búsqueda de explicaciones contrafactuales.

Entrada:

- ϕ. Modelo predictivo.
- R. Red Bayesiana construida con variable clase y predictoras correspondientes a ϕ.
- MM. Función que indica el mínimo y máximo de cada variable.
- $\mathbf{x}$. Instancia con la que se ha hecho una predicción $y = \phi(\mathbf{x})$.
- y_c. Salida que se esperaba que produjera $\phi(\mathbf{x})$.
- $|\boldsymbol{P}|$. Tamaño de población.
- δ. Umbral de cambio mínimo en variables continuas.
- β. Umbral de plausibilidad mínima de las explicaciones contrafactuales producidas.
- p. Número de iteraciones máximas seguidas sin incrementar hipervolumen.
- B. Distribución del número de variables cambiadas en mutación, e.g. Boltzmann.

Procedimiento:

1. Inicializar población $\boldsymbol{P}$ con $|\boldsymbol{P}|$ copias de $\mathbf{x}$, y mutar cada $\mathbf{x'} \in \boldsymbol{P}$:
   - El número de variables mutadas se muestrea cada vez de B.
   - Para cada variable x'_i a mutar (seleccionadas al azar, en orden topológico), se muestrea el nuevo valor usando R con $P(X_i|\boldsymbol{Pa}_i)$, con los valores de $\mathbf{x'}$ en los padres.
2. Calcular el valor de los objetivos a optimizar f_i en cada $\mathbf{x'}$ y el hipervolumen correspondiente, así como el valor de las restricciones g_i:
   2.1. $f_1 = -1 \cdot P(y_c|\mathbf{x'};\phi)$, donde $P(y_c|\mathbf{x'};\phi)$ es fijado a 1 o 0 si ϕ no es probabilístico. Multiplicado por -1 para que sea a minimizar.
   2.2. $f_2 = D(\mathbf{x},\mathbf{x'})$.
   2.3. $g_1 = \beta - logP(\mathbf{x'}, y_c)$, $g_1 \leq 0$ no viola la restricción.
   2.4. $g_2 = \text{máx}_y\, P(y|\mathbf{x'};\phi) - P(y_c|\mathbf{x'};\phi)$, $g_2 \leq 0$ no viola la restricción.

   Nota. Cada f_i a minimizar toma el valor $f_i^{(\text{máx})} + \text{máx}(g_1, 0) + \text{máx}(g_2, 0)$ cuando $g_1 > 0$ o $g_2 > 0$, con $f_i^{(\text{máx})}$ el valor máximo posible de f_i, que son 0 y 1 para f_1 y f_2 respectivamente.
3. Mientras no se superen p iteraciones seguidas sin incrementar el hipervolumen de P **hacer**:
   3.1. Seleccionar y cruzar individuos de $\boldsymbol{P}$, con el criterio de selección original de Deb y col. (1995). Cruzar intercambiando el valor de una variable al azar.
   3.2. Mutar cada $\mathbf{x'} \in \boldsymbol{P}$ como en la inicialización de la población $\boldsymbol{P}$.
   3.3. Reparar cada $\mathbf{x'} \in \boldsymbol{P}$ modificando cada componente $x'_i \in \mathbf{x'}$.
      - Se mantiene cada variable en su rango según $MM(X_i)$.
      - Se hace $x'_i \leftarrow x_i$ si $d(x, x_i) < \delta$.
   3.4. Calcular el valor de los objetivos a optimizar f_i en cada $\mathbf{x'}$ y el hipervolumen correspondiente, así como el valor de las restricciones g_i.
4. **Devolver** soluciones no dominadas de P donde no se tenga $g_1 > 0$ o $g_2 > 0$.

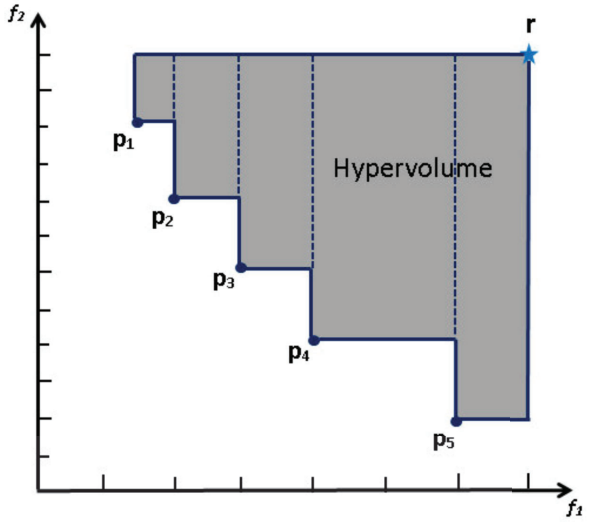


Figura 3.19: Ilustración del hipervolumen. Imagen del trabajo de Lwin y col. (2017).

Sección 3.2 (véase el listado en la Tabla B.4), que contenía la variable categórica de historial de enfermedades cardiovasculares. La estructura de la red G se aprendió con una búsqueda voraz que maximizaba la *cross-validated likelihood* con 10 particiones y los parámetros $\boldsymbol{\theta}$ mediante máxima verosimilitud. Se escogió un paciente cuya probabilidad de supervivencia se estimaba como 0.363 al hacer una predicción con la propia red Bayesiana semiparamétrica. El objetivo es encontrar explicaciones contrafactuales tales que la probabilidad de supervivencia aumente sin alejarse demasiado de las características originales.

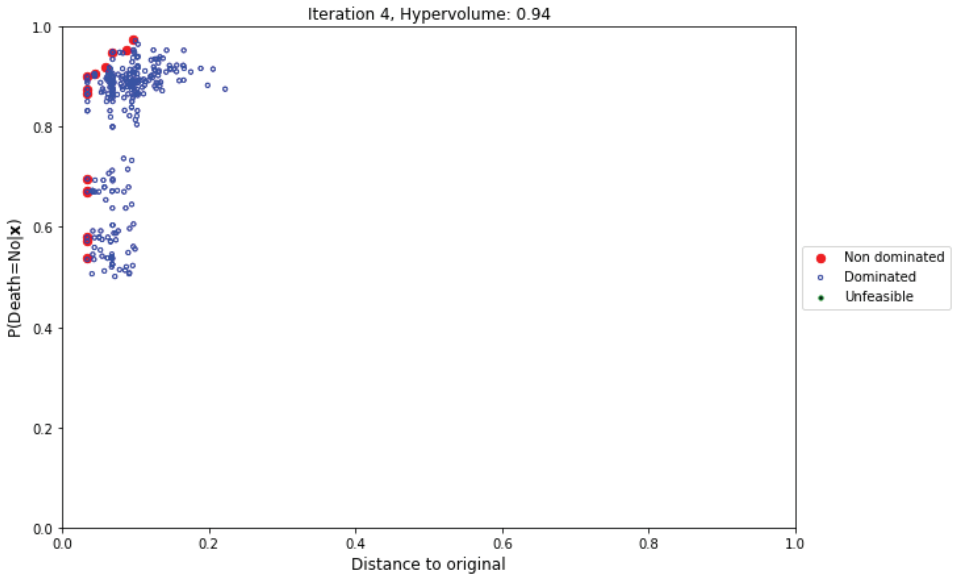
La red Bayesiana utilizada en el proceso de búsqueda del algoritmo fue la misma cuya predicción se buscaba explicar, sin embargo, podría haberse utilizado la red para buscar la explicación a la salida de cualquier modelo, como por ejemplo un RF. Se configuró la generación de la población inicial con 300 individuos. Para la condición de parada, se estableció una paciencia $p = 1$ con mínimo incremento $h = 0.01$. Se escogió $\delta = 0.2$ como umbral de cambio mínimo y $\beta = -77.04$, estimado como se explicó en párrafos anteriores. La búsqueda se realizó impidiendo el cambio de la variable edad, para observar los cambios en otras variables que fuerzan menos la salida. Pasadas 4 iteraciones y con un hipervolumen de 0.94 el algoritmo alcanzó la convergencia; el resultado del proceso se puede observar en la Figura 3.20.

Las explicaciones que incrementan menos la probabilidad de supervivencia, a cambio son más cercanas a la instancia original. Esto se ve en la Figura 3.20a, donde los pequeños cambios en la bilirrubina directa incrementan ligeramente la probabilidad de supervivencia, mientras que para tener probabilidades altas de supervivencia, se sugiere reducir significativamente los niveles de creatinina, aunque no como para llegar al intervalo de normalidad (véase Tabla A.1). También se puede observar comparando la explicación 11 con la 12 que incrementar la proteína C reactiva (aunque

manteniéndola en rangos normales) incrementaría la probabilidad de supervivencia bajo unas mismas condiciones del paciente según el modelo; esto llama la atención porque normalmente valores más altos indican mayor inflamación.

	Oxygen saturation (first measure)	Direct bilirubin	C-reactive protein	Lactate dehydrogenase (LDH)	Creatinine	Eosinophil %	Urea	Distance	P(Death=No|x)
Original	93.600	0.290	0.800	259.000	5.400	0.300	96.000	nan	0.363
0	93.600	0.050	0.800	259.000	5.400	0.300	96.000	0.034	0.538
1	93.600	0.000	0.800	259.000	5.400	0.300	96.000	0.034	0.572
2	93.600	0.290	0.800	259.000	5.400	1.224	96.000	0.034	0.579
3	93.600	0.290	0.800	259.000	5.400	1.568	96.000	0.034	0.670
4	93.600	0.290	0.800	259.000	5.400	1.575	96.000	0.034	0.672
5	93.600	0.290	0.800	259.000	5.400	1.776	96.000	0.034	0.695
6	93.600	0.290	0.800	259.000	2.325	0.300	96.000	0.034	0.867
7	93.600	0.290	0.800	259.000	2.300	0.300	96.000	0.034	0.875
8	93.600	0.290	0.800	259.000	1.915	0.300	96.000	0.034	0.901
9	93.600	0.290	0.800	259.000	1.980	0.300	100.627	0.044	0.905
10	93.600	0.207	0.800	259.000	2.165	0.300	96.000	0.059	0.921
11	93.600	0.000	0.800	259.000	1.805	0.300	96.000	0.069	0.949
12	93.600	0.000	3.384	259.000	1.805	0.300	96.000	0.088	0.953
13	92.473	0.290	0.800	237.903	1.964	1.575	96.000	0.097	0.974

(a) Tabla de explicaciones contrafactuales. El rojo indica decremento de la variable, y el azul el incremento. Solo se muestran variables que han llegado a modificarse.



(b) Estado final de la población en el espacio objetivo.

Figura 3.20: Resultados de aplicar algoritmo de búsqueda de contrafactuales.

Capítulo 4

Conclusiones y trabajo futuro

Uno de los primeros objetivos planteados para este trabajo fue analizar el impacto por olas de distintas variables en la mortalidad, tiempo de ingreso, y estancia en UCI/UCIR. De estas variables de interés, el enfoque se ha terminado poniendo en la mortalidad, puesto que en las últimas olas había muy pocos pacientes marcados como ingresados en la UCI/UCIR. El tiempo de ingreso se ha tenido en cuenta como objetivo secundario, incluyéndose en algunas de las actividades realizadas, como la obtención de redes Bayesianas discretas.

El hito principal en el logro de este objetivo es la implementación del *dashboard interactivo* para comparar la importancia por olas de las distintas variables según distintas metodologías, muestras de los pacientes, y codificación de los datos. El uso de diferentes métodos ha permitido comprobar que existe mucha redundancia entre variables, puesto que según el método elegido, puede verse incrementada o decrementada significativamente la importancia de una variable. Por ello, entre las tres metodologías de estimación de importancia disponibles, se ha asegurado que una identificara de forma individual la importancia de cada variable, para no desestimar la información contenida en una predictora al no ser identificada como relevante por métodos más complejos.

Mediante el uso del *dashboard*, ha sido fácil realizar observaciones como que de todas las variables inflamatorias, el conteo relativo de linfocitos ha sido la más estable en su utilidad como predictora, y más aún a medida que avanza la estancia del paciente. Se ha visto en cambio, que otras variables inflamatorias como la proteína C-reactiva tiene mayor importancia al inicio del ingreso del paciente que cuando ha transcurrido la primera semana.

Tener un *dashboard* de este tipo, ha permitido en definitiva, coger perspectiva en un conjunto de datos de gran complejidad, que incluye un gran número de variables, y codifica información dinámica de distintas duraciones y frecuencias para cada pa-

ciente. Teniendo en cuenta que no se han encontrado trabajos en el estado del arte con análisis por olas (aunque sí con datos abarcando varias olas (Aznar-Gimeno y col., 2021)), se considera que esta herramienta es una buena contribución para entender el cambio de la relación de las variables con la mortalidad del virus en sus distintas variantes. Por ello, se ha encapsulado en un contenedor de *Docker*[1] para facilitar un futuro despliegue en la web (ya puesto en marcha por un miembro del CIG-UPM), quedando a disposición de los clínicos interesados.

Otro de los objetivos planteados fue investigar el rendimiento de distintos modelos predictivos en cada ola. Nuevamente, se acabó restringiendo el enfoque únicamente a la variable de mortalidad del paciente, quedando para el trabajo futuro el estudio de las otras variables clase de interés. Esto implicó realizar el trabajo equivalente a varios trabajos del estado del arte, que comparaban el rendimiento en clasificación de distintos modelos en el ámbito del Covid-19, pero incrementando la complejidad al tener que hacerse el ejercicio para distintas agrupaciones de olas. Todo ello además, con la prueba de una rama nueva de clasificadores Bayesianos propuesta en este mismo trabajo, que en algunos casos, ha obtenido resultados comparables a otros de los más potentes del estado del arte. Para poder cumplir el objetivo eficientemente en esta combinación de olas y modelos, se ha automatizado en la medida de lo posible la resolución de problemáticas como la selección de hiperparámetros, utilizando una API común en los clasificadores.

De la realización de este objetivo, se ha concluido que si nos guiamos por el AUC, la regresión logística, siendo un modelo interpretable, fácil de obtener, y utilizado en el entorno médico, es un buen modelo a utilizar en olas futuras si se busca realizar predicciones de la mortalidad de los pacientes. Otra opción sería utilizar GBDT, que de forma general obtiene los mejores resultados en cuanto a AUC, pero el problema es que su opacidad impediría su uso en un ámbito clínico. Sin embargo, combinado con un algoritmo que explique la salida del modelo, por ejemplo con el algoritmo de búsqueda de explicaciones contrafactuales propuesto en la Sección 3.6, podría llegar a ser una alternativa más.

Hay que matizar que, aunque los modelos anteriores destacan en cuanto AUC, al observar los valores de la especificidad y sensibilidad, vemos que siempre tienen valores bajos en alguna de las dos; y la tasa de aciertos es poco informativa por el desbalance presente en la mortalidad como variable clase. Por tanto, antes de utilizar un modelo ϕ en clasificación, habría que buscar el umbral óptimo de clasificación α, tal que si $P(\texttt{Muerte=Sí}|\boldsymbol{x}^{(N+1)};\phi) > \alpha$, entonces la salida es `Muerte=Sí`, y `Muerte=No` en caso contrario, siendo $\boldsymbol{x}^{(N+1)}$ una nueva instancia a clasificar.

Normalmente se buscaría el valor de α que optimice una métrica M de clasificación promediada en una validación cruzada. En trabajos pasados de este proyecto,

[1] `https://www.docker.com/`

miembros del equipo médico indicaron que $M(\phi) = 1/4 \cdot \texttt{Especificidad}(\phi) + 3/4 \cdot \texttt{Sensibilidad}(\phi)$ sería una buena métrica del rendimiento de un modelo ϕ para la clasificación de futuros pacientes. Queda para el trabajo futuro repetir la evaluación del rendimiento de clasificadores por olas incluyendo una estimación del máximo valor alcanzable del mencionado M tras la selección óptima de α, que serviría para precisar la validez de cada modelo en cuanto a predicción de mortalidad.

Finalmente, para realizar un estudio de la distribución de variables en cada ola y búsqueda de factores de riesgo para la mortalidad en las dos variantes más recientes, se han desarrollado dos redes Bayesianas discretas, con las que se puede interactuar en el *software* GeNIe. Dichas redes permiten realizar inferencias y visualizar el flujo de la información más fácilmente, reduciendo el riesgo de llegar a conclusiones erróneas por problemas como el sesgo de selección. Entre otras conclusiones, se ha visto que la variable de vacunación no tenía un impacto en la mortalidad para pacientes de mismos perfiles de edades. Para ambas variantes, edades superiores a 65 años incrementaban significativamente la probabilidad de fallecer. En este perfil de edades, para la variante Delta, son de riesgo los pacientes con valores altos de RDW, ritmo cardíaco alto, o valores bajos de la hemoglobina. En Ómicron, son de mayor riesgo los pacientes con valores altos de temperatura, valores bajos o altos de la cuenta de linfocitos, o valores altos de INR.

Cabe mencionar que al emplear la red Bayesiana discreta de Delta para la predicción de los datos correspondientes a la variante Ómicron (combinación de entrenamiento y validación, pero no test), se obtuvo un AUC de 0.74, comparable al de la propia red de Ómicron. Queda como trabajo futuro profundizar más en las implicaciones de este resultado.

También se ha visto que la discretización proporcionada por el equipo médico puede perder mucha información de algunas variables, como por ejemplo la proteína C-reactiva. Por tanto, una posible línea de trabajo futuro a seguir es la mejora de la discretización de las variables con mayor granularidad, permitiendo así un análisis más preciso de la interacción entre variables. Otra posible línea de investigación sería la construcción de redes Bayesianas discretas con datos de todas las olas, con el enfoque de obtener estructuras que permitan entender el cambio de la distribución de las variables con las sucesivas variantes. Finalmente, habría que investigar la capacidad de los distintos modelos predictivos de realizar predicciones de la mortalidad y otras variables de interés, habiendo sido entrenados con datos de la ola anterior.

Bibliografía

Alderisi, S. (2020). Machine learning applied to COVID-19 [Tesis de maestría, Universidad Politécnica de Madrid.]

Allison, P. D. (2001). *Missing Data*. Sage publications.

Altmann, A., Toloşi, L., Sander, O. y Lengauer, T. (2010). Permutation importance: A corrected feature importance measure. En: *Bioinformatics* 26.10, págs. 1340-1347.

Armañanzas, R., Díaz, A., Martínez-García, M. y Mazuelas, S. (2021). Derivation of a cost-sensitive COVID-19 mortality risk indicator using a multistart framework. En: *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, págs. 2179-2186.

Atienza, D., Larrañaga, P. y Bielza, C. (2022). Hybrid semiparametric Bayesian networks. En: *TEST* 31.2, págs. 299-327. ISSN: 1863-8260.

Aznar-Gimeno, R., Esteban, L., Labata-Lezaun, G., Del-Hoyo-alonso, R., Abadia-Gallego, D., Paño-Pardo, J., Esquillor-Rodrigo, M., Lanas, Á. y Serrano, M. (2021). A clinical decision web to predict ICU admission or death for patients hospitalised with COVID-19 using machine learning algorithms. En: *International Journal of Environmental Research and Public Health* 18.16.

Bahdanau, D., Cho, K. y Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. En: *arXiv preprint arXiv:1409.0473*.

Bellazzi, R. y Zupan, B. (2008). Predictive data mining in clinical medicine: Current issues and guidelines. En: *International Journal of Medical Informatics* 77.2, págs. 81-97.

Berkson, J. (1944). Application of the logistic function to bio-assay. En: *Journal of the American Statistical Association* 39.227, págs. 357-365.

Bernaola, N., Lima, G. de, Riaño, M. A., Llanos, L., Heili-Frades, S., Sanchez, O., Lara, A., Plaza, G., Carballo, C., Gallego, P., Larrañaga, P. y Bielza, C. (2022). Decision trees for COVID-19 prognosis learned from patient data: Desaturating the ER with Artificial Intelligence. En: *medRxiv*.

Bielza, C. y Larranaga, P. (2014). Discrete Bayesian network classifiers: A survey. En: *ACM Computing Surveys (CSUR)* 47.1, págs. 1-43.

Blank, J. y Deb, K. (2020). Pymoo: Multi-objective optimization in python. En: *IEEE Access* 8, págs. 89497-89509.

Bolt, M. A., MaWhinney, S., Pattee, J. W., Erlandson, K. M., Badesch, D. B. y Peterson, R. A. (2022). Inference following multiple imputation for generalized additive models: An investigation of the median p-value rule with applications to the Pulmonary Hypertension Association Registry and Colorado COVID-19 hospitalization data. En: *BMC Medical Research Methodology* 22.1, págs. 1-14.

Bottino, F., Tagliente, E., Pasquini, L., Napoli, A. D., Lucignani, M., Figà-Talamanca, L. y Napolitano, A. (2021). COVID mortality prediction with machine learning methods: A systematic review and critical appraisal. En: *Journal of Personalized Medicine* 11.9, pág. 893.

Breiman, L. (2001). Random forests. En: *Machine Learning* 45.1, págs. 5-32.

Breiman, L., Friedman, J. H., Olshen, R. A. y Stone, C. J. (2017). *Classification and Regression Trees*. Routledge.

Burballa, C., Crespo, M., Redondo-Pachón, D., Pérez-Sáez, M. J., Mir, M., Arias-Cabrales, C., Francés, A., Fumadó, L., Cecchini, L. y Pascual, J. (2018). MDRD or CKD-EPI for glomerular filtration rate estimation in living kidney donors. En: *Nefrología (English Edition)* 38.2, págs. 207-212.

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H. y Chen, K. (2015). XGBoost: Extreme gradient boosting. En: *R package version 0.4-2* 1.4, págs. 1-4.

Chen, Y.-C., Wheeler, T. A. y Kochenderfer, M. J. (2017). Learning discrete Bayesian networks from continuous data. En: *Journal of Artificial Intelligence Research* 59, págs. 103-132.

Chow, C. y Liu, C. (1968). Approximating discrete probability distributions with dependence trees. En: *IEEE Transactions on Information Theory* 14.3, págs. 462-467.

Cooper, G. F. y Yoo, C. (2013). Causal discovery from a mixture of experimental and observational data. En: *arXiv preprint arXiv:1301.6686*.

Dandl, S., Molnar, C., Binder, M. y Bischl, B. (2020). Multi-objective counterfactual explanations. En: *International Conference on Parallel Problem Solving from Nature*. Springer, págs. 448-469.

Dash, R., Paramguru, R. L. y Dash, R. (2011). Comparative analysis of supervised and unsupervised discretization techniques. En: *International Journal of Advances in Science and Technology* 2.3, págs. 29-37.

Deb, K. y Agrawal, R. (1995). Simulated binary crossover for continuous search space. En: *Complex Systems* 9.2, págs. 115-148.

Deb, K., Pratap, A., Agarwal, S. y Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. En: *IEEE Transactions on Evolutionary Computation* 6.2, págs. 182-197.

Defazio, A., Bach, F. y Lacoste-Julien, S. (2014). SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. En: *Advances in Neural Information Processing Systems* 27.

Dempster, A. P., Laird, N. M. y Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. En: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1, págs. 1-22.

Druzdzel, M. J. (1999). SMILE: Structural Modeling, Inference, and Learning Engine and GeNIe: A development environment for graphical decision-theoretic models. En: *Association for the Advancement of Artificial Intelligence*, págs. 902-903.

Edmonds, J. (1967). Optimum branchings. En: *Journal of Research of the National Bureau of Standards B* 71.4, págs. 233-240.

Feng, S., Hategeka, C. y Grépin, K. A. (2021). Addressing missing values in routine health information system data: An evaluation of imputation methods using data from the Democratic Republic of the Congo during the COVID-19 pandemic. En: *Population Health Metrics* 19.1, págs. 1-14.

Fenton, N. E. (2020). A note on 'Collider bias undermines our understanding of COVID-19 disease risk and severity' and how causal Bayesian networks both expose and resolve the problem. En: *arXiv preprint arXiv:2005.08608*.

Fenton, N. E., McLachlan, S., Lucas, P., Dube, K., Hitman, G. A., Osman, M., Kyrimi, E. y Neil, M. (2021). A Bayesian network model for personalised COVID19 risk assessment and contact tracing. En: *medRxiv*.

Fenton, N. E., Neil, M., Osman, M. y McLachlan, S. (2020). COVID-19 infection and death rates: The need to incorporate causal explanations for the data and avoid bias in testing. En: *Journal of Risk Research* 23.7-8, págs. 862-865.

Fernández, R. R., De Diego, I. M., Aceña, V., Fernández-Isabel, A. y Moguerza, J. M. (2020). Random forest explainability using counterfactual sets. En: *Information Fusion* 63, págs. 196-207.

Ferrao, J. C., Oliveira, M. D., Janela, F. y Martins, H. M. (2016). Preprocessing structured clinical data for predictive modeling and decision support. En: *Applied Clinical Informatics* 7.04, págs. 1135-1153.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. En: *Annals of Statistics*, págs. 1189-1232.

Friedman, N., Geiger, D. y Goldszmidt, M. (1997). Bayesian network classifiers. En: *Machine Learning* 29.2, págs. 131-163.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. En: *Computers & Operations Research* 13.5, págs. 533-549.

Glymour, C., Zhang, K. y Spirtes, P. (2019). Review of causal discovery methods based on graphical models. En: *Frontiers in Genetics* 10, pág. 524.

Guyon, I., Weston, J., Barnhill, S. y Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. En: *Machine Learning* 46.1, págs. 389-422.

Hothorn, T. y Zeileis, A. (2017). Transformation forests. En: *arXiv preprint arXiv:1701.02110*.

Hutchinson, G. E. (1957). Concluding Remarks. En: *Population Studies: Animal Ecology and Demography. Cold Spring Harbor Symposia on Quantitative Biology* 22, pág. 416.

Huyut, M. T. y Üstündağ, H. (2022). Prediction of diagnosis and prognosis of COVID-19 disease by blood gas parameters using decision trees machine learning model: A retrospective observational study. En: *Medical Gas Research* 12.2, pág. 60.

Iannone, R., Allaire, J. y Borges, B. (2018). flexdashboard: R markdown format for flexible dashboards. En: *R package version 0.5* 1.

Jia, Y., McDermid, J. y Habli, I. (2021). Enhancing the value of counterfactual explanations for deep learning. En: *International Conference on Artificial Intelligence in Medicine*, págs. 389-394.

Kirkpatrick, S., Gelatt Jr, C. D. y Vecchi, M. P. (1983). Optimization by simulated annealing. En: *Science* 220.4598, págs. 671-680.

Ko, H., Chung, H., Kang, W., Park, C., Kim, D., Kim, S., Chung, C., Ko, R., Lee, H., Seo, J., Choi, T.-Y., Jaimes, R., Kim, K. y Lee, J. (2020). An artificial intelligence model to predict the mortality of COVID-19 patients at hospital admission time using routine blood samples: Development and validation of an ensemble model. En: *Journal of Medical Internet Research* 22.12, e25442.

Koller, D. y Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

Kovalev, M., Utkin, L., Coolen, F. y Konstantinov, A. (2021). Counterfactual explanation of machine learning survival models. En: *Informatica* 32.4, págs. 817-847.

Kuhn, M. (2008). Building predictive models in R using the caret package. En: *Journal of Statistical Software* 28, págs. 1-26.

Li, S., Lin, Y., Zhu, T., Fan, M., Xu, S., Qiu, W., Chen, C., Li, L., Wang, Y., Yan, J., Wong, J., Naing, L. y Xu, S. (2021). Development and external evaluation of predictions models for mortality of COVID-19 patients using machine learning method. En: *Neural Computing and Applications*, págs. 1-10.

Lucic, A., Ter Hoeve, M. A., Tolomei, G., De Rijke, M. y Silvestri, F. (2022). Cf-gnnexplainer: Counterfactual explanations for graph neural networks. En: *International Conference on Artificial Intelligence and Statistics*. PMLR, págs. 4499-4511.

Lundberg, S. M. y Lee, S.-I. (2017). A unified approach to interpreting model predictions. En: *Advances in Neural Information Processing Systems* 30.

Lwin, K. T., Qu, R. y MacCarthy, B. L. (2017). Mean-VaR portfolio optimization: A nonparametric approach. En: *European Journal of Operational Research* 260.2, págs. 751-766.

Marchese Robinson, R. L., Palczewska, A., Palczewski, J. y Kidley, N. (2017). Comparison of the predictive performance and interpretability of random forest and linear models on benchmark data sets. En: *Journal of Chemical Information and Modeling* 57.8, págs. 1773-1792.

Marcot, B. G. y Penman, T. D. (2019). Advances in Bayesian network modelling: Integration of modelling technologies. En: *Environmental Modelling & Software* 111, págs. 386-393.

Meganck, S., Leray, P. y Manderick, B. (2006). Learning causal Bayesian networks from observations and experiments: A decision theoretic approach. En: *International Conference on Modeling Decisions for Artificial Intelligence*. Springer, págs. 58-69.

Menze, B. H., Kelm, B. M., Masuch, R., Himmelreich, U., Bachert, P., Petrich, W. y Hamprecht, F. A. (2009). A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data. En: *BMC Bioinformatics* 10.1, págs. 1-16.

Mihaljevič, B., Bielza, C. y Larrañaga, P. (2019). bnclassify: Learning Bayesian network classifiers. En: *R Journal* 10.2, págs. 455-468.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. y Dean, J. (2013). Distributed representations of words and phrases and their compositionality. En: *Advances in Neural Information Processing Systems* 26.

Minsky, M. (1961). Steps toward artificial intelligence. En: *Proceedings of the IRE* 49.1, págs. 8-30.

Molnar, C. (2022). *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. 2.ª ed.

Nelder, J. A. y Mead, R. (1965). A simplex method for function minimization. En: *The Computer Journal* 7.4, págs. 308-313.

Pazzani, M. y Billsus, D. (1997). Learning and revising user profiles: The identification of interesting web sites. En: *Machine learning* 27.3, págs. 313-331.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R. y Dubourg, V. (2011). Scikit-learn: Machine learning in Python. En: *The Journal of Machine Learning Research* 12, págs. 2825-2830.

Quinlan, J. R. (1987). Decision trees as probabilistic classifiers. En: *Proceedings of the Fourth International Workshop on Machine Learning*. Elsevier, págs. 31-37.

Ramírez-Gallego, S., García, S., Benítez, J. M. y Herrera, F. (2015). Multivariate discretization based on evolutionary cut points selection for classification. En: *IEEE Transactions on Cybernetics* 46.3, págs. 595-608.

Rancoita, P. M., Zaffalon, M., Zucca, E., Bertoni, F. y De Campos, C. P. (2016). Bayesian network data imputation with application to survival tree analysis. En: *Computational Statistics & Data Analysis* 93, págs. 373-387.

Riaño, M. A. (2021). Avances en árboles de decisión y su aplicación para clasificar enfermos críticos de COVID-19 [Tesis de maestría, Universidad Politécnica de Madrid.

Ritchie, H., Mathieu, E., Rodés-Guirao, L., Appel, C., Giattino, C., Ortiz-Ospina, E., Hasell, J., Macdonald, B., Beltekian, D. y Roser, M. (2022). Coronavirus Pandemic (COVID-19). En: *Our World in Data*. https://ourworldindata.org/coronavirus.

Ross, B. C. (2014). Mutual information between discrete and continuous data sets. En: *PLOS ONE* 9.2, págs. 1-5.

Rubin, D. B. (1976). Inference and missing data. En: *Biometrika* 63.3, págs. 581-592.

Rubin, D. B. y Schenker, N. (1991). Multiple imputation in health care databases: An overview and some applications. En: *Statistics in Medicine* 10.4, págs. 585-598.

Ruggieri, A., Stranieri, F., Stella, F. y Scutari, M. (2020). Hard and soft EM in Bayesian network learning from incomplete data. En: *Algorithms* 13.12, pág. 329.

Sahami, M. (1996). Learning limited dependence Bayesian classifiers. En: págs. 335-338.

Sánchez-Montañés, M., Rodríguez-Belenguer, P., Serrano-López, A. J., Soria-Olivas, E. y Alakhdar-Mohmara, Y. (2020). Machine learning for mortality analysis in patients with COVID-19. En: *International Journal of Environmental Research and Public Health* 17.22, pág. 8386.

Schirato, L., Makina, K., Flanders, D., Pouriyeh, S. y Shahriar, H. (2021). COVID-19 mortality prediction using machine learning techniques. En: *2021 IEEE International Conference on Digital Health (ICDH)*. IEEE, págs. 197-202.

Scott, D. W. (2015). *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley & Sons.

Scutari, M. (2009). Learning Bayesian networks with the bnlearn R package. En: *arXiv preprint arXiv:0908.3817*.

Shanbehzadeh, M., Orooji, A. y Kazemi-Arpanahi, H. (2021). Comparing of data mining techniques for predicting in-hospital mortality among patients with covid-19. En: *Journal of Biostatistics and Epidemiology* 7.2, págs. 154-173.

Silander, T., Roos, T., Kontkanen, P. y Myllymäki, P. (2008). Factorized normalized maximum likelihood criterion for learning Bayesian network structures. En: *Proceedings of the 4th European Workshop on Probabilistic Graphical Models (PGM-08)*, págs. 257-272.

Snoek, J., Larochelle, H. y Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. En: *Advances in Neural Information Processing Systems* 25, págs. 2951-2959.

Syswerda, G. (1989). Uniform crossover in genetic algorithms. En: *Proceedings of the Third International Conference on Genetic Algorithms*, págs. 2-9.

Talvitie, T., Eggeling, R. y Koivisto, M. (2019). Learning Bayesian networks with local structure, mixed variables, and exact algorithms. En: *International Journal of Approximate Reasoning* 115, págs. 69-95.

Tsirtsis, S., De, A. y Rodriguez, M. (2021). Counterfactual explanations in sequential decision making under uncertainty. En: *Advances in Neural Information Processing Systems* 34, págs. 30127-30139.

Vaid, A., Somani, S., Russak, A., Freitas, J. de, Chaudhry, F., Paranjpe, I., Johnson, K., Lee, S., Miotto, R., Richter, F., Zhao, S., Beckmann, N., Naik, N., Kia, A., Timsina, P., Lala, A., Paranjpe, M., Golden, E., Danieletto, M., Singh, M., Meyer, D., O'Reilly, P., Huckins, L., Kovatch, P., Finkelstein, J., Freeman, R., Argulian, E., Kasarskis, A., Percha, B., Aberg, J., Bagiella, E., Horowitz, C., Murphy, B., Nestler, E., Schadt, E., Cho, J., Cordon-Cardo, C., Fuster, V., Charney, D., Reich, D., Bottinger, E., Levin, M., Narula, J., Fayad, Z., Just, A., Charney, A., Nadkarni, G. y Glicksberg, B. (2020). Machine learning to predict mortality and critical events in a cohort of patients with COVID-19 in New York City: Model development and validation. En: *Journal of Medical Internet Research* 22.11, e24018.

Van Buuren, S. (2011). Multiple imputation of multilevel data. En: *Handbook of Advanced Multilevel Analysis*. Routledge, págs. 181-204.

— (2018). *Flexible Imputation of Missing Data*. CRC Press.

Van Buuren, S. y Groothuis-Oudshoorn, K. (2011). MICE: Multivariate imputation by chained equations in R. En: *Journal of Statistical Software* 45, págs. 1-67.

Vepa, A., Saleem, A., Rakhshan, K., Daneshkhah, A., Sedighi, T., Shohaimi, S., Omar, A., Salari, N., Chatrabgoun, O., Dharmaraj, D., Sami, J., Parekh, S., Ibrahim, M., Raza, M., Kapila, P. y Chakrabarti, P. (2021). Using machine learning algorithms to develop a clinical decision-making tool for COVID-19 inpatients. En: *International Journal of Environmental Research and Public Health* 18.12, pág. 6228.

Verma, S., Dickerson, J. e Hines, K. (2020). Counterfactual explanations for machine learning: A review. En: *arXiv preprint arXiv:2010.10596*.

Wachter, S., Mittelstadt, B. y Russell, C. (2017). Counterfactual explanations without opening the black box: Automated decisions and the GDPR. En: *Harv. JL & Tech.* 31, pág. 841.

Wang, J., Yu, H., Hua, Q., Jing, S., Liu, Z., Peng, X., Cao, C. y Luo, Y. (2020). A descriptive study of random forest algorithm for predicting COVID-19 patients outcome. En: *PeerJ* 8.

Wang, Z., Samsten, I. y Papapetrou, P. (2021). Counterfactual explanations for survival prediction of cardiovascular ICU patients. En: *International Conference on Artificial Intelligence in Medicine*. Springer, págs. 338-348.

Wu, X., Liu, X., Zhou, Y., Yu, H., Li, R., Zhan, Q., Ni, F., Fang, S., Lu, Y., Ding, X., Liu, H., Ewing, R. M., Jones, M. G., Hu, Y., Nie, H. y Wang, Y. (2021). 3-month, 6-month, 9-month, and 12-month respiratory outcomes in patients following COVID-19-related hospitalisation: A prospective study. En: *The Lancet Respiratory Medicine* 9.7, págs. 747-754. ISSN: 2213-2600.

Wynants, L., Van Calster, B., Collins, G., Riley, R., Heinze, G., Schuit, E., Bonten, M., Damen, J., Debray, T., De Vos, M., Dhiman, P., Haller, M., Harhay, M., Henckaerts, L., Kreuzberger, N., Lohmann, A., Luijken, K., Ma, J., Andaur Navarro, C., Reitsma, J., Sergeant, J., Shi, C., Skoetz, N., Smits, L., Snell, K., Sperrin, M., Spijker, R., Steyerberg, E., Takada, T., Van Kuijk, S., Van Royen, F., Wallisch, C.,

Hooft, L., Moons, K. y Van Smeden, M. (2020). Prediction models for diagnosis and prognosis of covid-19: Systematic review and critical appraisal. En: *The BMJ* 369.

Zhang, S. (2012). Nearest neighbor selection for iteratively kNN imputation. En: *Journal of Systems and Software* 85.11, págs. 2541-2552.

Zhu, J. S., Ge, P., Jiang, C., Zhang, Y., Li, X., Zhao, Z., Zhang, L. y Duong, T. Q. (2020). Deep-learning artificial intelligence analysis of clinical variables predicts mortality in COVID-19 patients. En: *Journal of the American College of Emergency Physicians Open* 1.6, págs. 1364-1373.

Apéndice A

Variables candidatas para predicción de mortalidad en Covid-19

Variable	Category	[Sex (Age)] Normal results	Units
Age	Profile	Vitals and lab tests only	none
BMI	Profile	Vitals and lab tests only	none
Sex	Profile	Vitals and lab tests only	none
Vaccinated (>=1 dose)	Profile	Vitals and lab tests only	none
Arterial hypertension disease history	Comorbilities	Vitals and lab tests only	none
Cancer disease history	Comorbilities	Vitals and lab tests only	none
Cardiac disease history	Comorbilities	Vitals and lab tests only	none
Diabetic	Comorbilities	Vitals and lab tests only	none
Lung disease history	Comorbilities	Vitals and lab tests only	none
Neurological disease history	Comorbilities	Vitals and lab tests only	none
Renal disease history	Comorbilities	Vitals and lab tests only	none
Smoker	Comorbilities	Vitals and lab tests only	none
Body temperature (daily minimum)	Vitals	[X ($-\infty$,∞)] (35,38)	°C
Body temperature (first measure)	Vitals	[X ($-\infty$,∞)] (35,38)	°C
Body temperaure (daily maximum)	Vitals	[X ($-\infty$,∞)] (35,38)	°C
Diastolic blood pressure (daily maximum)	Vitals	[X ($-\infty$,∞)] (40,60)	mmHg
Diastolic blood pressure (daily minimum)	Vitals	[X ($-\infty$,∞)] (40,60)	mmHg
Diastolic blood pressure (first measure)	Vitals	[X ($-\infty$,∞)] (40,60)	mmHg
Heart rate (daily maximum)	Vitals	[X ($-\infty$,∞)] (50,100)	bpm
Heart rate (daily minimum)	Vitals	[X ($-\infty$,∞)] (50,100)	bpm
Heart rate (first measure)	Vitals	[X ($-\infty$,∞)] (50,100)	bpm
Oxygen saturation (daily maximum)	Vitals	[X ($-\infty$,∞)] (95,100)	%
Oxygen saturation (daily minimum)	Vitals	[X ($-\infty$,∞)] (95,100)	%
Oxygen saturation (first measure)	Vitals	[X ($-\infty$,∞)] (95,100)	%
Systolic blood pressure (daily maximum)	Vitals	[X ($-\infty$,∞)] (100,140)	mmHg
Systolic blood pressure (daily minimum)	Vitals	[X ($-\infty$,∞)] (100,140)	mmHg
Systolic blood pressure (first measure)	Vitals	[X ($-\infty$,∞)] (100,140)	mmHg
Activated Partial Thromboplastin Time (aPTT)	Lab tests	[X ($-\infty$,∞)] (22.5,36)	s
Alanine transaminase (ALT)	Lab tests	[M ($-\infty$,∞)] (0,41) [F ($-\infty$,∞)] (0,35)	UI/L
Albumin	Lab tests	[X ($-\infty$,∞)] (3.5,5.2)	g/dL
Aspartate transaminase (AST)	Lab tests	[M ($-\infty$,∞)] (0,40) [F ($-\infty$,∞)] (0,32)	U/L
Basophil count	Lab tests	[X ($-\infty$,∞)] (0,0.3)	x 10^3 μL
Basophils %	Lab tests	[X ($-\infty$,∞)] (0,2)	%
Blood gas test pH	Lab tests	[X ($-\infty$,∞)] (7.35,7.45)	none
Blood urea nitrogen (BUN)	Lab tests	[X ($-\infty$,∞)] (8,23)	mg/dL
C-reactive protein	Lab tests	[X ($-\infty$,∞)] (0,0.5)	mg/dL
Calcium	Lab tests	[X ($-\infty$,60)] (8.6,10) [X (60,90)] (8.8,10.2) [X (90, ∞)] (8.2,9.6)	mg/dL
Creatinine	Lab tests	[M ($-\infty$,∞)] (0.67,1.17) [F ($-\infty$,∞)] (0.51,0.95)	mg/dL
Current bicarbonate (blood gas test)	Lab tests	[X ($-\infty$,∞)] (26,32)	mmol/L
D-Dimer	Lab tests	[X ($-\infty$,∞)] (68,494)	μg/mL
Derived fibrinogen	Lab tests	[X ($-\infty$,∞)] (200,400)	mg/dL
Direct bilirubin	Lab tests	[X ($-\infty$,∞)] (0,0.2)	mg/dL
Eosinophil %	Lab tests	[X ($-\infty$,∞)] (1,5)	%
Eosinophil count	Lab tests	[X ($-\infty$,∞)] (0,0.5)	x 10^3 μL
Estimated glomerular filtration rate (eGFR) ckd-epi	Lab tests	[X ($-\infty$,∞)] (60, ∞)	mL/min/1.73m²
Ferritin	Lab tests	[M ($-\infty$,∞)] (17.9,464) [F ($-\infty$,50)] (6.24,137) [F (50, ∞)] (11.1,264)	ng/mL
Gamma-glutamyltransferase (GGT)	Lab tests	[X ($-\infty$,∞)] (0,40)	IU/L
Glucose	Lab tests	[X ($-\infty$,∞)] (74,109)	mg/dL
Hematocrit	Lab tests	[M ($-\infty$,∞)] (39,50) [F ($-\infty$,∞)] (36,43)	%
Hemoglobin	Lab tests	[M ($-\infty$,∞)] (13,17) [F ($-\infty$,∞)] (12,15)	g/dL
Hemolysis index	Lab tests	[X ($-\infty$,∞)] (0,6)	μU/mL
International normalized ratio (INR)	Lab tests	[X ($-\infty$,∞)] (0.85,1.2)	none
Lactate dehydrogenase (LDH)	Lab tests	[X ($-\infty$,∞)] (120,246)	UI/L
Leukocytes count	Lab tests	[X ($-\infty$,∞)] (3.5,12)	x 10^3 μL
Lymphocyte %	Lab tests	[X ($-\infty$,∞)] (20,45)	%
Lymphocyte count	Lab tests	[X ($-\infty$,∞)] (1.2,5)	x 10^3 μL
Mean corpuscular hemoglobin	Lab tests	[X ($-\infty$,∞)] (27,32)	pg
Mean corpuscular hemoglobin concentration (MCHC)	Lab tests	[X ($-\infty$,∞)] (31.5,34.5)	g/dL
Mean corpuscular volume	Lab tests	[X ($-\infty$,∞)] (80,100)	fL
Mean platelet volume	Lab tests	[X ($-\infty$,∞)] (9,13)	fL
Monocyte count	Lab tests	[X ($-\infty$,∞)] (2,10)	x 10^3 μL
Monocytes %	Lab tests	[X ($-\infty$,∞)] (2,10)	%
Neutrophil count	Lab tests	[X ($-\infty$,∞)] (1.7,8)	x 10^3 μL
Partial pressure of CO2 (Blood gas test)	Lab tests	[X ($-\infty$,∞)] (41,51)	mmHg
Partial pressure of oxygen (Blood gas test)	Lab tests	[X ($-\infty$,∞)] (24,40)	mmHg
Partial Thromboplastin Time ratio	Lab tests	[X ($-\infty$,∞)] (0.8,1.3)	none
Platelets	Lab tests	[X ($-\infty$,∞)] (150,450)	x 10^3 μL
Potassium	Lab tests	[X ($-\infty$,∞)] (3.5,5.1)	mmol/L
Prothrombin time (PT)	Lab tests	[X ($-\infty$,∞)] (10,14)	s
Prothrombin Time (Quick)	Lab tests	[X ($-\infty$,∞)] (70,130)	%
Red blood cells	Lab tests	[M ($-\infty$,∞)] (4.3,5.9) [F ($-\infty$,∞)] (3.5,5.8)	x 10^6 dL
Red Cell Blood Distribution Width (RDW)	Lab tests	[X ($-\infty$,∞)] (11.2,15.2)	%
Segmented neutrophils %	Lab tests	[X ($-\infty$,∞)] (40,75)	%
Sodium	Lab tests	[X ($-\infty$,∞)] (136,145)	mmol/L
Total bilirubin	Lab tests	[X ($-\infty$,∞)] (0.3,1.2)	mg/dL
Total CO2 (blood gas test)	Lab tests	[X ($-\infty$,∞)] (23,29)	mmol/L
Urea	Lab tests	[X ($-\infty$,∞)] (17,49)	mg/dl

Tabla A.1: Variables candidatas para predicción de mortalidad en Covid-19. Se indica para las variables de laboratorio los intervalos de referencia, con el sexo y rango de edad asociado si lo hay.

Apéndice B

Selección de variables con *recursive feature elimination* (RFE)

	Method[preprocessing] (AUC +/- AUC std. dev.)			
	GBM (0.91 +/- 0.023)	**RF (0.91 +/- 0.024)**	**GBM[discretized] (0.87 +/- 0.030)**	**RF[discretized] (0.86 +/- 0.032)**
1	Age	Age	Age	Age
2	Albumin	Albumin	Albumin	Albumin
3	Blood urea nitrogen (BUN)	Blood urea nitrogen (BUN)	Aspartate transaminase (AST)	Blood urea nitrogen (BUN)
4	Body temperature (daily minimum)	Body temperature (daily minimum)	Blood gas test pH	Cancer disease history
5	Body temperaure (daily maximum)	Body temperaure (daily maximum)	Blood urea nitrogen (BUN)	Cardiac disease history
6	Calcium	Calcium	Body temperaure (daily maximum)	Creatinine
7	Diastolic blood pressure (daily maximum)	Creatinine	Cancer disease history	D-Dimer
8	Diastolic blood pressure (daily minimum)	Eosinophil %	Cardiac disease history	Eosinophil %
9	Eosinophil %	Estimated glomerular filtration rate (eGFR) ckd-epi	Creatinine	Glucose
10	Ferritin	Glucose	Eosinophil %	Heart rate (daily maximum)
11	Glucose	Heart rate (daily maximum)	Glucose	Hematocrit
12	Heart rate (daily maximum)	Hematocrit	Heart rate (daily maximum)	Hemoglobin
13	Heart rate (daily minimum)	Hemoglobin	Lactate dehydrogenase (LDH)	Lactate dehydrogenase (LDH)
14	Hematocrit	International normalized ratio (INR)	Leukocytes count	Leukocytes count
15	Hemoglobin	Lactate dehydrogenase (LDH)	Mean corpuscular hemoglobin concentration (MCHC)	Lymphocyte %
16	International normalized ratio (INR)	Lymphocyte %	Mean corpuscular volume	Lymphocyte count
17	Lactate dehydrogenase (LDH)	Lymphocyte count	Monocytes %	Mean corpuscular hemoglobin concentration (MCHC)
18	Lymphocyte %	Mean corpuscular hemoglobin concentration (MCHC)	Neutrophil count	Neutrophil count
19	Lymphocyte count	Mean corpuscular volume	Oxygen saturation (daily maximum)	Oxygen saturation (daily maximum)
20	Mean corpuscular hemoglobin concentration (MCHC)	Oxygen saturation (daily maximum)	Platelets	Partial Thromboplastin Time ratio
21	Mean corpuscular volume	Oxygen saturation (daily minimum)	Prothrombin Time (Quick)	Platelets
22	Oxygen saturation (daily maximum)	Platelets	Red blood cells	Prothrombin time (PT)
23	Oxygen saturation (daily minimum)	Prothrombin time (PT)	Red Cell Blood Distribution Width (RDW)	Prothrombin Time (Quick)
24	Partial pressure of oxygen (Blood gas test)	Prothrombin Time (Quick)	Segmented neutrophils %	Red blood cells
25	Platelets	Red blood cells	Sodium	Red Cell Blood Distribution Width (RDW)
26	Prothrombin time (PT)	Red Cell Blood Distribution Width (RDW)	Systolic blood pressure (first measure)	Segmented neutrophils %
27	Red blood cells	Segmented neutrophils %	Total CO2 (blood gas test)	Sodium
28	Red Cell Blood Distribution Width (RDW)	Urea	Urea	Total CO2 (blood gas test)
29	Systolic blood pressure (daily maximum)			Urea
30	Urea			

Tabla B.1: Selección de variables con RFE para variante Wuhan según método (*Generalized Boosted Models*, *random forests*) y preprocesamiento. Se incluye la estimación en validación cruzada repetida (10 particiones, 3 repeticiones) del AUC para cada método en dicho proceso de RFE.

	Method[preprocessing] (AUC +/- AUC std. dev.)			
	GBM (0.86 +/- 0.035)	**RF (0.85 +/- 0.047)**	**GBM[discretized] (0.84 +/- 0.035)**	**RF[discretized] (0.81 +/- 0.047)**
1	Age	Age	Age	Age
2	Albumin	Albumin	Albumin	Alanine transaminase (ALT)
3	Blood urea nitrogen (BUN)	Blood urea nitrogen (BUN)	Blood urea nitrogen (BUN)	Arterial hypertension disease history
4	Body temperature (daily minimum)	C-reactive protein	Body temperaure (daily maximum)	Aspartate transaminase (AST)
5	Body temperaure (daily maximum)	Creatinine	Creatinine	Blood urea nitrogen (BUN)
6	C-reactive protein	Current bicarbonate (blood gas test)	Current bicarbonate (blood gas test)	Body temperaure (daily maximum)
7	Current bicarbonate (blood gas test)	Eosinophil %	Heart rate (daily maximum)	C-reactive protein
8	D-Dimer	Estimated glomerular filtration rate (eGFR) ckd-epi	Lymphocyte %	Calcium
9	Diastolic blood pressure (daily maximum)	Heart rate (daily maximum)	Neurological disease history	Creatinine
10	Estimated glomerular filtration rate (eGFR) ckd-epi	Hematocrit	Oxygen saturation (daily maximum)	Eosinophil %
11	Glucose	Hemoglobin	Oxygen saturation (first measure)	Estimated glomerular filtration rate (eGFR) ckd-epi
12	Heart rate (daily maximum)	International normalized ratio (INR)	Partial pressure of oxygen (Blood gas test)	Ferritin
13	Hematocrit	Lactate dehydrogenase (LDH)	Red blood cells	Heart rate (daily maximum)
14	International normalized ratio (INR)	Lymphocyte %	Red Cell Blood Distribution Width (RDW)	International normalized ratio (INR)
15	Lactate dehydrogenase (LDH)	Lymphocyte count	Sodium	Lactate dehydrogenase (LDH)
16	Lymphocyte %	Mean corpuscular volume	Systolic blood pressure (first measure)	Lymphocyte %
17	Mean corpuscular volume	Monocytes %	Total bilirubin	Neurological disease history
18	Monocytes %	Oxygen saturation (daily maximum)	Total CO2 (blood gas test)	Oxygen saturation (daily maximum)
19	Oxygen saturation (daily maximum)	Oxygen saturation (daily minimum)	Urea	Oxygen saturation (daily minimum)
20	Oxygen saturation (daily minimum)	Partial pressure of CO2 (Blood gas test)		Oxygen saturation (first measure)
21	Partial pressure of CO2 (Blood gas test)	Prothrombin time (PT)		Partial pressure of oxygen (Blood gas test)
22	Partial pressure of oxygen (Blood gas test)	Prothrombin Time (Quick)		Prothrombin time (PT)
23	Platelets	Red blood cells		Red blood cells
24	Red Cell Blood Distribution Width (RDW)	Red Cell Blood Distribution Width (RDW)		Red Cell Blood Distribution Width (RDW)
25	Segmented neutrophils %	Segmented neutrophils %		Segmented neutrophils %
26	Systolic blood pressure (daily maximum)	Sodium		Systolic blood pressure (daily maximum)
27	Systolic blood pressure (first measure)	Systolic blood pressure (daily maximum)		Systolic blood pressure (first measure)
28	Total CO2 (blood gas test)	Systolic blood pressure (first measure)		Total CO2 (blood gas test)
29	Urea	Total CO2 (blood gas test)		Urea
30		Urea		

Tabla B.2: Selección de variables con RFE para variante Alfa según método (*Generalized Boosted Models*, *random forests*) y preprocesamiento. Se incluye la estimación en validación cruzada repetida (10 particiones, 3 repeticiones) del AUC para cada método en dicho proceso de RFE.

	Method[preprocessing] (AUC +/- AUC std. dev.)			
	GBM (0.85 +/- 0.057)	**RF (0.84 +/- 0.056)**	**GBM[discretized] (0.83 +/- 0.044)**	**RF[discretized] (0.80 +/- 0.051)**
1	Activated Partial Thromboplastin Time (aPTT)	Activated Partial Thromboplastin Time (aPTT)	Age	Age
2	Age	Age	Albumin	Arterial hypertension disease history
3	Albumin	Albumin	Aspartate transaminase (AST)	Aspartate transaminase (AST)
4	Blood urea nitrogen (BUN)	Arterial hypertension disease history	Blood urea nitrogen (BUN)	Blood urea nitrogen (BUN)
5	Body temperaure (daily maximum)	Blood urea nitrogen (BUN)	Body temperaure (daily maximum)	Body temperaure (daily maximum)
6	Calcium	Body temperaure (daily maximum)	Cardiac disease history	Cardiac disease history
7	Current bicarbonate (blood gas test)	D-Dimer	Current bicarbonate (blood gas test)	Current bicarbonate (blood gas test)
8	D-Dimer	Eosinophil %	Diastolic blood pressure (daily minimum)	Eosinophil %
9	Heart rate (daily maximum)	Eosinophil count	Eosinophil %	Heart rate (daily maximum)
10	Heart rate (daily minimum)	Estimated glomerular filtration rate (eGFR) ckd-epi	Heart rate (daily maximum)	Heart rate (first measure)
11	Heart rate (first measure)	Heart rate (daily maximum)	Heart rate (first measure)	Hematocrit
12	Lactate dehydrogenase (LDH)	Heart rate (daily minimum)	Hemoglobin	Hemoglobin
13	Lymphocyte %	Heart rate (first measure)	Leukocytes count	Hemolysis index
14	Lymphocyte count	International normalized ratio (INR)	Lung disease history	International normalized ratio (INR)
15	Monocytes %	Lactate dehydrogenase (LDH)	Mean corpuscular hemoglobin	Lung disease history
16	Oxygen saturation (daily minimum)	Leukocytes count	Oxygen saturation (first measure)	Lymphocyte %
17	Partial Thromboplastin Time ratio	Lymphocyte %	Platelets	Lymphocyte count
18	Red Cell Blood Distribution Width (RDW)	Lymphocyte count	Potassium	Mean corpuscular hemoglobin
19		Monocytes %	Prothrombin time (PT)	Mean corpuscular hemoglobin concentration (MCHC)
20		Neutrophil count	Red Cell Blood Distribution Width (RDW)	Oxygen saturation (daily maximum)
21		Oxygen saturation (daily maximum)	Sex	Oxygen saturation (first measure)
22		Oxygen saturation (daily minimum)	Smoker	Prothrombin time (PT)
23		Partial Thromboplastin Time ratio	Sodium	Prothrombin Time (Quick)
24		Prothrombin time (PT)	Systolic blood pressure (daily maximum)	Red blood cells
25		Prothrombin Time (Quick)	Systolic blood pressure (first measure)	Red Cell Blood Distribution Width (RDW)
26		Red blood cells	Total CO2 (blood gas test)	Smoker
27		Red Cell Blood Distribution Width (RDW)	Vaccinated (>=1 dose)	Systolic blood pressure (daily maximum)
28		Segmented neutrophils %		Systolic blood pressure (first measure)
29		Total CO2 (blood gas test)		Urea
30		Urea		Vaccinated (>=1 dose)

Tabla B.3: Selección de variables con RFE para variante Delta según método (*Generalized Boosted Models*, *random forests*) y preprocesamiento. Se incluye la estimación en validación cruzada repetida (10 particiones, 3 repeticiones) del AUC para cada método en dicho proceso de RFE.

	Method[preprocessing] (AUC +/- AUC std. dev.)			
	GBM (0.82 +/- 0.070)	**RF (0.80 +/- 0.063)**	**GBM[discretized] (0.73 +/- 0.086)**	**RF[discretized] (0.72 +/- 0.079)**
1	Activated Partial Thromboplastin Time (aPTT)	Age	Age	Age
2	Age	Blood gas test pH	Albumin	Albumin
3	Blood urea nitrogen (BUN)	Blood urea nitrogen (BUN)	Aspartate transaminase (AST)	Aspartate transaminase (AST)
4	C-reactive protein	C-reactive protein	Blood gas test pH	Blood urea nitrogen (BUN)
5	Creatinine	Cardiac disease history	Blood urea nitrogen (BUN)	Body temperaure (daily maximum)
6	D-Dimer	Creatinine	Body temperaure (daily maximum)	Cardiac disease history
7	Derived fibrinogen	Derived fibrinogen	Cardiac disease history	Creatinine
8	Eosinophil %	Direct bilirubin	Creatinine	D-Dimer
9	Glucose	Eosinophil %	Diastolic blood pressure (first measure)	Eosinophil %
10	Heart rate (daily maximum)	Estimated glomerular filtration rate (eGFR) ckd-epi	Heart rate (first measure)	Estimated glomerular filtration rate (eGFR) ckd-epi
11	Heart rate (first measure)	Glucose	Hematocrit	Ferritin
12	Hematocrit	Heart rate (daily maximum)	International normalized ratio (INR)	Heart rate (daily maximum)
13	Hemoglobin	Heart rate (daily minimum)	Lactate dehydrogenase (LDH)	Heart rate (first measure)
14	Hemolysis index	Hematocrit	Lymphocyte count	International normalized ratio (INR)
15	Lactate dehydrogenase (LDH)	Hemoglobin	Mean corpuscular hemoglobin	Lymphocyte count
16	Lymphocyte %	Lactate dehydrogenase (LDH)	Mean corpuscular hemoglobin concentration (MCHC)	Mean corpuscular hemoglobin concentration (MCHC)
17	Mean corpuscular hemoglobin concentration (MCHC)	Lymphocyte %	Mean corpuscular volume	Monocytes %
18	Mean corpuscular volume	Mean corpuscular hemoglobin concentration (MCHC)	Monocytes %	Neutrophil count
19	Monocytes %	Mean corpuscular volume	Oxygen saturation (first measure)	Oxygen saturation (daily maximum)
20	Oxygen saturation (daily maximum)	Monocytes %	Partial Thromboplastin Time ratio	Oxygen saturation (daily minimum)
21	Oxygen saturation (daily minimum)	Oxygen saturation (daily maximum)	Red Cell Blood Distribution Width (RDW)	Oxygen saturation (first measure)
22	Oxygen saturation (first measure)	Oxygen saturation (daily minimum)	Renal disease history	Prothrombin time (PT)
23	Partial Thromboplastin Time ratio	Oxygen saturation (first measure)	Smoker	Prothrombin Time (Quick)
24	Prothrombin time (PT)	Prothrombin time (PT)	Sodium	Red blood cells
25	Segmented neutrophils %	Segmented neutrophils %	Systolic blood pressure (daily maximum)	Red Cell Blood Distribution Width (RDW)
26	Systolic blood pressure (daily maximum)	Sodium	Total CO2 (blood gas test)	Segmented neutrophils %
27	Systolic blood pressure (daily minimum)	Systolic blood pressure (daily maximum)	Vaccinated (>=1 dose)	Systolic blood pressure (daily maximum)
28	Systolic blood pressure (first measure)	Urea		Systolic blood pressure (first measure)
29	Urea			Urea

Tabla B.4: Selección de variables con RFE para variante Ómicron según método (*Generalized Boosted Models*, *random forests*) y preprocesamiento. Se incluye la estimación en validación cruzada repetida (10 particiones, 3 repeticiones) del AUC para cada método en dicho proceso de RFE.

Apéndice C

Redes Bayesianas discretas

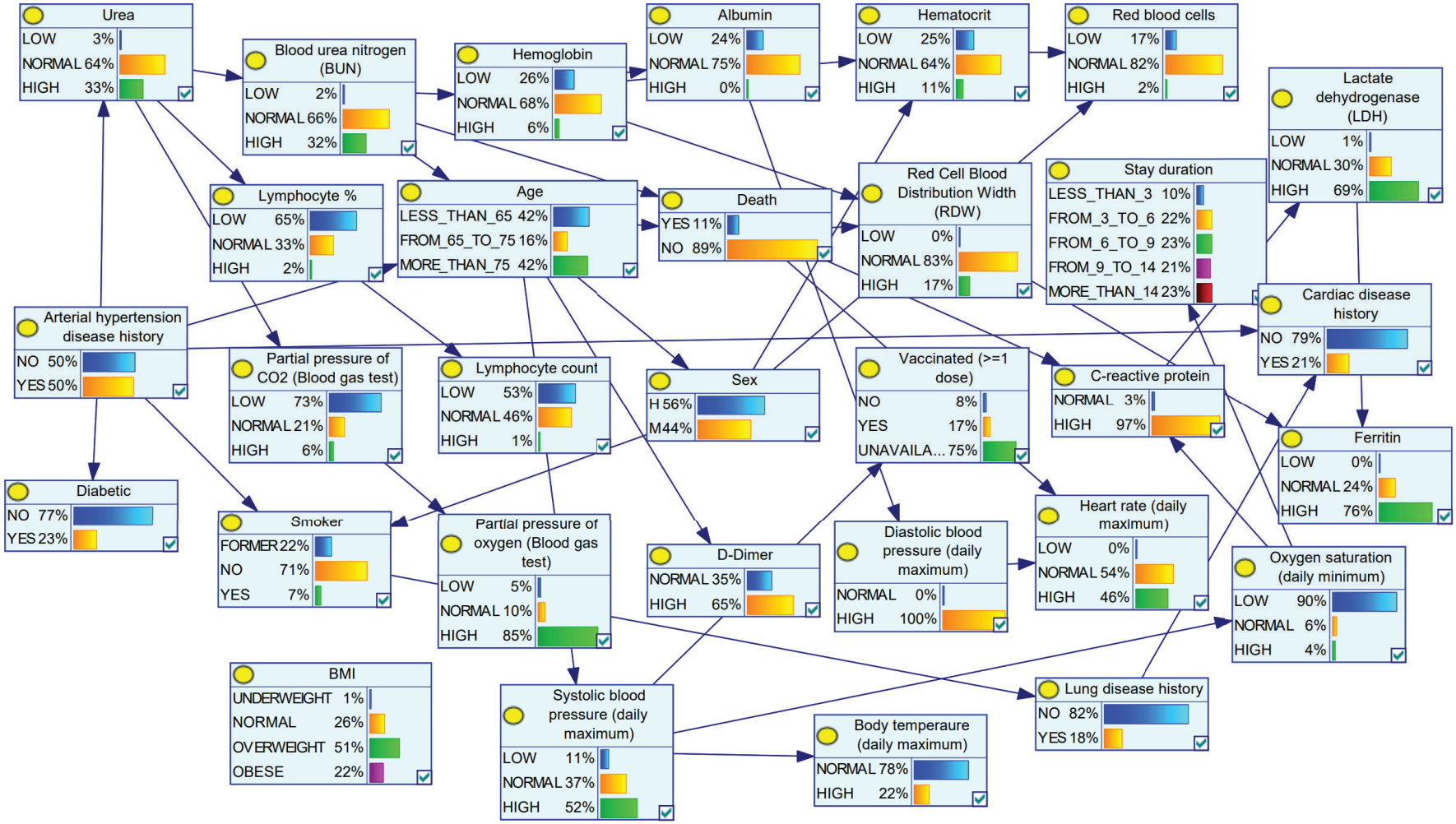


Figura C.1: Red Bayesiana discreta para la variante Delta.

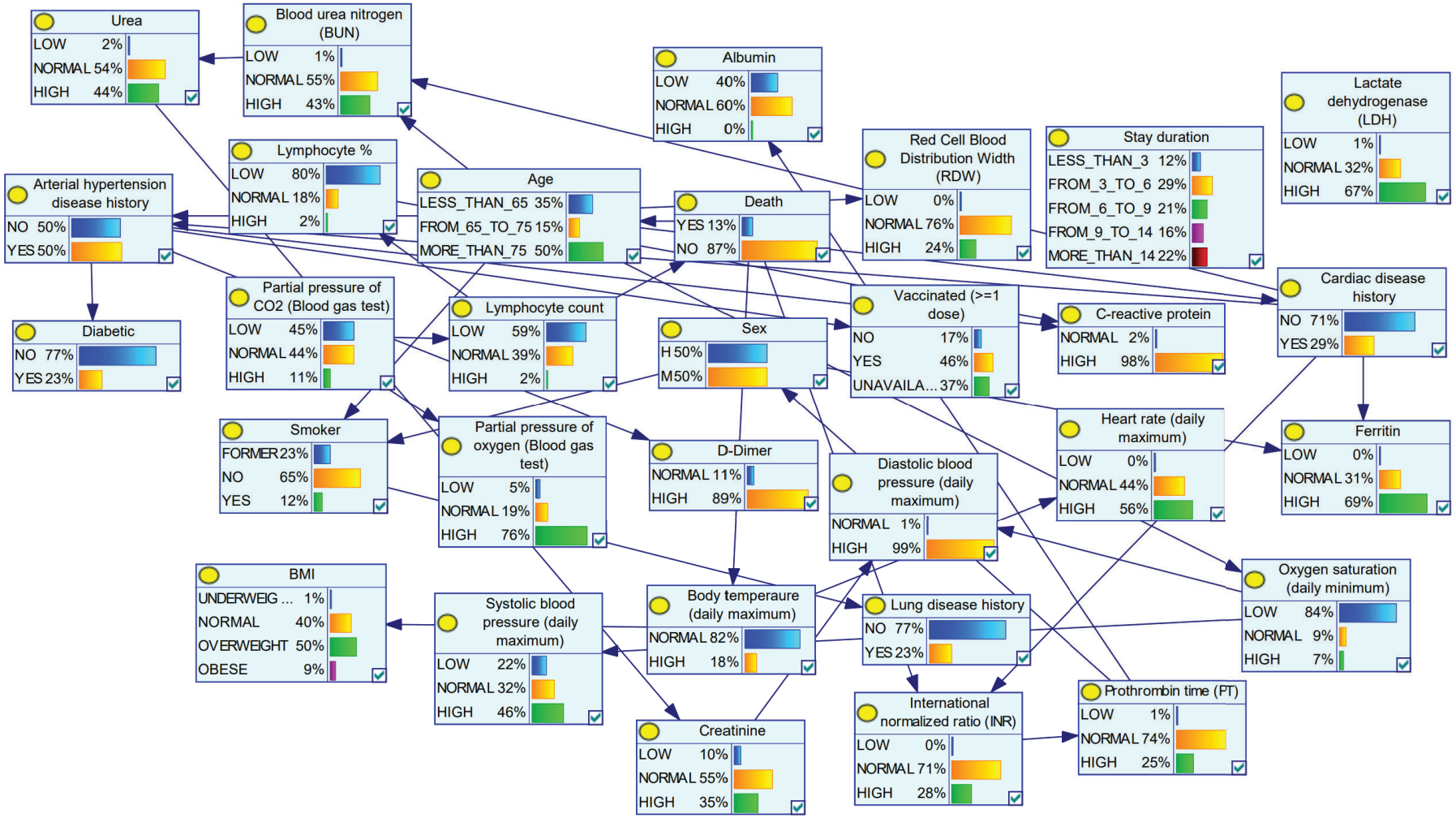


Figura C.2: Red Bayesiana discreta para la variante Ómicron.