



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Informática



TFG del Grado en Ingeniería Informática

**Investigación, creación y
explotación de modelos para el
estudio del COVID-19 en
pacientes del Servicio Cántabro de
Salud**



Presentado por Segundo González González
en Universidad de Burgos — 6 de julio de 2023

Tutor: Carlos López Nozal
Contutor: José Francisco Díez Pastor



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Informática



D. Carlos López Nozal, profesor del departamento de Ingeniería Informática,
área de Lenguajes y Sistemas Informáticos.

Expone:

Que el alumno D. Segundo González González, con DNI 20.219.016-S, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado "Investigación, creación y explotación de modelos para el estudio del COVID-19 en pacientes del Servicio Cántabro de Salud".

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 6 de julio de 2023

Vº. Bº. del Tutor:

D. Carlos López Nozal

Vº. Bº. del Contutor:

D. José Francisco Díez Pastor

Resumen

El COVID-19 es una enfermedad respiratoria causada por el coronavirus SARS-CoV-2. Fue identificado por primera vez en la ciudad de Wuhan, China, en diciembre de 2019 y desde entonces se ha propagado a nivel mundial, convirtiéndose en una pandemia.

El impacto del COVID-19 en la sociedad ha sido significativo, afectando la salud pública, la economía global, la educación y la vida cotidiana de las personas. Muchos países han implementado medidas de confinamiento y restricciones en los desplazamientos para frenar la propagación del virus, lo que ha generado impactos socioeconómicos y cambios en la forma en que las personas trabajan, estudian y se relacionan entre sí.

Los síntomas más comunes del COVID-19 incluyen fiebre, tos seca y dificultad para respirar. Sin embargo, algunas personas pueden ser asintomáticas o presentar síntomas leves, mientras que otras pueden desarrollar complicaciones graves, especialmente aquellos con enfermedades subyacentes y personas de edad avanzada. Por todo esto, adquiere gran importancia tener una prueba que identifique qué pacientes tienen COVID-19 lo antes posible y su grado de severidad para poder empezar a tratarlos inmediatamente.

Para este proyecto se cuenta con datos proporcionados por el Instituto de Investigación Sanitaria de Valdecilla (IDIVAL). El objetivo principal de este trabajo será la investigación y creación de modelos de aprendizaje automático que facilite el triaje en la admisión de pacientes con COVID-19.

Descriptores

COVID-19, clasificación, aprendizaje automático, diagnóstico, triaje, predicción, soporte a la decisión, python.

Abstract

COVID-19 is a respiratory disease caused by the SARS-CoV-2 coronavirus. It was first identified in the city of Wuhan, China, in December 2019 and has since spread globally, becoming a pandemic.

The impact of COVID-19 on society has been significant, affecting public health, the global economy, education, and people's daily lives. Many countries have implemented lockdown measures and travel restrictions to slow the spread of the virus, which has generated socioeconomic impacts and changes in the way people work, study and interact with each other.

The most common symptoms of COVID-19 include fever, dry cough, and shortness of breath. However, some people may be asymptomatic or have mild symptoms, while others may develop serious complications, especially those with underlying diseases and the elderly. For all this, it is of great importance to have a test that identifies which patients have COVID-19 as soon as possible and its degree of severity in order to start treating them immediately.

For this TFG there are data provided by the Valdecilla Health Research Institute (IDIVAL). The objective of this work will be the research and creation of a machine learning model that facilitates triage in the admission of patients with COVID-19.

Keywords

COVID-19, classification, machine learning, diagnosis, triage, prediction, decision support, python.

Índice general

Índice general	III
Índice de figuras	V
Índice de tablas	VI
Introducción	1
1.1. Machine Learning en el triaje del COVID-19	2
Objetivos del proyecto	4
2.1. Objetivos principales	4
2.2. Objetivos personales	5
Conceptos teóricos	6
3.1. Minería de Datos	6
3.2. CRISP-DM	6
3.3. Imputación de datos	9
Técnicas y herramientas	11
4.1. Docker	11
4.2. Anaconda	11
4.3. Python	12
4.4. Bibliotecas de Python	12
4.5. Git	14
4.6. LATEX	15
Aspectos relevantes del desarrollo del proyecto	16
5.1. Estado del arte	16
5.2. Preparación del entorno de trabajo	17
5.3. Cohorte de datos	18

ÍNDICE GENERAL

IV

5.4. Réplica del trabajo original	22
5.5. Búsqueda del clasificador óptimo con <i>Autosklearn</i>	23
5.6. Optimización de hiperparámetros	27
Trabajos relacionados	28
6.1. Búsqueda de trabajos relacionados	28
6.2. Clasificación de variables, técnicas y herramientas	31
Conclusiones y Líneas de trabajo futuras	36
7.1. Conclusiones	36
7.2. Líneas de trabajo futuras	37
Bibliografía	38

Índice de figuras

1.1.	Casos COVID-19 en todo el mundo [5].	1
1.2.	Diagrama esquemático representando el proceso de desarrollo de un modelo de aprendizaje automático. [21]	2
3.3.	Metodología de datos CRISP-DM [7]	7
5.4.	Jupyter Docker Stacks [17]	17
5.5.	Valores nulos en el dataset	22
5.6.	Autosklearn en una imagen [4]	23
5.7.	Resultado experimento réplica con 10 repeticiones	24
5.8.	Resultado experimento réplica con 5 repeticiones	25
5.9.	Resultado experimento con todas las caract. y 10 repeticiones	25
5.10.	Resultado experimento con todas las caract. y 5 repeticiones	26
5.11.	Pipeline profiler experimento con todas las caract. y 5 repeticiones	26
5.12.	Contribución media de las características de cada clase	27

Índice de tablas

5.1.	Variables, técnicas y herramientas más utilizadas	17
5.2.	Variables del dataset	21
6.3.	Trabajos relacionados con este proyecto	31
6.4.	Clasificación de variables, técnicas y herramientas	35

Introducción

El síndrome respiratorio agudo severo coronavirus 2 (SARS-CoV-2), más conocido como COVID-19, tuvo sus primeros casos en la ciudad de Wuhan, en la provincia de Hubei (China) a finales de 2019, en la que se detectaron los primeros casos de una enfermedad respiratoria desconocida. Desde entonces se ha expandido a lo largo del mundo creando una pandemia con más de 676.609.837 de casos confirmados y con un total de 6.881.955 de muertes asociadas. En España, la incidencia de casos confirmados es de 13.770.429 de pacientes con una mortalidad de 119.479 personas.¹

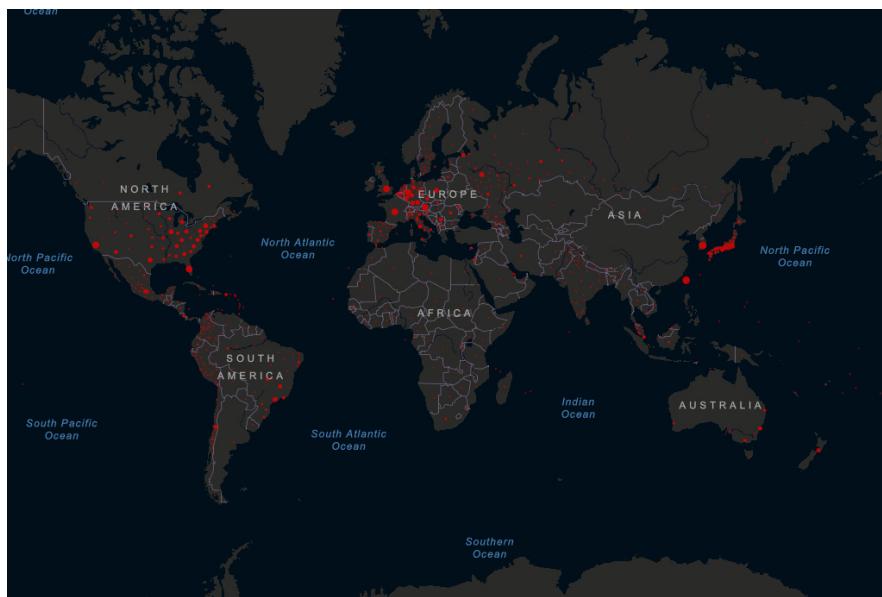


Figura 1.1: Casos COVID-19 en todo el mundo [5].

¹Datos extraídos del cuadro de mando de COVID-19 de la Universidad John HopKins (JHU) a fecha 29/06/2023

En Cantabria, pese a ser una comunidad autónoma con una densidad de población media, el número de casos confirmados asciende a 131.501 en tanto que la enfermedad ha producido 815 fallecidos.²

La prueba más fiable en la detección de la enfermedad es la Reverse Transcription-Polymerase Chain Reaction (RT-PCR) cuyo resultado tarda en devolverse unas horas. Los resultados de la prueba PCR solo ayudan a detectar si un paciente tiene la enfermedad (positivo) o no (negativo). Además, de esta información, sería interesante intentar predecir la posible evolución de la enfermedad en el paciente. Es por esto de la importancia de desarrollar modelos de aprendizaje automático que ayuden en el triaje de los pacientes de forma rápida para poder controlar la enfermedad en los centros hospitalarios.

1.1. Machine Learning en el triaje del COVID-19

La aparición de la pandemia desencadenada por el COVID-19 produjo muchos casos de pacientes con síntomas en muy poco tiempo, lo que condujo a una falta de recursos hospitalarios. Esto obligó a buscar alternativas en el aprendizaje automático para producir modelos [16][23][10][21][11] que pudieran ayudar en el triaje de los pacientes que llegasen al hospital y así ofrecer a los profesionales sanitarios una herramienta con la que poder tomar decisiones.

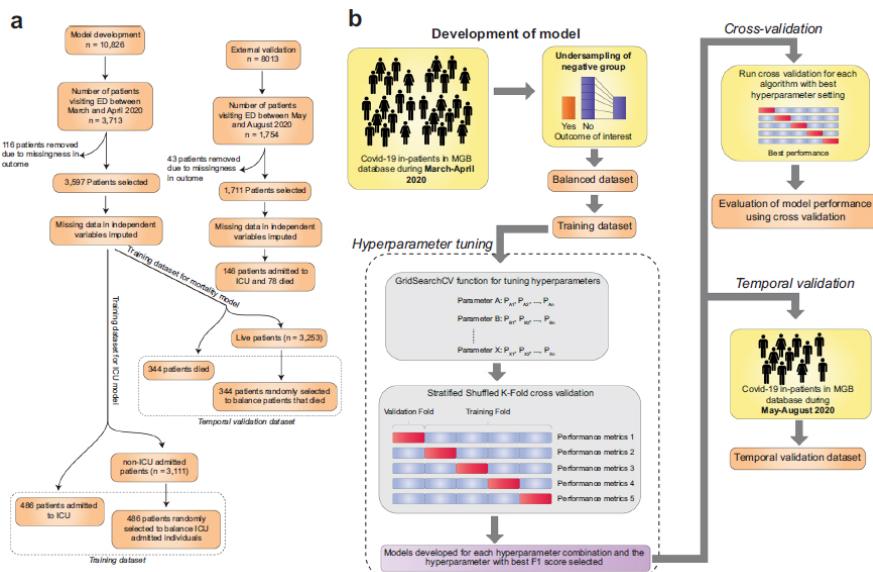


Figura 1.2: Diagrama esquemático representando el proceso de desarrollo de un modelo de aprendizaje automático. [21]

²Datos extraídos del portal del Servicio Cántabro de Salud (SCS) con ultima actualización a fecha 30/03/2022

El artículo científico que se toma como referencia *Innate and Adaptive Immune Assessment at Admission to Predict Clinical Outcome in COVID-19 Patients* [18] evalúa un amplio panel multiparamétrico de anticuerpos de componentes celulares y humorales de la respuesta inmunitaria innata y adaptativa para buscar biomarcadores que pronosticasen el COVID-19.

El conjunto de datos utilizado está formado por muestras tomadas a 155 pacientes al ingreso del hospital y se categorizaron como leves o graves, en el caso de requerir oxigenoterapia. El modelo predictivo que se utilizó es la regresión logística e incluyó las siguientes características: la edad, la ferritina, el dímero D, la suma absoluta de linfocitos, el % monocitos no clásicos, C4 y el % de $CD8^+CD27^-CD28^-$.

La edad, el dímero D y la ferritina son características utilizadas habitualmente por los científicos en modelos predictivos, tal y como veremos en el apartado [trabajos relacionados](#).

En el presente trabajo, se abordará el triaje de pacientes a partir de marcadores recogidos en la admisión con aprendizaje automático. Replicaremos el modelo de regresión logística utilizado en el artículo original y buscaremos nuevos modelos predictivos que pudieran mejorar sus datos. Los términos marcador, variable y característica son equivalentes y dependen del contexto en el que se utilicen. En entornos hospitalarios se utiliza marcador y variable y en entornos de investigación en ciencias de datos se utilizan característica y variable.

Objetivos del proyecto

El objeto de este proyecto es desarrollar modelos de aprendizaje automático que posibiliten a los médicos realizar un triaje en la admisión de pacientes con sintomatología COVID-19 a partir de variables/marcadores que se puedan obtener de forma rápida sin tener que esperar a una prueba PCR positiva y que, además, pueda predecir la gravedad de la enfermedad para permitir conocer qué recursos hospitalarios (ventilación mecánica, ingreso en UCI, etc...) va a necesitar cada paciente.

2.1. Objetivos principales

- Replicar el trabajo *Innate and Adaptive Immune Assessment at Admission to Predict Clinical Outcome in COVID-19 Patients* [18].
- Buscar trabajos relacionados y clasificar las variables, técnicas y herramientas que se utilizaron.
- Estudiar los datos proporcionados por el Instituto de Investigación Sanitaria de Valdecilla (IDIVAL) y considerar como procesarlos para poder obtener un modelo de predicción que mejore los rendimientos del trabajo *Innate and Adaptive Immune Assessment at Admission to Predict Clinical Outcome in COVID-19 Patients* [18].
- Utilizar librerías de aprendizaje automático que ayuden a encontrar el mejor método de clasificación de entre diferentes métodos a partir de las variables/marcadores para el Covid-19 y disponibles en el conjunto de datos.
- Minimizar las características a utilizar para su utilización en el modelo a desarrollar.
- Investigar qué hiperparámetros pueden mejorar el método de clasificación elegido para el modelo final.

2.2. Objetivos personales

- Poner en práctica los conocimientos obtenidos de minería de datos.
- Desarrollar el proyecto de manera colaborativa en Github.
- Documentar la memoria del proyecto con L^AT_EX.

Conceptos teóricos

3.1. Minería de Datos

La minería de datos es un proceso de exploración y análisis de grandes conjuntos de datos con el objetivo de descubrir patrones, relaciones y conocimientos útiles. Se utiliza en diversas industrias y disciplinas para tomar decisiones basadas en datos y obtener información valiosa. La cantidad de datos que se produce cada día crece de forma exponencial en todos los aspectos de la vida. Esto nos ha llevado a utilizar técnicas como el aprendizaje automático para poder explotar esos datos y sacarles un rendimiento.

Para poder extraer conocimiento de esta ingente cantidad de datos existen diversas metodologías como CRISP-DM, Scrum, Kanban... Nosotros hemos utilizado la primera para crear los modelos de aprendizaje automático.

3.2. CRISP-DM

La metodología CRISP-DM (Cross-Industry Standard Process for Data Mining) es un enfoque utilizado para abordar proyectos de minería de datos. Aunque originalmente se desarrolló para aplicaciones comerciales, se puede adaptar para analizar y abordar los desafíos relacionados con el COVID-19.

Se compone de 6 fases:

- **Entendimiento del Negocio** Comprensión clara de los objetivos y requisitos relacionados con el proyecto.
- **Entendimiento de los Datos** Recopilación y exploración de datos para comprender su calidad, estructura y características, identificando posibles problemas o limitaciones que nos podrían afectar a los análisis posteriores.

- **Preparación de los datos** Realización de tareas de limpieza y transformación de los datos, imputando datos que pudieran faltar, eliminando duplicados, normalizando dichos datos y seleccionando características relevantes.
- **Modelado** y análisis de datos para responder a las preguntas planteadas en la etapa de comprensión del negocio, incluyendo la construcción de modelos predictivos. En esta fase puede ser necesario volver a la fase anterior para volver a preparar los datos.
- **Evaluación** de los modelos desarrollados, analizando los resultados y comparándolos con los objetivos iniciales. Se considera la precisión, la eficacia y la calidad de los modelos.
- **Despliegue** del modelo construido que haya dado mejor resultado para que pueda ser utilizado con otros datos.

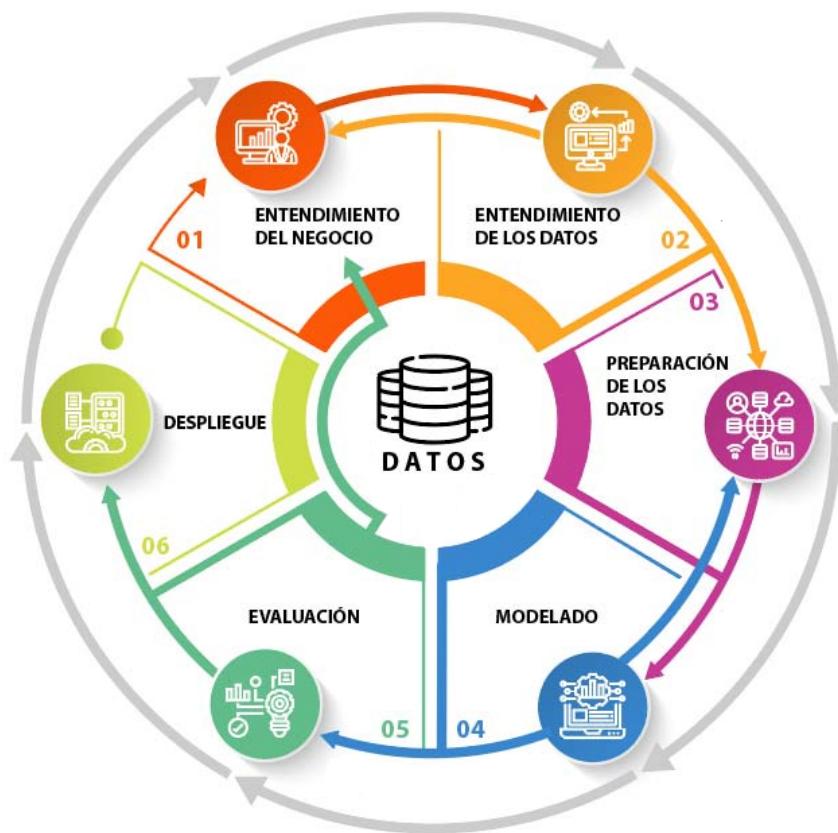


Figura 3.3: Metodología de datos CRISP-DM [7]

A continuación, se proporciona un resumen de cómo se ha aplicado la metodología CRISP-DM a este proyecto:

Entendimiento del negocio

El triaje sanitario es el proceso de evaluar la gravedad de la condición de un paciente y determinar el orden de atención médica en situaciones en las que los recursos son limitados. La implementación de modelos de aprendizaje automático en los centros hospitalarios permitiría ayudar en la clasificación y priorización de pacientes basándose en datos clínicos y síntomas relacionados con la enfermedad. Además, esto permitiría conocer de antemano la necesidad de recursos hospitalarios, como camas de cuidados intensivos o respiradores, y planificar en consecuencia.

El aprendizaje automático puede ser una herramienta útil en el triaje sanitario, pero no debe reemplazar la experiencia y el juicio clínico de los profesionales de la salud. Los modelos de aprendizaje automático deben utilizarse como una herramienta de apoyo a la toma de decisiones, pero siempre debe haber una supervisión y revisión médica adecuada para tomar decisiones finales.

El desarrollo de modelos de aprendizaje automático puede ser de gran utilidad en el triaje de pacientes y en la gestión de recursos en los centros hospitalarios durante la pandemia de la enfermedad. Estos modelos podrían ayudar a controlar la propagación de la enfermedad y mejorar la atención médica de los pacientes.

Entendimiento de los datos

La cohorte de datos han sido proporcionados por el Instituto de Investigación Sanitaria de Valdecilla (IDIVAL) sobre una población de 305 pacientes, cuyas muestras fueron recogidas entre Abril de 2020 y Marzo de 2021 y que tuvieron un resultado positivo en COVID-19. El valor de clasificación es el **score** de evolución clínica que tiene un valor de 1 a 5, siendo 1=asintomático; 2=tratamiento con gafas nasales; 3=tratamiento con ventimask; 4=Ingreso en UCI; 5=exitus.

El dataset se compone de un archivo excel con las siguientes características:

- 420 filas o determinaciones.
- 97 columnas o marcadores.
- Sólo hay 158 filas con el valor de clasificación score cumplimentado.

Investigando los datos, vemos características que no vamos a utilizar debido a qué son fechas o números identificativos de muestras o pacientes que no nos aportan ningún tipo de información.

Preparación de los datos

Creamos un nuevo dataset copia del original y eliminamos las columnas que comentamos en la sección anterior que no nos aportan datos para clasificación.

Asimismo, observamos que el valor de clasificación *score* tiene muchas filas vacías, lo que no nos permitiría realizar la clasificación, por lo que también las eliminamos.

Una vez tenemos las características con las que vamos a trabajar, observamos que tienen muchos valores perdidos, por lo que utilizaremos la **imputación de datos** para otorgar un valor aproximado para ese dato perdido.

Modelado

Aquí se aplican técnicas de modelado y análisis de datos para responder a las preguntas planteadas en la etapa de entendimiento del negocio. En el caso de este proyecto, se han realizado varios modelos, utilizando **Autosklearn** para realizar una búsqueda del clasificador que mejor funciona con el dataset. Los modelos generados con **Ida** han sido los dos métodos de clasificación con los mejores resultados.

Evaluación

En esta fase, se evalúan los modelos desarrollados. Se analizan los resultados y se comparan con el trabajo de partida inicial. Si los modelos no cumplen con los requisitos, se pueden realizar ajustes o mejoras.

Despliegue

Los modelos desarrollados se podrán poner a disposición de los profesionales para poder obtener un triaje de pacientes con síntomas COVID-19 a partir de una serie de muestras que se les recoge en el hospital.

3.3. Imputación de datos

La imputación de datos se refiere al proceso de reemplazar los valores perdidos en un conjunto de datos con valores estimados o inferidos. Es una técnica comúnmente utilizada en el análisis de datos para abordar el problema de los valores perdidos, ya que estos pueden afectar la calidad y la validez de los resultados obtenidos.

Existen varios métodos y técnicas, pero en los modelos de este proyecto he utilizado dos:

- **Imputación simple** Consiste en reemplazar los valores perdidos por un único valor, como la media, la mediana o el valor más frecuente del conjunto de datos. Esta técnica es rápida y sencilla, pero no tiene en cuenta la relación entre las variables y puede introducir sesgos en los resultados.
- **Imputación por vecinos más cercanos** Se basa en encontrar observaciones similares a aquellas con valores que faltan y utilizar los valores de las observaciones vecinas para estimar los valores perdidos. Esta técnica es útil cuando los datos tienen una estructura de vecindad o proximidad.

Técnicas y herramientas

A continuación se enumeran y detallan someramente algunas técnicas y herramientas que se ha utilizado en la realización y desarrollo de este proyecto:

4.1. Docker

Docker[8] es un software libre y de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos. En este TFG, he utilizado Docker instalado en un NAS de QNAP para crear un contenedor con Anaconda Distribution y así poder realizar los diversos modelos en python. <https://www.docker.com>

Docker Hub

Es un servicio en línea proporcionado por Docker que sirve como repositorio central para imágenes de contenedores Docker. Es un registro público y gratuito donde los desarrolladores y las organizaciones pueden almacenar, compartir y descargar imágenes de contenedores Docker.

4.2. Anaconda

Anaconda[9] es una distribución de Python ampliamente utilizado por los desarrolladores. <https://www.anaconda.com>

Jupyter Notebooks

Dentro de la distribución de python de Anaconda se encuentran los cuadernos Jupyter[12]. También ampliamente utilizados, permiten presentar en un mismo documento texto maquetado con Markdown, código en diferentes lenguajes y los resultados de la ejecución de dicho código, pudiendo representar

dichos datos en forma de texto o gráficos, lo que representa una herramienta fundamental para la formación y para la explicación paso a paso de códigos de desarrollo. <https://www.jupyter.org>

4.3. Python

Es un lenguaje de programación interpretado y de alto nivel. Es conocido por su sintaxis sencilla y legible, lo que lo hace ideal para principiantes. Python[22] es versátil y se puede utilizar para desarrollar una amplia gama de aplicaciones, desde scripts simples hasta aplicaciones web y científicas de alto rendimiento. Tiene una amplia biblioteca estándar que cubre muchas tareas comunes, lo que facilita el desarrollo de aplicaciones sin necesidad de depender de bibliotecas externas. Python es orientado a objetos, lo que permite una programación modular y estructurada. Es interpretado y multiplataforma, lo que significa que el código fuente se ejecuta directamente y puede ejecutarse en diferentes sistemas operativos.

4.4. Bibliotecas de Python

Python cuenta con una amplia variedad de bibliotecas que proporcionan funcionalidades adicionales y extienden las capacidades del lenguaje. Estas bibliotecas cubren una amplia gama de áreas y permiten a los desarrolladores ahorrar tiempo y esfuerzo al aprovechar el trabajo previo de otros programadores.

Pandas

Ofrece estructuras de datos y herramientas de análisis de datos de alto rendimiento. Permite manipular, limpiar y analizar datos de manera eficiente.

Numpy

Proporciona un soporte completo para arreglos multidimensionales y operaciones matemáticas de alto rendimiento. Es ampliamente utilizado en el procesamiento numérico y científico de datos.

Matplotlib

Es una biblioteca de trazado de gráficos 2D que permite crear visualizaciones de datos de alta calidad. Se utiliza comúnmente para generar gráficos, histogramas, diagramas de dispersión y mucho más.

Scikit-learn

Es una biblioteca de aprendizaje automático de código abierto que proporciona algoritmos para tareas comunes, como clasificación, regresión, agrupación y selección de características.

StratifiedKFold

Es una técnica de validación cruzada utilizada en el aprendizaje automático para evaluar y validar modelos de clasificación. Es una extensión de la validación cruzada k-fold estándar, pero con la adición de la estratificación de las muestras en cada pliegue. La estratificación se refiere a que cada pliegue o partición de los datos de entrenamiento y prueba mantiene la misma proporción de clases o etiquetas que el conjunto de datos original.

Pipeline

Es una secuencia ordenada de pasos que se aplican a los datos para realizar tareas de preprocesamiento, transformación y modelado. Un pipeline facilita la organización y automatización del flujo de trabajo en el desarrollo de modelos de aprendizaje automático.

Seaborn

Es una biblioteca de visualización de datos basada en Matplotlib y es ampliamente utilizada en el análisis exploratorio de datos y la creación de gráficos estadísticos. Proporciona una interfaz más sencilla y atractiva visualmente para crear gráficos informativos y estéticamente agradables.

Autosklearn

Es una biblioteca de aprendizaje automático automatizado (AutoML). Proporciona una interfaz fácil de usar que automatiza el proceso de selección de algoritmos, preprocesamiento de datos, ajuste de hiperparámetros y generación de modelos de aprendizaje automático.

Pipeline Profiler

Es una herramienta que se utiliza en el contexto del aprendizaje automático y los pipelines para analizar y comprender el rendimiento y el comportamiento de un pipeline de manera detallada. Proporciona información valiosa sobre cómo se están ejecutando los diferentes componentes del pipeline y ayuda a identificar cuellos de botella y posibles áreas de mejora en términos de tiempo de ejecución y eficiencia. Al utilizar esta información, se pueden realizar ajustes y optimizaciones para mejorar el rendimiento y la eficiencia del pipeline.

Optuna

Proporciona un enfoque basado en pruebas para encontrar automáticamente los mejores valores de hiperparámetros para un modelo de aprendizaje automático. Optuna ayuda a simplificar y automatizar este proceso al buscar de manera eficiente el espacio de hiperparámetros para encontrar la mejor configuración.

Shap

SHAP (SHapley Additive exPlanations) es utilizada para explicar las predicciones de modelos de aprendizaje automático. Proporciona una forma de entender y descomponer la contribución de cada característica o variable en el resultado de la predicción de un modelo. Proporciona una forma intuitiva y cuantitativa de entender cómo las características individuales influyen en las predicciones del modelo, lo que puede ser útil para la depuración de modelos, la toma de decisiones y la confianza en los resultados obtenidos.

Regresión Logística

Es un algoritmo de aprendizaje automático supervisado utilizado para predecir la probabilidad de que una variable binaria dependiente esté presente en función de variables independientes. Aunque el nombre incluye la palabra "egresión", en realidad es un algoritmo de clasificación.

Lda

Linear Discriminant Analysis (LDA) es un algoritmo de aprendizaje supervisado utilizado para la clasificación y reducción de dimensionalidad, que busca encontrar una proyección lineal óptima para maximizar la separabilidad entre diferentes clases.

4.5. Git

Es un sistema de control de versiones distribuido ampliamente utilizado para el desarrollo de software y el seguimiento de cambios en archivos y proyectos.

Github

Es una plataforma en línea basada en Git que permite a los desarrolladores almacenar, colaborar, gestionar y compartir proyectos de software de manera eficiente. Es uno de los servicios de alojamiento de repositorios más populares y ampliamente utilizados en la comunidad de desarrollo de software.

4.6. L^AT_EX

Es un sistema de composición de documentos ampliamente utilizado para crear documentos de alta calidad, especialmente en ámbitos académicos, científicos y técnicos. A diferencia de los procesadores de texto tradicionales, LaTeX se enfoca en la estructura y el contenido del documento en lugar del formato visual, lo que permite obtener resultados profesionales y consistentes. Se utiliza comúnmente para crear documentos científicos, tesis académicas, artículos de investigación, informes técnicos, libros y presentaciones.

Overleaf

Es una plataforma en línea que permite a los usuarios crear, editar y colaborar en documentos LaTeX de manera colaborativa. Es especialmente popular entre la comunidad académica y científica debido a su facilidad de uso y la capacidad de trabajar en documentos LaTeX sin la necesidad de instalar software adicional.

Aspectos relevantes del desarrollo del proyecto

Después de la pandemia que sufrimos por el virus COVID-19 y, por ser trabajador en un hospital, habiendo sido testigo de forma directa del caos que se produjo en los hospitales por la acumulación masiva de casos y la dificultad de diagnosticar esta enfermedad de forma rápida, me propuse investigar de qué forma podría contribuir para poder paliar mejorar esta situación.

Realizando una investigación de los artículos científicos publicados acerca de modelos predictivos de COVID-19, encontré el artículo *Innate and Adaptive Immune Assessment at Admission to Predict Clinical Outcome in COVID-19 Patients*[18] de investigadores del IDIVAL y me puse en contacto con ellos para mostrar interés en experimentar con su dataset aplicándole técnicas de minería de datos para intentar mejorar su resultado.

5.1. Estado del arte

Se realizó una investigación de los diferentes artículos encontrados relacionados con el desarrollo de modelos de aprendizaje automático que pudieran ayudar en el diagnóstico de pacientes con síntomas equivalentes a los desarrollados por el COVID-19.

Se confeccionó un excel con una pestaña con el listado de los artículos escogidos, fecha de publicación, descripción y el motivo del interés al escogerlo. En las demás pestañas se fueron recogiendo de cada artículo las variables escogidas para los experimentos, así como las técnicas y las herramientas utilizadas a la hora de construir los modelos. En la tabla 5.1 se expone un balance de las más usadas:

Variables	Técnicas	Herramientas
Age (11)	AUC ROC (11)	Python (11)
Suma de Linfocitos (8)	Regresión Logística (9)	R (3)
D-Dimer (7)	Random Forest (6)	
Ferritin (4)	XGB (6)	
C Reactive Protein (4)	Extra Trees (3)	

Tabla 5.1: Variables, técnicas y herramientas más utilizadas

En el siguiente capítulo [6. Trabajos relacionados](#) se desarrolla más esta información de los artículos en dos tablas.

Tanto el libro excel en el que se recogen el listado de los artículos y el estudio de sus variables, técnicas y herramientas; así como los artículos descargados en formato pdf, se encuentran en el repositorio del trabajo en Github: [1. Trabajos Relacionados](#)

5.2. Preparación del entorno de trabajo

Para el desarrollo de los cuadernos Jupyter de Anaconda y la ejecución de los mismos, se ha utilizado la dockerización de Anaconda bajo un entorno hardware en un NAS QNAP TS-251+ con un procesador Intel de 4 núcleos y 16Gb de RAM.

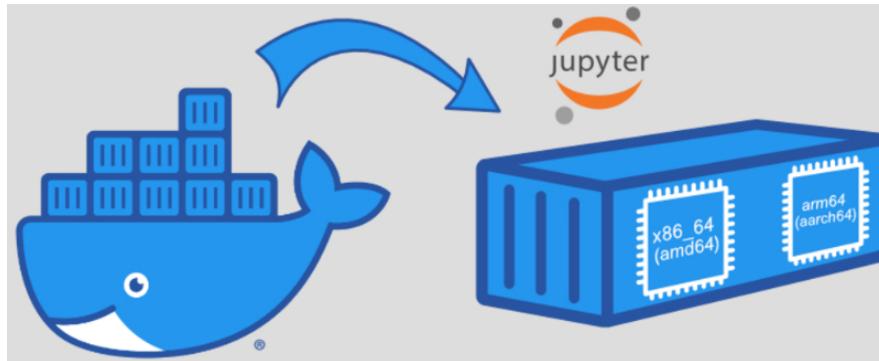


Figura 5.4: Jupyter Docker Stacks [17]

Bajo la aplicación *Container Station* se instaló la imagen del contenedor *jupyter/tensorflow-notebook* [13] que se descargó de la plataforma Docker Hub. En la instalación se añadió la creación de un volumen de almacenamiento persistente para poder ir guardando de forma segura los cuadernos que se iban realizando. Este contenedor se escogió porque ya incorpora gran parte del ecosistema de bibliotecas científicas de python; otras bibliotecas fueron

instaladas en el arranque del contenedor ya que son utilizadas en este trabajo, como son Autosklearn, Pipeline profiler, Shap y Optuna.

La elección de utilizar como entorno de trabajo los cuadernos jupyter dockerizados es porque proporcionan una serie de ventajas:

- Portabilidad: Es fácilmente portable entre diferentes sistemas operativos y entornos de desarrollo, lo que evita problemas de configuración y compatibilidad.
- Reproducibilidad: Se crean imágenes de contenedores que contienen todas las dependencias y configuraciones específicas de tu Jupyter Notebook, lo que facilita la reproducción de resultados y la colaboración.
- Aislamiento: Los paquetes o bibliotecas instalados en el contenedor no afectarán a tu sistema host, y viceversa.
- Escalabilidad: Permite manejar grandes conjuntos de datos y ejecutar cálculos intensivos de manera eficiente utilizando recursos distribuidos.
- Administración simplificada: Docker facilita la gestión de múltiples Jupyter Notebooks y entornos. Puedes crear y administrar diferentes contenedores para proyectos específicos o diferentes configuraciones de entorno.
- Flexibilidad: Permite personalizar y configurar tu entorno para instalar las bibliotecas y herramientas necesarias, configurar extensiones y temas, y ajustar los recursos del contenedor para optimizar el rendimiento.

5.3. Cohorte de datos

La cohorte de datos ha sido proporcionada por el Instituto de Investigación Sanitaria de Valdecilla (IDIVAL) sobre una población de 305 pacientes, cuyas muestras fueron recogidas entre Abril de 2020 y Marzo de 2021. Está constituida por las variables que aparecen en la tabla 5.2

Variable	Descripción
IDInm	Identificación de la muestra
NH	Número de historia clínica

continúa en la página siguiente

continúa desde la página anterior

Variable	Descripción
AntiN	Respuesta anti proteína N de SARS-CoV-2
AntiS	Respuesta anti proteína S de SARS-CoV-2
AntiM	Respuesta anti proteína M de SARS-CoV-2
age	Edad
gender	Género
score	Score de evolución clínica
Line	Timepoint
Timepoint	Pacientes en función de covid y tiempo de seguimiento
IL6	Niveles de interleucina 6 en suero
ferritina	Niveles de ferritina sérica
troponina	Niveles de troponina sérica
LDH	Niveles de lactato deshidrogenasa sérico
PCR	Niveles de proteína C reactiva sérica
procalcit	Niveles de procalcitonina sérica
DimD	Niveles de Dímero D en suero
fechaAnalisis	Fecha de análisis
fechaInicioSintomas	Fecha de Inicio de Síntomas
fechaPCRpos	Fecha de PCR positiva
fechaIngreso	Fecha de Ingreso
fechaIngresoUCI	Fecha de Ingreso UCI
fechaAltaUCI	Fecha de alta en UCI
fechaAlta	Fecha de Alta general
fechaExitus	Fecha de exitus
P3	% de Linfocitos T CD3
P4	% de Linfocitos T CD4
P8	% de Linfocitos T CD8
ratio	Ratio CD4/CD8
P19	% de Linfocitos B CD19
P16	% de Linfocitos NK (CD16CD56)
PNKT	% de Linfocitos NKT (CD3CD16CD56)
LINF ABS	Linfocitos absolutos
3	Linfocitos T CD3 absolutos

continúa en la página siguiente

continúa desde la página anterior

Variable	Descripción
4	Linfocitos T CD4 absolutos
8	Linfocitos T CD8 absolutos
19	Linfocitos B CD19 absolutos
NK	Linfocitos NK absolutos
NKT	Linfocitos NKT absolutos
IgG	Niveles séricos IgG
IgM	Niveles séricos IgM
IgA	Niveles séricos IgA
C3	Niveles séricos C3
C4	Niveles séricos C4
Pneut	% de neutrófilos
Plinf	% de linfocitos
Pmonoc	% de monocitos
Peos	% de eosinófilos
Pbas	% de basófilos
AbsNeut	Neutrófilos absolutos
AbsLinf	Linfocitos absolutos
AbsMonoc	Monocitos absolutos
AbsEos	Eosinófilos absolutos
AbsBas	Basófilos absolutos
TH1	% de Linfocitos TH1
TH117	% de Linfocitos TH1/TH17
TH17	% de Linfocitos TH17
TH2	% de Linfocitos TH2
TC1	% de Linfocitos Tc1
TC117	% de linfocitos Tc1/Tc17
TC17	% de linfocitos Tc17
TC2	% de linfocitos Tc2
THMEM	% de linfocitos Thelper memoria
THMEM1	% de Linfocitos TH1 memoria
THMEM117	% de Linfocitos TH1/TH17 memoria
THMEM17	% de Linfocitos TH17 memoria

continúa en la página siguiente

continúa desde la página anterior

Variable	Descripción
THMEM2	% de Linfocitos TH2 memoria
TCMEM	% de Linfocitos Tc memoria
TCMEM1	% de Linfocitos Tc1 memoria
TCMEM117	% de linfocitos Tc1/Tc17 memoria
TCMEM17	% de linfocitos Tc17 memoria
TCMEM2	% de linfocitos Tc2 memoria
LBCXCR5	% de LB CXCR5
LBCXCR5PD1	% de LB CXCR5 PD1
LTCXCR5	Linfocitos T CXCR5
LTCXCR5PD1	Linfocitos T CXCR5 PD1
MoClasicos	% de Monocitos clásicos
MoIntermedios	% de Monocitos intermedios
MoNOclasicos	% de Monocitos no clásicos
PNK56high16lo	% de linfocitos NK CD56high CD16low
PNK5616high	% de linfocitos NK CD56CD16high (citotóxicos)
CTL27p28p	% de LT citotóxicos $CD27^+CD28^+$
CTL27n28p	% de LT citotóxicos $CD27^-CD28^+$
CTL27p28n	% de LT citotóxicos $CD27^+CD28^-$
CTL27n28n	% de LT citotóxicos $CD27^-CD28^-$
TH27p28p	% de LT helper $CD27^+CD28^+$
TH27n28p	% de LT helper $CD27^-CD28^+$
TH27p28n	% de LT helper $CD27^+CD28^-$
TH27n28n	% de LT helper $CD27^-CD28^-$
CTLDRn38n	% de LT citotóxicos $HLADR^-CD38^-$
CTLDRp38n	% de LT citotóxicos $HLADR^+CD38^-$
CTLDRn38p	% de LT citotóxicos $HLADR^-CD38^+$
CTLDRp38p	% de LT citotóxicos $HLADR^+CD38^+$
THDRn38n	% de LT helper $HLADR^-CD38^-$
THDRp38n	% de LT helper $HLADR^+CD38^-$
THDRn38p	% de LT helper $HLADR^-CD38^+$
THDRp38	% de LT helper $HLADR^+CD38^+$

Tabla 5.2: Variables del dataset

Variable a predecir

La variable a predecir para desarrollar los modelos es el **score** de evolución clínica atendiendo a la gravedad de la enfermedad en el paciente y distinguiendo en *leve* si el paciente es leve y *moderado/grave* en el caso que el paciente tuviese que recibir una asistencia hospitalaria con ventilación mecánica.

Atendiendo a esta distinción, se observa que el conjunto de datos está

bastante balanceado con 73 pacientes leves y 82 moderados/graves.

5.4. Réplica del trabajo original

Como primer acercamiento, se realizó un cuaderno jupyter en python para replicar el trabajo original *Innate and Adaptive Immune Assessment at Admission to Predict Clinical Outcome in COVID-19 Patients*[18]: un modelo de regresión logística basada sólo en 7 variables de las recogidas en la cohorte de datos: age, ferritin, d-dimer, absolute lymphocite count, C4, % de $CD8^+CD27^-CD28^-$ y % of non-classical monocytes.

Primero se preparó el dataset para coincidir con los datos que se utilizan en el trabajo original, ya que se toman muestras de pacientes entre Abril y Octubre de 2020 y que hayan resultado positivos en una prueba PCR.

Una vez tenemos los datos preparados, comprobamos en la figura 5.5 que existen valores nulos en alguna de las variables, por lo que necesitaremos realizar una imputación de esos datos nulos; siguiendo las directrices que marca el trabajo, haremos una imputación de la media.

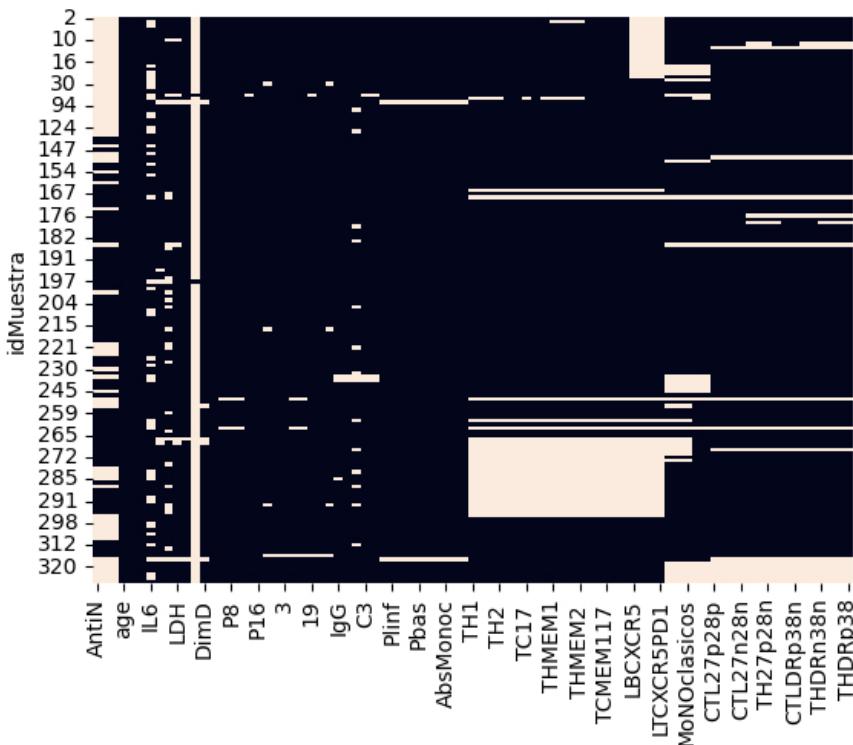


Figura 5.5: Valores nulos en el dataset

Se ha utilizado la validación cruzada con la biblioteca *StratifiedKFold* varias veces, cada vez con una semilla distinta, de forma que el resultado buscado (AUC ROC) es una media de todos los resultados de cada una de las repeticiones que se realizaron. Se han utilizado 5 repeticiones y 3 folds y 10 repeticiones y 10 folds, respectivamente.

En nuestro caso, el mejor resultado obtenido fue con la imputación de la media y 5 repeticiones y 3 folds, con lo que obtuvimos un AUC ROC de **76,36 %**.

Entendemos que la desviación obtenida con el resultado del trabajo original (78 %) viene dada por el hecho de haber utilizado la validación cruzada en nuestra réplica.

El cuaderno *Réplica trabajo* se encuentra en el repositorio del trabajo en Github: [2. Cuadernos TFG](#)

5.5. Búsqueda del clasificador óptimo con *Autosklearn*

En la búsqueda de un clasificador óptimo para la cohorte de datos, se ha utilizado la biblioteca de python *Autosklearn* que es una biblioteca de aprendizaje automático (AutoML) que permite automatizar el proceso de construcción de modelos de aprendizaje automático. En la figura 5.6 se puede observar su funcionamiento.

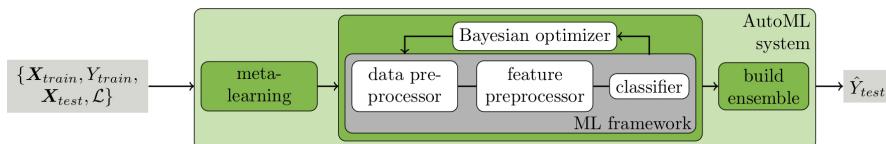


Figura 5.6: Autosklearn en una imagen [4]

Para llevar una correlación con la réplica del apartado anterior, se han tenido en cuenta dos factores a la hora de realizar los cuadernos:

- Utilizar sólo las características de la réplica o todas las características.
- Hacer validación cruzada de 5 o 10 repeticiones con *StratifiedKFold*.

A partir de esta información, se crearon cuatro cuadernos jupyter con la misma estructura, pero alternando los factores anteriores.

La preparación de los datos es similar al cuaderno de réplica de datos, tan solo varía en el caso de escoger todas las variables.

Una vez creados los dataframes de entrenamiento (X) y de test (y), se convierten los datos a un formato compatible con autosklearn y se dividen en subconjuntos aleatorios con la función de sklearn *train_test_split* con un *test_size* de 0.3.

A continuación, se configuraaron los modelos de automl para que realicen una búsqueda del mejor clasificador e hiperparámetros en un tiempo de 1 hora, limitando el tiempo por clasificador en 80 segundos, aplicando una estrategia de resamplado con StratifiedKFold (alternando los splits en 5 y 10 repeticiones) utilizando 4 procesos paralelos y estableciendo como métrica de búsqueda el AUC ROC.

El límite de tiempo ocupado por clasificador de 80 segundos, se estableció atendiendo a las diversas pruebas que se realizaron con autosklearn con anterioridad, observando en los resultados el número de algoritmos perdidos por tiempo.

Estos fueron los resultados obtenidos en los 4 cuadernos:

1. Con características de la réplica:

- 10 repeticiones: **5.7**

Clasificador: Passive_aggressive

AUC ROC: 82,25 %

```
Clasificador: passive_aggressive
ROC_AUC: 0.8225308641975309
: {'balancing:strategy': 'weighting',
 'classifier:_choice_': 'passive_aggressive',
 'data_preprocessor:_choice_': 'feature_type',
 'feature_preprocessor:_choice_': 'pca',
 'classifier:passive_aggressive:C': 3.4924618602130666,
 'classifier:passive_aggressive:average': 'False',
 'classifier:passive_aggressive:fit_intercept': 'True',
 'classifier:passive_aggressive:loss': 'hinge',
 'classifier:passive_aggressive:tol': 0.02742707204783625,
 'data_preprocessor:feature_type:numerical_transformer:imputation:strategy': 'mean',
 'data_preprocessor:feature_type:numerical_transformer:rescaling:_choice_': 'power_transformer',
 'feature_preprocessor:pca:keep_variance': 0.570116377723543,
 'feature_preprocessor:pca:whiten': 'False'}
```

Figura 5.7: Resultado experimento réplica con 10 repeticiones

- 5 repeticiones: **5.8**

Clasificador: Extra_trees

AUC ROC: 79,83 %

```

Clasificador: extra_trees
ROC_AUC: 0.7983585858585858

: {'balancing:strategy': 'weighting',
 'classifier:_choice_': 'extra_trees',
 'data_preprocessor:_choice_': 'feature_type',
 'feature_preprocessor:_choice_': 'pca',
 'classifier:extra_trees:bootstrap': 'False',
 'classifier:extra_trees:criterion': 'gini',
 'classifier:extra_trees:max_depth': 'None',
 'classifier:extra_trees:max_features': 0.06257991893936973,
 'classifier:extra_trees:max_leaf_nodes': 'None',
 'classifier:extra_trees:min_impurity_decrease': 0.0,
 'classifier:extra_trees:min_samples_leaf': 17,
 'classifier:extra_trees:min_samples_split': 6,
 'classifier:extra_trees:min_weight_fraction_leaf': 0.0,
 'data_preprocessor:feature_type:numerical_transformer:imputation:strategy': 'mean',
 'data_preprocessor:feature_type:numerical_transformer:rescaling:_choice_': 'power_transformer',
 'feature_preprocessor:pca:keep_variance': 0.6104375056103275,
 'feature_preprocessor:pca:whiteen': 'False'}

```

Figura 5.8: Resultado experimento réplica con 5 repeticiones

2. Todas las características:

- 10 repeticiones: **5.9**

Clasificador: Gradient_boosting**AUC ROC:** 82,07 %

```

Clasificador: gradient_boosting
ROC_AUC: 0.820679012345679

: {'balancing:strategy': 'weighting',
 'classifier:_choice_': 'gradient_boosting',
 'data_preprocessor:_choice_': 'feature_type',
 'feature_preprocessor:_choice_': 'no.preprocessing',
 'classifier:gradient_boosting:early_stop': 'off',
 'classifier:gradient_boosting:l2_regularization': 1.505819024921544e-08,
 'classifier:gradient_boosting:learning_rate': 0.11722534056513453,
 'classifier:gradient_boosting:loss': 'auto',
 'classifier:gradient_boosting:max_bins': 255,
 'classifier:gradient_boosting:max_depth': 'None',
 'classifier:gradient_boosting:max_leaf_nodes': 34,
 'classifier:gradient_boosting:min_samples_leaf': 1,
 'classifier:gradient_boosting:scoring': 'loss',
 'classifier:gradient_boosting:tol': 1e-07,
 'data_preprocessor:feature_type:numerical_transformer:imputation:strategy': 'median',
 'data_preprocessor:feature_type:numerical_transformer:rescaling:_choice_': 'robust_scaler',
 'data_preprocessor:feature_type:numerical_transformer:rescaling:robust_scaler:q_max': 0.8041525977058757,
 'data_preprocessor:feature_type:numerical_transformer:rescaling:robust_scaler:q_min': 0.2629890783770267}

```

Figura 5.9: Resultado experimento con todas las caract. y 10 repeticiones

- 5 repeticiones: **5.10**

Clasificador: Lda**AUC ROC:** 83,78 %

```

Clasificador: lda
ROC_AUC: 0.8377525252525253

: {'balancing:strategy': 'weighting',
 'classifier:_choice_': 'lda',
 'data_preprocessor:_choice_': 'feature_type',
 'feature_preprocessor:_choice_': 'select_percentile_classification',
 'classifier:lda:shrinkage': 'auto',
 'classifier:lda:tol': 6.0032858147419916e-05,
 'data_preprocessor:feature_type:numerical_transformer:imputation:strategy': 'median',
 'data_preprocessor:feature_type:numerical_transformer:rescaling:_choice_': 'robust_scaler',
 'feature_preprocessor:select_percentile_classification:percentile': 14.37566755814856,
 'feature_preprocessor:select_percentile_classification:score_func': 'mutual_info',
 'data_preprocessor:feature_type:numerical_transformer:rescaling:robust_scaler:q_max': 0.9463407580961114,
 'data_preprocessor:feature_type:numerical_transformer:rescaling:robust_scaler:q_min': 0.26267480188270903}

```

Figura 5.10: Resultado experimento con todas las caract. y 5 repeticiones

Una vez reajustado el modelo, se utilizó la biblioteca *Pipeline profiler* 5.11 para comparar los resultados obtenidos en cada experimento realizado.

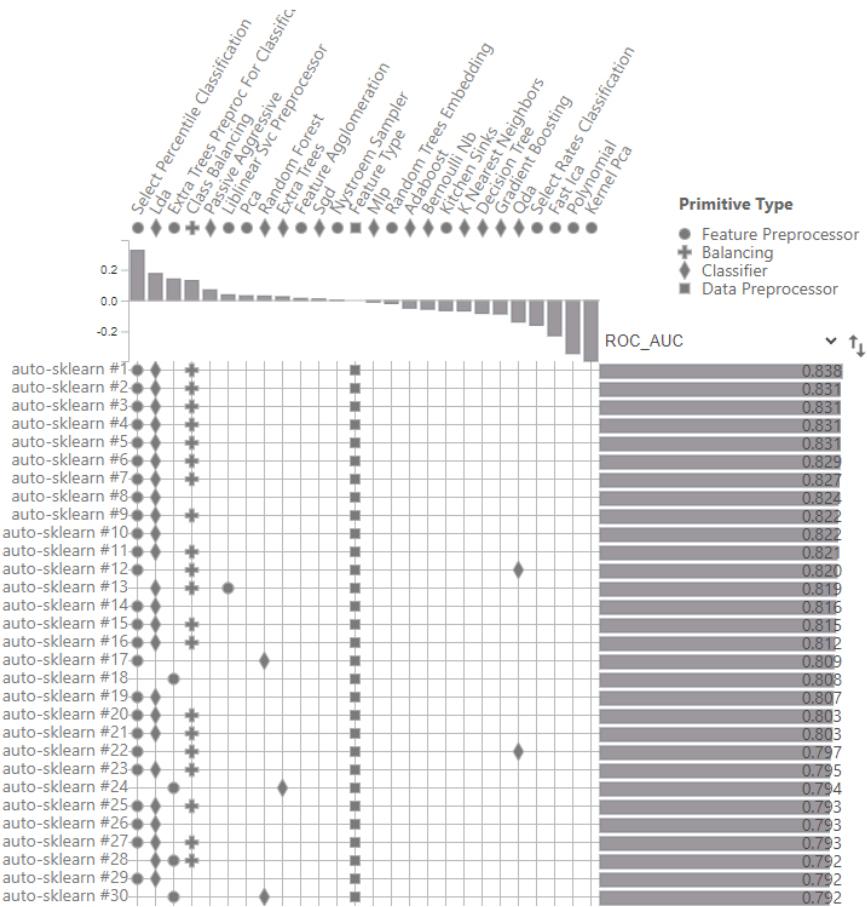


Figura 5.11: Pipeline profiler experimento con todas las caract. y 5 repeticiones

Los cuadernos de *Autosklearn* se encuentran en el repositorio del trabajo en Github: [2. Cuadernos TFG](#)

5.6. Optimización de hiperparámetros

En este apartado, se utiliza la biblioteca Optuna para realizar la optimización de hiperparámetros del modelo LDA (Análisis de Discriminante Lineal). Se define la función objetivo que toma los hiperparámetros sugeridos por Optuna y entrena el modelo en el conjunto de entrenamiento, luego se evalúa su rendimiento en el conjunto de prueba. Optuna realiza la búsqueda de hiperparámetros y guarda los mejores hiperparámetros encontrados.

Además, se utiliza la biblioteca Shap para realizar una exploración de importancia de características del modelo entrenado [5.12](#). Se crea un objeto Explainer con el mejor modelo y se calculan los valores SHAP para el conjunto de prueba. Finalmente, se visualiza la importancia

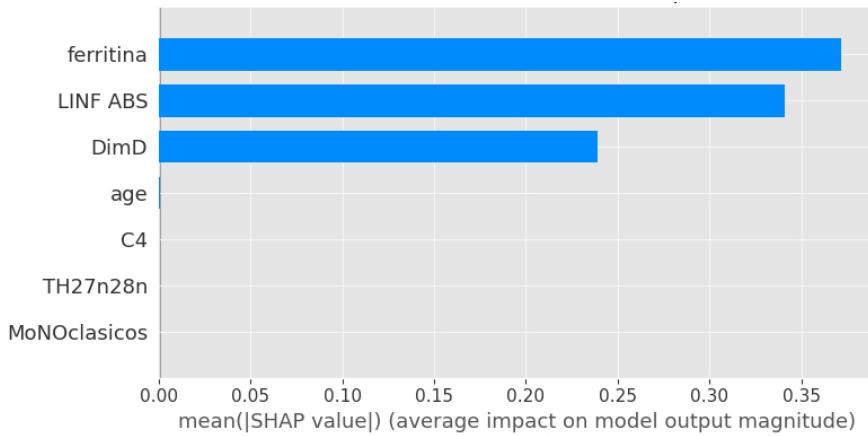


Figura 5.12: Contribución media de las características de cada clase

Los cuadernos de *Optimización de hiperparámetros* se encuentran en el repositorio del trabajo en Github: [2. Cuadernos TFG](#)

Trabajos relacionados

6.1. Búsqueda de trabajos relacionados

Investigación en los principales buscadores de artículos científicos acerca de documentos publicados relacionados con palabras como COVID-19, clasificación, aprendizaje automático, triaje y score. De entre todos, se discriminaron por los artículos que se basaran en *datasets* que contuviesen marcadores recogidos a los pacientes para el diagnóstico del COVID-19.

En la tabla 6.3, se enumeran el listado de los trabajos relacionados con la fecha de publicación y una breve descripción.

Título	Fecha	Descripción
Innate and Adaptive Immune Assessment at Admission to Predict Clinical Outcome in COVID-19 Patients [18]	29/07/2021	Predicción de la severidad de la enfermedad en la admisión del paciente categorizada por los requisitos de oxigenoterapia
Machine-learning-based COVID-19 mortality prediction model and identification of patients at low and high risk of dying [2]	08/09/2021	Modelo de predicción de mortalidad para pacientes COVID-19 hospitalizados, así como una clasificación de pacientes para verificar los grupos de bajo y alto riesgo

continúa en la página siguiente

continúa desde la página anterior

Título	Fecha	Descripción
Deep forest model for diagnosing COVID-19 from routine blood tests [1]	17/08/2021	Uso de técnicas de machine learning basadas en datos clínicos y/o de laboratorio para la detección temprana de COVID-19
Development of a severity of disease score and classification model by machine learning for hospitalized COVID-19 patients [15]	21/04/2021	Modelo de machine learning para identificar de manera temprana a los pacientes que morirán o requerirán ventilación mecánica durante la hospitalización a partir de las características clínicas y de laboratorio obtenidas en el momento del ingreso
Machine learning based predictors for COVID-19 disease severity [16]	25/02/2021	Clasificación mediante random forest de la severidad en los pacientes con COVID-19 para calcular si necesitarán ventilación mecánica o/y ingreso en UCI
Prediction model and risk scores of ICU admission and mortality in COVID-19 [23]	30/07/2020	Desarrolla scores basados en las características clínicas en el momento de la admisión para predecir el ingreso y la mortalidad en UCI en pacientes con COVID-19
Scoring systems for predicting mortality for severe patients with COVID-19 [20]	15/07/2020	Definición de sistemas de score para predecir el ingreso en UCI y/o la mortalidad en pacientes de COVID-19
A Multimodality Machine Learning Approach to Differentiate Severe and Nonsevere COVID-19: Model Development and Validation [3]	07/04/2021	Uso de machine learning para diferenciar con precisión los tipos clínicos de COVID-19 graves y no graves en función de múltiples características médicas y proporcionar predicciones fiables del tipo clínico de la enfermedad

continúa en la página siguiente

continúa desde la página anterior

Título	Fecha	Descripción
Multivariable mortality risk prediction using machine learning for COVID-19 patients at admission (AICOVID) [10]	17/06/2021	Desarrollar y validar puntuaciones de riesgo de mortalidad individualizadas basadas en los datos clínicos y de laboratorio anonimizados en el momento del ingreso y determinar la probabilidad de Muertes a los 7 y 28 días
A systematic review of prediction models to diagnose COVID-19 in adults admitted to healthcare centers [14]	18/06/2021	Identificar, comparar y evaluar el desempeño de los modelos de predicción para el diagnóstico de COVID-19 en pacientes adultos en un entorno de atención médica
Machine learning is the key to diagnose COVID-19: a proof-of-concept study [6]	30/03/2021	Desarrollar y evaluar modelos de aprendizaje automático utilizando datos clínicos y de laboratorio de rutina para mejorar el rendimiento de RT-PCR y CT de tórax para el diagnóstico de COVID-19 entre pacientes hospitalizados después de una urgencia
Comparing machine learning algorithms for predicting ICU admission and mortality in COVID-19 [21]	21/05/2021	Compara el rendimiento de 18 algoritmos de aprendizaje automático para predecir la admisión y la mortalidad en la UCI entre pacientes con COVID-19
Rapid identification of SARS-CoV-2-infected patients at the emergency department using routine testing [11]	29/06/2020	Desarrollar un algoritmo para evaluar rápidamente el riesgo de una persona de infección por SARS-CoV-2 en el servicio de urgencias

continúa en la página siguiente

continúa desde la página anterior		
Título	Fecha	Descripción
Machine Learning for Mortality Analysis in Patients with COVID-19 [19]	12/11/2020	Modelos de ML para encontrar reglas de decisión interpretables para estimar el riesgo de mortalidad de los pacientes que se pueden obtener del árbol de decisiones y que puede ser crucial en la priorización de la atención y los recursos médicos

Tabla 6.3: Trabajos relacionados con este proyecto

6.2. Clasificación de variables, técnicas y herramientas

Una vez escogidos los trabajos sobre los que se basa la investigación para este proyecto, aparece la necesidad de clasificar de cada trabajo qué variables, técnicas y herramientas utilizaron los diversos autores en su investigación. De esta manera, se da la posibilidad de ver qué variables son las más usadas en las investigaciones, así como qué técnicas más se repiten y cuáles son las herramientas que se utilizan.

En la siguiente tabla 6.4 se expone qué variables con qué técnicas y herramientas se utilizaron en los trabajos relacionados.

Variables	Técnicas	Herramientas
Age, Ferritin, D-dimer, C4, % de $CD8^+CD27^-CD28^-$, Suma absoluta de Linfocitos, % de Monocitos no clásicos	Regresión Logística	GraphPad (Prism)
AST, WBC, Lymphocytes, Neutrophils, GGT, AGE, Basophils, Eosinophils, ALT, Platelets, gender, CRP, ALP, LDH, Monocytes	Extra trees, XG-Boost, LightGBM, SHapley Additive exPlanations (SHAP)	Python (sklearn)

continúa en la página siguiente

continúa desde la página anterior

Variables	Técnicas	Herramientas
Old age, coronary heart disease (CHD), percentage of lymphocytes (LYM %), procalcitonin (PCT), D-dimer (DD)	LASSO regression, Python Roc curve	
demographic variables (including age and sex), individual comorbidities and Charlson Comorbidity Index, chronic medical treatment, clinical characteristics, physical examination parameters, biochemical parameters	Random forest, Xg-boost, Logistic regression, AUC	Python (sklearn, xgboost, eli5)
Leukocytes, Neutrophils, Lymphocytes, Eosinophil, Hemoglobin, Hematocrit, Platelets, ESR, BUN, Creatinine, Na, K, Ferritin, CRP, PCT, Lactate, Troponin, CK, BNP, LDH, Fibrinogen, ALT, AST, Albumin, D-dimer, Bilirubin, Prothrombin, APTT, pH, PaCO ₂ , FiO ₂ _lab, Bicarbonate, CAD, Diabetes, Age >65, AMS, Dementia, Nursing home, Q2 saturation <88, yno2, Consolidation, Hypertension, Atrial fibrillation, Alcohol, Chest pain, Peripheral vascular disease, Stroke, Headache, Dyspnea, CRP, Lactate, Smoking	SIMPLS, AUC (Area under ROC curve), Decission Tree	Python

continúa en la página siguiente

continúa desde la página anterior		
Variables	Técnicas	Herramientas
age, sex, travel, contact history, and co-morbidities, fever, dyspnea, respiratory rate, and blood oxygen saturation (SpO ₂), RT-PCR, InterLeukin-6, D-Dimer, complete blood count, lipase, and C-reactive protein (CRP)	Random Forest, multilayer perceptron, support vector machines, gradient boosting, extra tree classifier, adaboost, Regresión, Validación cruzada quíntuple, AUC	Python
lactate dehydrogenase, procalcitonin, pulse oxygen saturation, smoking history, lymphocyte count, heart failure, chronic obstructive pulmonary disease, heart rate, age	Regresión Logística, AUC	TRIPOD
age, hypertension, cardiovascular disease, gender, diabetes, dimerized plasmin fragment D, high sensitivity troponin I, absolute neutrophil count, interleukin 6, lactate dehydrogenase	Random Forest, Predictive mean matching (PMM), Regresión Logística, AUC	R, Python
Age, Male Gender, Respiratory Distress, Diabetes Mellitus, Chronic Kidney Disease, Coronary Artery Disease, respiratory rate >24/min, oxygen saturation below 90 %, Lymphocyte % in DLC, INR, LDH, Ferritin	eXtreme Gradient Boosting (XGB), Random Forest, Regresión Logística, 'Time to event' using Cox Proportional Hazard Model, AUC ROC, Kaplan Meier (KM) Plots	Python

continúa en la página siguiente

continúa desde la página anterior		
Variables	Técnicas	Herramientas
	Prediction model study Risk of Bias Assessment Tool (PROBAST), XG-Boost, Regresión Logística, AUROC	Python
Cough, Hyperthermia, Myalgias, Asthenia, Diarrhea, Confusion, Furosemid (usual treatment), Base excess, Lactates, Red blood cell count, Mean platelet volume, Leukocytes, Neutrophils, Platelet count, Eosinophils percentage, Basophils percentage, Lymphocytes, Monocytes, Ionogram, Potassium, Phosphorus, Alanine aminotransferase, International normalized ratio, D-Dimer	binary logistic regression, random forest, artificial neural network, AUC, K-fold cross-validation	R-Studio (Dplyr, Purrr, missForest, Caret, corrplot, randomForest, neuralnet, pROC)
C-reactive protein, neutrophil percentages, lactate dehydrogenase, first respiratory, lower oxygen saturation, lymphocyte percentages, estimated glomerular filtration rate (eGFR) <60, high neutrophil percentage, high serum potassium, low lymphocyte percentages, high procalcitonin, D-dimer	AdaBoostClassifier, BaggingClassifier, GradientBoostingClassifier, RandomForestClassifier, XGBClassifier, ExtraTreesClassifier, LogisticRegression, DecisionTreeClassifier, LinearDiscriminantAnalysis, QuadraticDiscriminantAnalysis, MLPClassifier, PassiveAggressiveClassifier, Perceptron, LinearSVC, KNeighborsClassifier, GaussianNB	Python

continúa en la página siguiente

continúa desde la página anterior

Variables	Técnicas	Herramientas
Age, Gender, C-reactive protein, lactate dehydrogenase, ferritin, absolute neutrophil, lymphocyte counts	AUROC	Python, Excel 2010
Patient ID, Age and gender, COVID diagnostic (confirmed/pending confirmation), ER date in, ER specialty, ER diagnostic, and destination after ER, First and last constant measurements in the ER (heart rate, temperature, minimum and maximum arterial pressure, O2 saturation in blood), Admission date to the hospital, ICU date in, ICU date out, and number of days in the ICU (if applicable), Discharge date and destination (home/deceased/transferred to other hospital/voluntary discharge/transferred to a socio-sanitary center)	Logistic regression, Survival Analisys, Decision tree, Random Forest, Bayesian networks, Biclustering, AUC	R (Caret, e1701, ggplot2), Python (sklearn, pandas, numpy matplotlib)

Tabla 6.4: Clasificación de variables, técnicas y herramientas

Conclusiones y Líneas de trabajo futuras

Se exponen a continuación las conclusiones que se extraen de la realización de este proyecto, así como las posibles líneas de trabajo que se pueden considerar en un futuro.

7.1. Conclusiones

Relacionadas con los resultados

Los resultados obtenidos en los diferentes modelos realizados son parecidos a los obtenidos por el trabajo que se replicó. Esto se debe en parte a que se realizó sobre una pequeña cantidad de muestras y, por otra parte, a qué dichos datos no estaban del todo completos.

Estos resultados de los modelos se pueden mejorar con una mayor cantidad de muestras recogidas y con una mayor calidad en la recogida de los datos.

Técnicas

En este proyecto se ha realizado un seguimiento experimental riguroso:

- Se realizó un estudio pormenorizado de los trabajos relacionados y la utilización que en ellos se hace de características, técnicas y herramientas, que deparó un mejor conocimiento de la realización de experimentos de aprendizaje automático con marcadores clínicos.
- Se prepararon los datos de la cohorte proporcionada, ya que contenían variables que no aportaban una información clara para nuestra investigación (como las fechas) y algunos valores nulos que fueron subsanados con la imputación de datos.

- La realización de la réplica del trabajo *Innate and Adaptive Immune Assessment at Admission to Predict Clinical Outcome in COVID-19 Patients*[18] permitió comprender el porqué del trabajo original y el modelo de regresión logística aplicado.
- La utilización de autosklearn conllevó un estudio de esta biblioteca de python para poder entender qué valores introducir en el modelo automl que produjeran mejores resultados.
- Con la optimización de hiperparámetros con la biblioteca Optuna se pudo comprender cómo mejorar los experimentos en base a la realización de muchos otros experimentos variando los hiperparámetros.
- Conocimiento de biblioteca Shap para explorar la importancia de características del modelo de aprendizaje automático

7.2. Líneas de trabajo futuras

Como cualquier otro proyecto de índole experimental, una de las líneas de trabajo futuro es continuar con la experimentación. La tecnología avanza de forma imparable cada día y es por esto que, las bibliotecas científicas de software se actualizan y mejoran, las técnicas de aprendizaje automático se depuran y los sistemas dónde ejecutar las herramientas para los experimento no dejan de crecer en rendimiento. Todos estos factores hacen que un experimento realizado, pueda mejorarse de forma sustancial en el tiempo.

Otra línea clara a seguir es la distribución de los modelos para la utilización por parte de cualquier usuario, por ejemplo en un entorno web como se hizo en el artículo *Development of a severity of disease score and classification model by machine learning for hospitalized COVID-19 patients* [15] con la distribución de la página [Herokuapp](#); en esta aplicación, un profesional sanitario puede obtener una ayuda para identificar de manera temprana a los pacientes que morirán o requerirán ventilación mecánica durante la hospitalización a partir de las características clínicas y de laboratorio que debe introducir.

Bibliografía

- [1] Maryam AlJame, Ayyub Imtiaz, Imtiaz Ahmad, and Ameer Mohammed. Deep forest model for diagnosing covid-19 from routine blood tests. *Scientific Reports*, 1(16682), 2021. <https://www-nature-com.ubu-es.idm.oclc.org/articles/s41598-021-95957-w>.
- [2] Mohammad M. Banoei, Roshan Dinparastisaleh, Ali Vaeli Zadeh, and Mehdi Mirsaeidi. Machine-learning-based covid-19 mortality prediction model and identification of patients at low and high risk of dying. *Critical care*, 25(328), 2021. <https://ccforum.biomedcentral.com/articles/10.1186/s13054-021-03749-5>.
- [3] Yuanfang Chen, Liu Ouyang, Forrest S Bao, Qian Li, Lei Han, Hengdong Zhang, Baoli Zhu, Yaorong Ge, Patrick Robinson, Ming Xu, Jie Liu, and Shi Chen. A multimodality machine learning approach to differentiate severe and nonsevere covid-19: Model development and validation. *Journal Medical Internet Research*, 23(4), 2021. <https://doi.org/10.3390/biomedicines9080917>.
- [4] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Springenberg Jost, Blum Manuel, and Frank Hutter. How we made jupyter docker stacks multi-arch, 2020. <https://github.com/automl/auto-sklearn/>.
- [5] Center for Systems Science and Engineering. (s.f.). Covid-19 dashboard by the center for systems science and engineering (csse) at johns hopkins university, 2023. <https://coronavirus.jhu.edu/map.html>.
- [6] Cedric Gangloff, Sonia Rafi, Guillaume Bouzillé, Louis Soulat, and Marc Cuggia. Machine learning is the key to diagnose covid-19: a proof-of-concept study. *Scientific Reports*, 11(7166), 2021. <https://www-nature-com.ubu-es.idm.oclc.org/articles/s41598-021-86735-9>.

- [7] Pablo Haya. Esquema del ciclo crisp-dm est醤dar, 2021. <https://www.iic.uam.es/innovacion/metodologia-crisp-dm-ciencia-de-datos/>.
- [8] Solomon Hykes. Docker, 2013. [https://en.wikipedia.org/wiki/Docker_\(software\)](https://en.wikipedia.org/wiki/Docker_(software)).
- [9] Anaconda Inc. Anaconda distribution, 2012. [https://en.wikipedia.org/wiki/Anaconda_\(Python_distribution\)](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)).
- [10] Sujoy Kar, Rajesh Chawla, Sai Praveen Haranath, Suresh Ramasubban, Nagarajan Ramakrishnan, Raju Vaishya, Anupam Sibal, and Sangita Reddy. Multivariable mortality risk prediction using machine learning for covid-19 patients at admission (aicovid). *Scientific Reports*, 11(12801), 2021. <https://www-nature-com.ubu-es.idm.oclc.org/articles/s41598-021-92146-7>.
- [11] Steef Kurstjens, Armando van der Horst, Robert Herpers, Mick W. L. Geerits, Yvette C. M. Kluiters-de Hingh, Eva-Leonne Göttgens, Martinus J. T. Blaauw, Marc H. M. Thelen, Marc G. L. M. Elisen, and Ron Kusters. Rapid identification of sars-cov-2-infected patients at the emergency department using routine testing. *Clinical Chemistry and Laboratory Medicine (CCLM)*, 58(9), 2020. <https://www.degruyter.com/document/doi/10.1515/cclm-2020-0593/html>.
- [12] Jupyter Labs. Jupyter notebooks, 2014. <https://jupyter-notebook.readthedocs.io/en/latest/>.
- [13] Jupyter Labs. Jupyter docker stacks, 2015. <https://jupyter-docker-stacks.readthedocs.io/en/latest/#>.
- [14] Médéa Locquet, Anh Nguyet Diep, Charlotte Beaudart, Nadia Dardenne, Christian Brabant, Olivier Bruyère, and Anne-Françoise Donneau. A systematic review of prediction models to diagnose covid-19 in adults admitted to healthcare centers. *Archives of Public Health*, 79(105), 2021. <https://archpublichealth.biomedcentral.com/articles/10.1186/s13690-021-00630-3>.
- [15] Miguel Marcos, Moncef Belhassen-García, Antonio Sánchez-Puente, Jesús Sampedro-Gómez, Raúl Azibeiro, P-Ignacio Dorado-Díaz, Edgar Marcano-Millán, Carolina García-Vidal, María-Teresa Moreiro-Barroso, Noelia Cubino-Bóveda, María-Luisa Pérez-García, Beatriz Rodríguez-Alonso, Daniel Encinas-Sánchez, Sonia Peña-Balbuena, Eduardo Sobejano-Fuertes, Sandra Inés, Cristina Carbonell, Miriam López-Parra, Fernanda Andrade-Meira, Amparo López-Bernús, Catalina Lorenzo, Adela Carpio, David Polo-San-Ricardo, Miguel-Vicente Sánchez-Hernández, Rafael Borrás, Víctor Sagredo-Meneses, Pedro-L Sanchez, Alex Soriano,

- and José-Ángel Martín-Oterino. Development of a severity of disease score and classification model by machine learning for hospitalized covid-19 patients. *medRxiv*, 2020. <https://www.medrxiv.org/content/early/2020/07/14/2020.07.13.20150177>.
- [16] Dhruv Patel, Vikram Kher, Bhushan Desai, Xiaomeng Lei, Steven Cen, Neha Nanda, Ali Gholamrezanezhad, Vinay Duddalwar, Bino Varghese, and Assad A Oberai. Machine learning based predictors for covid-19 disease severity. *Scientific Reports*, 11(4673), 2021. <https://doi-org.ubu-es.idm.oclc.org/10.1038/s41598-021-83967-7>.
- [17] Ayaz Salikhov. How we made jupyter docker stacks multi-arch, 2023. <https://dev.to/mathbunnyru/how-we-made-jupyter-docker-stacks-multi-arch-3ic1>.
- [18] David San Segundo, Francisco Arnáiz de las Revillas, Patricia Lamadrid-Perojo, Alejandra Comins-Boo, Claudia González-Rico, Marta Alonso-Peña, Juan Irure-Ventura, José M. Olmos, María C. Fariñas, and Marcos López-Hoyos. Innate and adaptive immune assessment at admission to predict clinical outcome in covid-19 patients. *Biomedicines*, 9(8):917, 2021. <https://doi.org/10.3390/biomedicines9080917>.
- [19] Manuel Sánchez-Montañés, Pablo Rodríguez-Belenguer, Antonio J. Serrano-López, Emilio Soria-Olivas, and Yasser Alakhdar-Mohmara. Machine learning for mortality analysis in patients with covid-19. *International Journal of Environmental Research and Public Health*, 17(22), 2020. <https://www.mdpi.com/1660-4601/17/22/8386>.
- [20] Yufeng Shang, Tao Liu, Yongchang Wei, Jingfeng Li, Liang Shao, Minghui Liu, Yongxi Zhang, Zhigang Zhao, Haibo Xu, Zhiyong Peng, Xinghuan Wang, and Fuling Zhou. Scoring systems for predicting mortality for severe patients with covid-19. *EClinicalMedicine*, 24:100426, 2020. <https://doi.org/10.1016/j.eclinm.2020.100426>.
- [21] Sonu Subudhi, Ashish Verma, Ankit B. Patel, C. Corey Hardin, Melin J. Khandekar, Hang Lee, Dustin McEvoy, Triantafyllos Stylianopoulos, Lance L. Munn, Sayon Dutta, and Rakesh K. Jain. Comparing machine learning algorithms for predicting icu admission and mortality in covid-19. *npj Digital Medicine*, 4(87), 2021. <https://www-nature-com.ubu-es.idm.oclc.org/articles/s41746-021-00456-x>.
- [22] Guido van Rossum. Python, 1991. <https://es.wikipedia.org/wiki/Python>.
- [23] Zirun Zhao, Anne Chen, Wei Hou, James M. Graham, Haifang Li, Paul S. Richman, Henry C. Thode, Adam J. Singer, and Tim Q. Duong. Prediction model and risk scores of icu admission and mortality in covid-

19. *PLoS ONE*, 15(7), 2020. <https://doi.org/10.1371/journal.pone.0236618>.