Modelo basado en Análisis Clínicos rápidos para detectar infección por SARS-CoV-2 usando Inteligencia Artificial

Trabajo de Especialización

Autora:
Ing. Florencia Florio
florioflorencia@gmail.com

Supervisor:
Dr. Marcelo Soria

Buenos Aires, Septiembre de 2020

Maestría en Exploración de Datos y Descubrimiento del Conocimiento

Resumen: Ante el reciente brote del nuevo coronavirus SARS-CoV-2 existió faltante de tests RT-PCR para su detección. Se construyó y analizó un clasificador del resultado del test a partir de análisis clínicos de laboratorio de pacientes sospechosos de COVID-19 utilizando XGBoost. Los pacientes pertenecen al Hospital Israelita Albert Einstein (Brasil) y eran tanto ambulatorios como internados al momento del test. De las técnicas de modelado evaluadas, la que entregó mejores resultados (ROC-AUC = 68.7% ± 3.5%) consistió en un *stacking* de un árbol de decisión simple (RPart) ajustado sobre el subconjunto del 20% más importante de los atributos posteriormente ajustado con XGBoost. Se vió que las variables del hemograma resultaron entre las más importantes para llevar a cabo la clasificación por parte del modelo.

Palabras clave: *SARS-CoV-2, Machine Learning, XGBoost, Screening, Hemograma*

Índice

1 Introducción

COVID-19 es la enfermedad causada por el nuevo coronavirus SARS-CoV-2. Poco más de dos meses pasaron desde el inicio del brote en Wuhan, China en diciembre de 2019 hasta la declaración de "Emergencia sanitaria de preocupación internacional" por parte de la OMS (OMS 2020b). Actualmente, el brote de COVID-19 es una pandemia que afecta a varios paises a nivel mundial (OMS 2020a).

Los síntomas que habitualmente presenta la COVID-19 son fiebre, tos seca y cansancio. Estos suelen ser suaves y comenzar gradualmente. Aproximadamente el 80% de las personas se recupera sin necesidad de tratamiento u hospitalización, mientras que el 20% restante desarrolla un cuadro más grave con dificultad respiratoria (OMS 2020a), requiriendo asistencia médica de mayor o menor complejidad.

La reacción en cadena de la polimerasa con transcriptasa inversa (RT-PCR, del inglés *Reverse transcription polymerase chain reaction*) es actualmente considerado por sociedades científicas como el método más confiable en el diagnóstico de COVID-19 (Hong et al. 2020). Es una prueba mínimamente invasiva que se procesa en un laboratorio de análisis clínicos, normalmente a partir de una muestra de hisopado nasofaríngeo u orofaríngeo.

La pandemia puso en jaque a los sistemas de salud, los cuales no se encontraban preparados para lidiar con este desafío (Deloitte 2020) (Blumenthal et al. 2020) y la abrupta necesidad de recursos impactó en la disponibilidad de tests para el diagnóstico de la enfermedad (Millan, Navarro, and Kueffner 2020)(Pfeiffer, Anderson, and Van Woerkom 2020)(Cramer 2020).

Otros análisis clínicos de laboratorio *in vitro* se realizan habitualmente y son de utilidad (no exclusivamente en el caso de COVID-19) para evaluar la severidad de la enfermedad, definir un pronóstico, dar seguimiento a los pacientes y guiar el tratamiento y monitoreo terapéutico (Lippi and Plebani 2020a). En el caso de los pacientes hospitalizados por COVID-19, parecieran existir ciertos patrones de alteración en los valores de algunos de estos test, como por ejemplo en el hemograma y la fórmula leucocitaria (Lippi and Plebani 2020b).

El propósito del presente trabajo es desarrollar un clasificador del resultado del test para SARS-CoV-2 para casos sospechosos a partir de los resultados de laboratorio. El objetivo del modelo no es realizar un diagnóstico sino hacer *screening* para decidir si realizar o no el test RT-PCR. Se busca que el médico siga haciendo diagnóstico con las herramientas que acostumbra usar pero que aproveche los modelos predictivos para ahorrar recursos que son limitados, especialmente en el contexto de la pandemia.

- Clasificación = Negativo -> No se realiza test RT-PCR para SARS-CoV-2
- Clasificación = Positivo -> Se realiza test RT-PCR para SARS-CoV-2

El set de datos para construir el modelo proviene de un desafío público realizado a través de la plataforma *Kaggle* (Data4u 2020). Corresponden a datos anonimizados de pacientes del Hospital Israelita Albert Einstein de San Pablo, Brasil. Incluye resultados del test RT-PCR para SARS-CoV-2 como así también otros valores de laboratorio. También se indica si el paciente está admitido (internado) o no y en qué tipo de servicio (regular, semi-intensivo o intensivo). No se realiza integración con ningún otro set de datos.

Otro grupo de trabajo (Banerjee et al. 2020) utilizó el mismo dataset con un objetivo similar pero aplicando otros algoritmos, otras variables y entrenados sobre ciertos subconjuntos de la población.

2 Metodología y herramientas

1. Preprocesamiento
2. Análisis exploratorio
3. Construcción de clasificadores
4. Selección del mejor clasificador
5. Evaluación del clasificador seleccionado

2.1 Preprocesamiento

La calidad de los datos se define por su exactitud, completitud y consistencia (García, Luengo, and Herrera 2015). En la realidad, los datasets suelen presentar carencias en estas cualidades. Las técnicas de preprocesamiento de los datos pueden mejorar la calidad de los datos y por consiguiente, mejorar la precisión y eficacia de los procesos de minería (García, Luengo, and Herrera 2015).

En la etapa de preprocesamiento se realizaron las siguientes acciones, algunas de ellas son específicas para el dataset utilizado:

- Eliminación de columnas que son *outcome*
- Eliminación de columnas con todos los datos vacíos
- conversión de valores "" y "not_done" a NA
- Eliminación de variables con un único valor
- Reducción de cardinalidad (máximo 2 categorías) de variables categóricas
- Conversión a numérica de variable "pH en orina"
- *Feature engineering*

Las siguientes acciones fueron llevadas a cabo en función de los experimentos de construcción de clasificadores:

- Reducción de dimensionalidad
- Imputación de datos faltantes por media y moda
- Upsampling/downsampling de la clase minoritaria/mayoritaria

No se aplican otras transformaciones sobre las variables.

2.1.1 Limpieza

El set de datos tiene originalmente 5644 observaciones y 111 columnas. Una de dichas columnas corresponde al identificador del paciente (columna "patient_id"), el cual no se utiliza como atributo para modelar.

La clase de interés en el presente trabajo es el resultado del test para SARS-CoV-2. Las 3 columnas indicativas de la complejidad del servicio de internación no son considerados predictores sino posibles *outcomes* del resultado del test así que se excluyen del análisis y de la construcción del modelo predictivo.

Más allá de las 5 columnas vacías, las cuales son removidas del dataset, no existen columnas iguales entre sí (redundantes). Las variables que sólo adoptan un único valor se remueven por compatibilidad con otras técnicas a ser aplicadas.

Para las tareas de clasificación, es más conveniente que la cardinalidad (cantidad de niveles) de las variables no sea demasiado grande, especialmente cuando los distintos niveles están asociados a muy pocos registros, ya que algunos niveles se podrían perder al hacer las particiones del dataset necesarias para el entrenamiento y el *tuning* de los hiperparámetros.

Se observa que los *Leucocitos en orina* ("urine_leukocytes") tienen una cardinalidad muy alta (31 niveles) y no es clara la conversión a numérica sin el riesgo de introducir errores. Las variables *Cristales en orina* ("urine_crystals"), *Color de la orina* ("urine_color") y *Aspecto de la orina* ("urine_aspect"), por su parte, tienen entre 4 y 5 niveles. La mayoría de los niveles de estas variables poseen muy baja frecuencia por lo que se llevan a 2 niveles. Como regla general, se utiliza 0 para el valor "sano" y 1 para el valor "alterado". En todos los casos los NA permanecen como tales.

Para la determinación de *pH de la orina* ("urine_ph") no quedan dudas sobre los niveles por lo que es posible convertirla a numérica.

2.1.2 Tratamiento de datos faltantes

Este dataset presenta muchos faltantes (ver sección 3.5). Existen diversas estrategias para lidiar con esto (Han, Pei, and Kamber 2011) como por ejemplo:

- Ignorarlos
- Completar manualmente con el valor real
- Completar con una constante (por ejemplo, “Desconocido” o “Infinito”)
- Completar con una medida de tendencia central del atributo
- Completar con el valor más probable (determinado por regresión, inferencia o inducción de un árbol de decisión)

Salvo las dos primeras opciones, todas las estrategias sesgan los datos. El último enfoque utiliza la mayor información posible presente en los datos para predecir los faltantes por lo que existe una mayor chance de que se preserve la relación entre la variable imputada y el resto (Han, Pei, and Kamber 2011). Sin embargo, es demasiado costosa computacionalmente. Se opta entonces por imputación con la media para numéricas y la moda para categóricas.

2.1.3 *Feature engineering*

Se crea una variable booleana llamada “disease_detected” que adopta el valor 1 si al paciente se le detectó alguna infección según se indica en las columnas:

- respiratory_syncytial_virus
- influenza_a
- influenza_b
- influenza_a_rapid_test
- influenza_b_rapid_test
- inf_a_h1n1_2009
- parainfluenza_1
- parainfluenza_2
- parainfluenza_3
- parainfluenza_4
- coronavirusnl63
- coronavirus_hku1
- coronavirus229e
- coronavirusoc43
- rhinovirus_enterovirus
- chlamydophila_pneumoniae
- adenovirus
- bordetella_pertussis
- metapneumovirus

2.1.4 Reducción de dimensionalidad

El clasificador a partir del cual se elige el subconjunto de atributos se obtuvo ajustando un XGBoost con validación cruzada de 10 *folds* (ROC-AUC 0.697). En este caso no se imputaron los faltantes ni se alteró el balance de las clases.

El subconjunto de atributos elegido son los 20 más importantes (de los más de 100 que tenía originalmente) en el modelo anterior. Este subconjunto es luego utilizado para entrenar en los distintos experimentos.

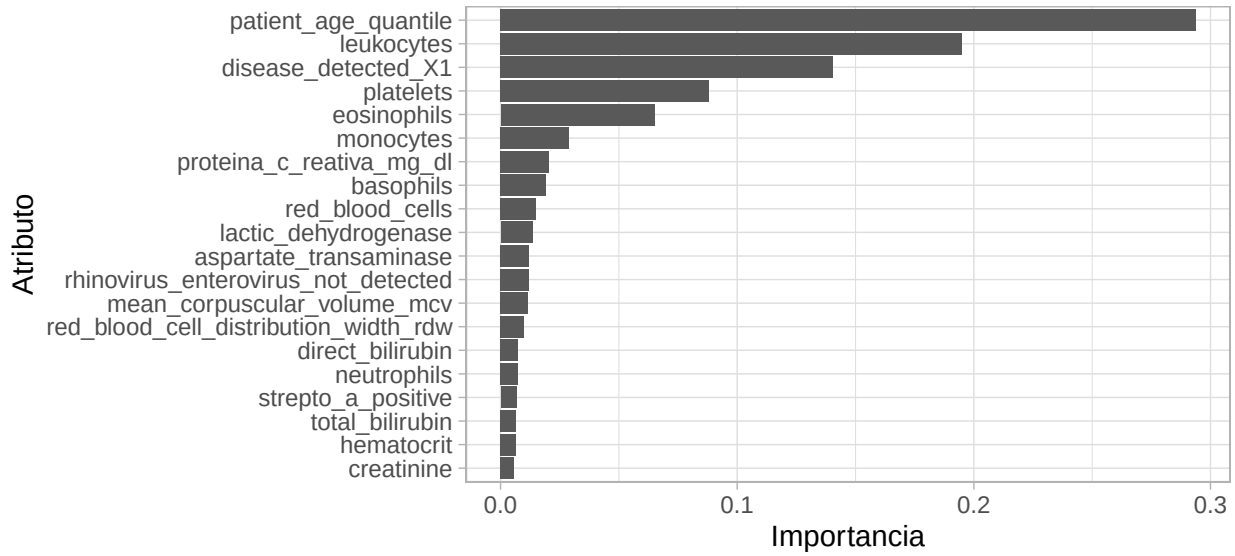

Figura 1: *Los 20 atributos que resultan más importantes en un ajuste de XGBoost con hiperparámetros ajustados.*

2.2 Tipo de clasificador

A grandes rasgos, la inducción de árboles de decisión es una técnica de aprendizaje supervisado en el cual el dataset es particionado recursivamente en subconjuntos más pequeños a medida que se construye el árbol (Han, Pei, and Kamber 2011). Los algoritmos basados en árboles de decisión son robustos a la correlación entre atributos (Chen, He, Benesty, and Tang 2020) y toleran cierto nivel de datos faltantes (Kuhn and Johnson 2019), lo cual reduce el esfuerzo de preparación de los datos que tienen esta característica.

Boosting es un método que aumenta el desempeño del clasificador global realizando un *ensemble* de clasificadores débiles (Han, Pei, and Kamber 2011). En particular, *gradient boosting* (Friedman 2001)(Friedman 2002) busca minimizar en cada iteración una función de pérdida mediante la adición de un árbol de decisión sin modificar los anteriores.

El algoritmo elegido para este trabajo es XGBoost, una implementación de *gradient boosting* de árboles de decisión que, por su eficiencia computacional, es escalable a datasets más grandes (Chen and Guestrin 2016). Emplea regularización para prevenir el *overfitting* (Chen and Guestrin 2016). Estas características lo hacen muy potente al punto que suele ser popular y desempeñarse muy bien en competencias de *Machine Learning* aplicado a diversos tipos de problemas (Chen and Guestrin 2016). Además, puede manejar datos dispersos, clases y variables no lineales.

Además, se compara con una forma de *ensemble* llamada *stacking*, la cual consiste en alimentar un modelo con las predicciones de uno anterior (Wolpert 1992). En este caso, el algoritmo inicial es un árbol de decisión simple y rápido de entrenar de tipo CART (*Classification And Regression Tree*).

2.3 Evaluación y selección del clasificador

Al momento de evaluar un clasificador, es importante estimar cómo será su desempeño en datos que nunca ha visto. El hecho de contar con un mecanismo confiable y objetivo para llevar a cabo esta evaluación permite luego comparar varios clasificadores para elegir el que mejor realiza la tarea de clasificación.

En función del objetivo y el uso que se le vaya a dar a un modelo, existen otros aspectos a evaluar además de su desempeño clasificador, como por ejemplo: velocidad, robustez, escalabilidad, interpretabilidad (Han, Pei, and Kamber 2011). Estos atributos no se evalúan con rigurosidad.

2.3.1 Métricas de evaluación de desempeño

En sistemas de clasificación de 2 clases existen principalmente los siguientes paradigmas:

- Sensibilidad-Especifidad
- Precisión-*Recall* (paradigma de recuperación de información)

En medicina se suele usar Sensibilidad-Especificidad aunque también Precisión-*Recall* pero es menos frecuente (Han, Pei, and Kamber 2011). En el presente trabajo se utilizará el primero de los enfoques.

Estas medidas se definen como:

$$Sensibilidad = \text{Fracción de verdaderos positivos} = \frac{VP}{P} = \frac{VP}{VP + FN}$$

Donde VP, son los verdaderos positivos; y FN, los falsos negativos. Sensibilidad y *Recall* son distintos nombres para el mismo concepto.

$$Especificidad = \text{Fracción de verdaderos negativos} = \frac{VN}{N} = \frac{VN}{VN + FP}$$

Donde VN, son los verdaderos negativos; y FP, los falsos positivos.

El modelo perfecto tiene Sensibilidad y Especificidad iguales a 1 y logra clasificar correctamente todos los positivos y los negativos. El escenario ideal es infrecuente y suele observarse los valores son menores a 1, existiendo además un compromiso entre ambas métricas.

La Sensibilidad y Especificidad dependen de tener predicciones “duras”, es decir “positivo” o “negativo”. La mayoría de los clasificadores producen probabilidades de clases que pueden ser convertidas a una clase definitiva eligiendo la clase que posee la probabilidad más grande (Kuhn and Johnson 2019). Esto es útil cuando las clases se encuentran desbalanceadas, ya que el punto de corte óptimo podría no ser 0.5, como implementan por defecto los algoritmos.

La curva ROC, a diferencia de la Sensibilidad y Especificidad, considera todos los posibles puntos de corte sobre la probabilidad al momento de asignar una clase a la predicción. Grafica Sensibilidad vs 1-Especificidad, dando cuenta del compromiso que existe entre ambos valores. Su área bajo la curva (ROC-AUC) resulta útil para la evaluación de modelos (Kuhn and Johnson 2019). Su valor máximo es de 1. A este caso aplica mejor el paradigma de Sensibilidad-Especificidad.

Si bien lo que se busca en este trabajo es tener confianza en la clase “negativo” (alta Especificidad) para poder así evitar realizar un test RT-PCR, esto de ninguna manera se puede lograr a expensas de una alta tasa de falsos positivos (baja Sensibilidad) porque su impacto anularía ampliamente el ahorro de tests. Eligiendo el modelo con ROC-AUC más alto, para un Sensibilidad dada, se va a obtener una Especificidad mayor.

Para el modelo elegido se calcula el nuevo umbral de clasificación, optimizando Sensibilidad y Especificidad buscando el máximo del estadístico J (Youden 1950). El nuevo umbral da lugar a las estimaciones de Sensibilidad y Especificidad. Al igual que como ocurre con la optimización de hiperparámetros, este nuevo punto de corte se obtiene con el set de entrenamiento y se aplica en el conjunto de prueba, como se haría con datos nuevos.

2.3.2 Técnicas de evaluación de desempeño

Evaluar el desempeño del modelo en el mismo conjunto de datos en el que fue entrenado no resulta en una medida confiable de su capacidad de clasificar observaciones nuevas dado que sólo medirá qué tan bien

se ajusta a los datos de entrenamiento. En otras palabras, esto solo da cuenta de cuánto el clasificador "memorizó" los datos de entrenamiento y no de cuánto realmente "aprendió" de ellos.

Por el motivo anterior, para evaluar un modelo se requiere contar con un conjunto de datos que no hayan sido utilizados durante el entrenamiento. Existen varias técnicas para lograr esto: *holdout*, validación cruzada (*cross-validation*), *bootstrap*. Estas técnicas permiten obtener una estimación confiable del desempeño de un modelo.

En este trabajo para estimar la performance se utiliza validación cruzada de 10 iteraciones, un método recomendado debido a su relativamente bajos sesgo y varianza (Han, Pei, and Kamber 2011). Esto significa hacer 10 particiones o *folds* aleatorios y mutuamente excluyentes del dataset y en cada una de las 10 iteraciones seleccionar 1 partición distinta para testeo y las 9 restantes para entrenamiento. Lo anterior resulta en 10 modelos y 10 valores para las métricas de evaluación. Los modelos son utilizados solo para estimar el desempeño y luego son descartados.

Finalmente, se realiza una evaluación en datos nuevos separados antes de la validación cruzada mediante *holdout* del 25% del total. La estimación de performance por validación cruzada debería ser cercana al valor en el set de *holdout*, normalmente llamado conjunto de prueba (Kuhn and Johnson 2019).

Dado que el dataset presenta mucho desbalanceo de clases, tanto el *holdout* inicial como la validación cruzada, se estratifican por el resultado del test SARS-CoV-2 para asegurar que se mantenga la misma proporción de cada clase, "positivo" y "negativo", a lo largo de las distintas particiones.

2.3.3 Ajuste de hiperparámetros

El ajuste o *tuning* de hiperparámetros consiste en encontrar la combinación óptima de valores para el dataset. En este trabajo se realiza mediante optimización bayesiana. Este es un método iterativo: en cada iteración se realiza una validación cruzada de 10 *folds* y se calcula la métrica de performance, en este caso ROC-AUC. Con el correr de las iteraciones, se logran encontrar aquellos parámetros que ajustan un modelo que produzca la mayor ROC-AUC media en los 10 *folds* de la validación cruzada.

Hiperparámetros XGBoost a ajustar:

- trees
- learn_rate
- tree_depth
- min_n
- loss_reduction
- sample_size
- mtry

Hiperparámetros de RPart a ajustar:

- cost_complexity
- tree_depth
- min_n

2.3.4 Experimentos: Modelos a construir

- Base (sin imputación y sin *upsampling* o *downsampling*)
- Imputación por moda (variables categóricas) y media (variables numéricas)
- *Upsampling* de clase minoritaria (es decir, los "positivos")
- *Downsampling* de clase mayoritaria (es decir, los "negativos")
- Combinación de imputación y *upsampling* o *downsampling*

Los anteriores se repiten para tres datasets:

1. Conjunto completo de 100 atributos
2. Subconjunto con los 20 mejores atributos
3. Subconjunto #2 con una columna adicional de predicciones de un RPart (*stacking*).

2.4 Hardware y software

El *hardware* utilizado consiste en una *notebook* con procesador AMD Ryzen 5 3500U (2.10 GHz), 12,0 GB de RAM (9,95 utilizables) con sistema operativo Windows 10 de 64 bits. El lenguaje de programación es R versión 4.0.2. El entorno de trabajo elegido es RStudio versión 1.3.959.

Principales librerías de R utilizadas:

- Tidyverse (Wickham 2019)
- Tidymodels (Kuhn and Wickham 2020)
- Tidyposterior (Kuhn 2020)
- Probably (Kuhn and Vaughan 2020)

Motores para los clasificadores: RPart (Therneau, Atkinson, and Ripley 2019) y XGBoost (Chen, He, Benesty, Khotilovich, et al. 2020).

3 Análisis exploratorio

3.1 Pantallazo general del dataset

El dataset preprocesado posee predominio de variables numéricas continuas respecto a variables categóricas (discretas). A grandes rasgos, el número de observaciones faltantes es muy alto (ver sección 3.5), al punto tal que no existen registros (filas) completos.

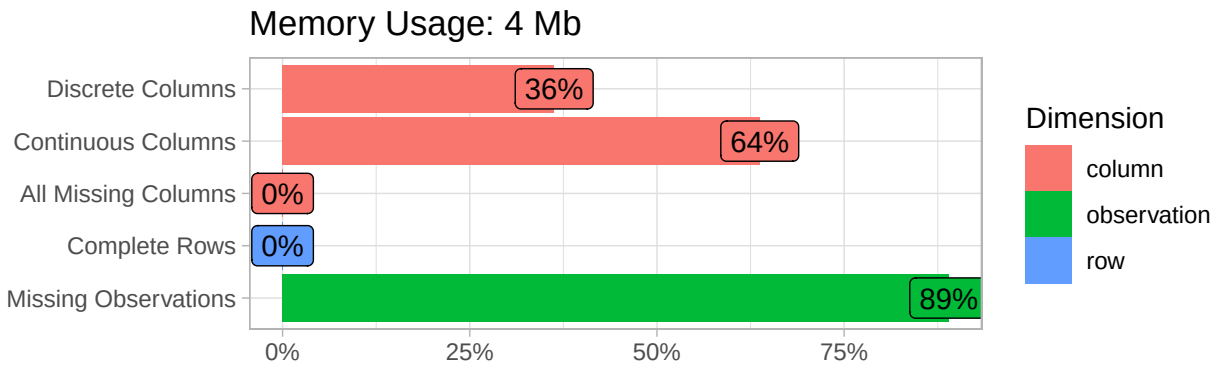

Figura 2: *Resumen del dataset.*

La cantidad de IDs únicos de pacientes, 5644, es igual a la cantidad de registros. Asumiendo que los pacientes fueron identificados unívocamente en origen, se toma lo anterior como validación de que no existen registros duplicados en el set de datos.

Las clases están desbalanceadas. La clase *sars_cov_2_exam_result* = “positivo” es poco frecuente respecto al “negativo”.

Tabla 1: *Distribución de las clases.*

Resultado test SARS-CoV-2 (RT-PCR)	n	Proporción
positive	558	0.099
negative	5086	0.901

3.2 Población

La muestra de pacientes se compone principalmente de pacientes ambulatorios (97%) con algunos pocos internados en servicio regular, unidad de cuidados semi-intensivos o intensivos. La tasa de positividad más alta se encuentra en el grupo de los internados en servicio regular; mientras que la más baja, en el grupo de ambulatorios.

Tabla 2: *Observaciones según admisión*

Tipo de admisión	n
Ambulatorio	5474
Servicio regular	79
Unidad cuidados semi-intensivos	50
Unidad cuidados intensivos	41

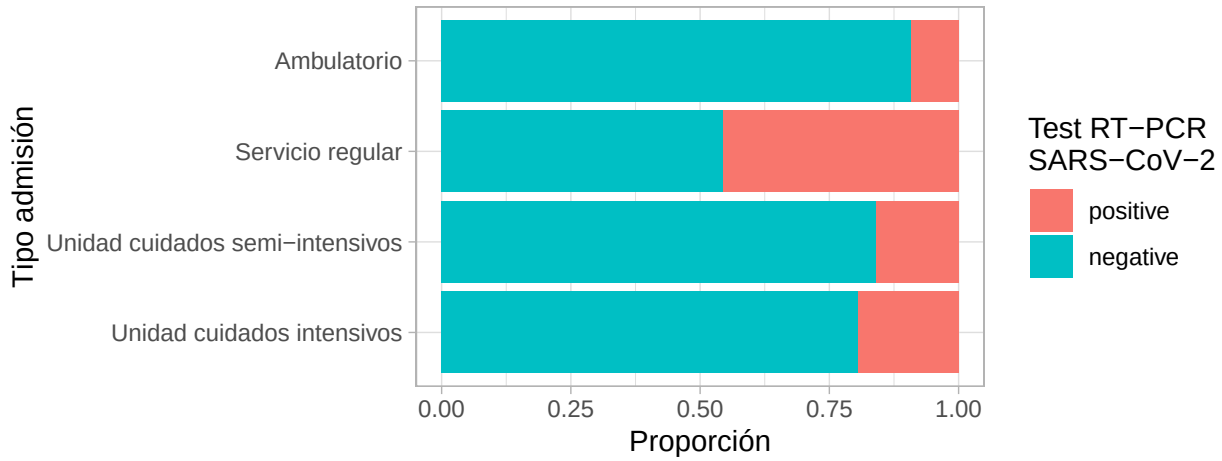

Figura 3: *Positividad del test SARS-CoV-2 (RT-PCR) según tipo de admisión del paciente.*

La variable *Edad del Paciente* se encuentra dividida en 20 cuantiles. Los 3 primeros cuantiles aparentarían tener menor tasa de positividad. A simple vista, en el resto de los cuantiles no parecerían existir diferencias en la tasa de positivos.

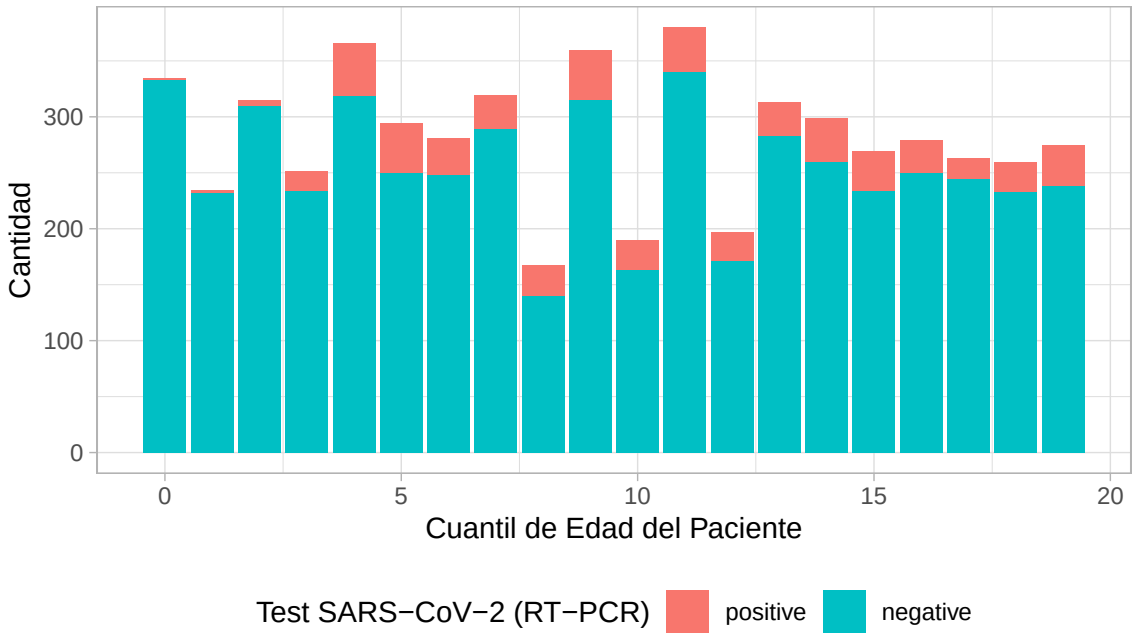

Figura 4: *Positividad del test SARS-CoV-2 (RT-PCR) según cuantil de edad del paciente.*

3.3 Variables numéricas

Los parámetros clínicos numéricos están centrados en media cero (salvo por los cuantiles de edad). En los diagramas de caja según la clase se observa mucha presencia de outliers pero eso no afecta a XGBoost. Según el Test de Wilcoxon no apareado sobre los valores medios de los grupos “positivo” y “negativo”, las siguientes variables resultan tener diferencias estadísticamente significativas ($p <= 0.05$):

- Del hemograma:
  - Hemoglobina
  - Glóbulos rojos
  - Leucocitos
  - Monocitos
  - Eosinófilos
  - Plaquetas
  - Volumen plaquetario medio
- Parámetros hepáticos:
  - Aspartato transaminasa, AST
  - Gamma-glutamil transferasa, GGT
  - Proteina C Reactiva
- Otros parámetros de química clínica:
  - Creatinina
  - Calcio ionizado

Más de la mitad de las variables que resultaron tener diferencias significativas son determinaciones que forman parte del hemograma y la fórmula leucocitaria (MedlinePlus, n.d.). Casi todas ellas se encuentran entre las anormalidades de laboratorio más representativas en COVID-19 (Lippi and Plebani 2020b). A excepción de la hemoglobina, todas las variables del hemograma que se detallaron en el listado de arriba, resultaron también en el top 20 de importancia de variables en XGBoost (ver sección 2.1.4).

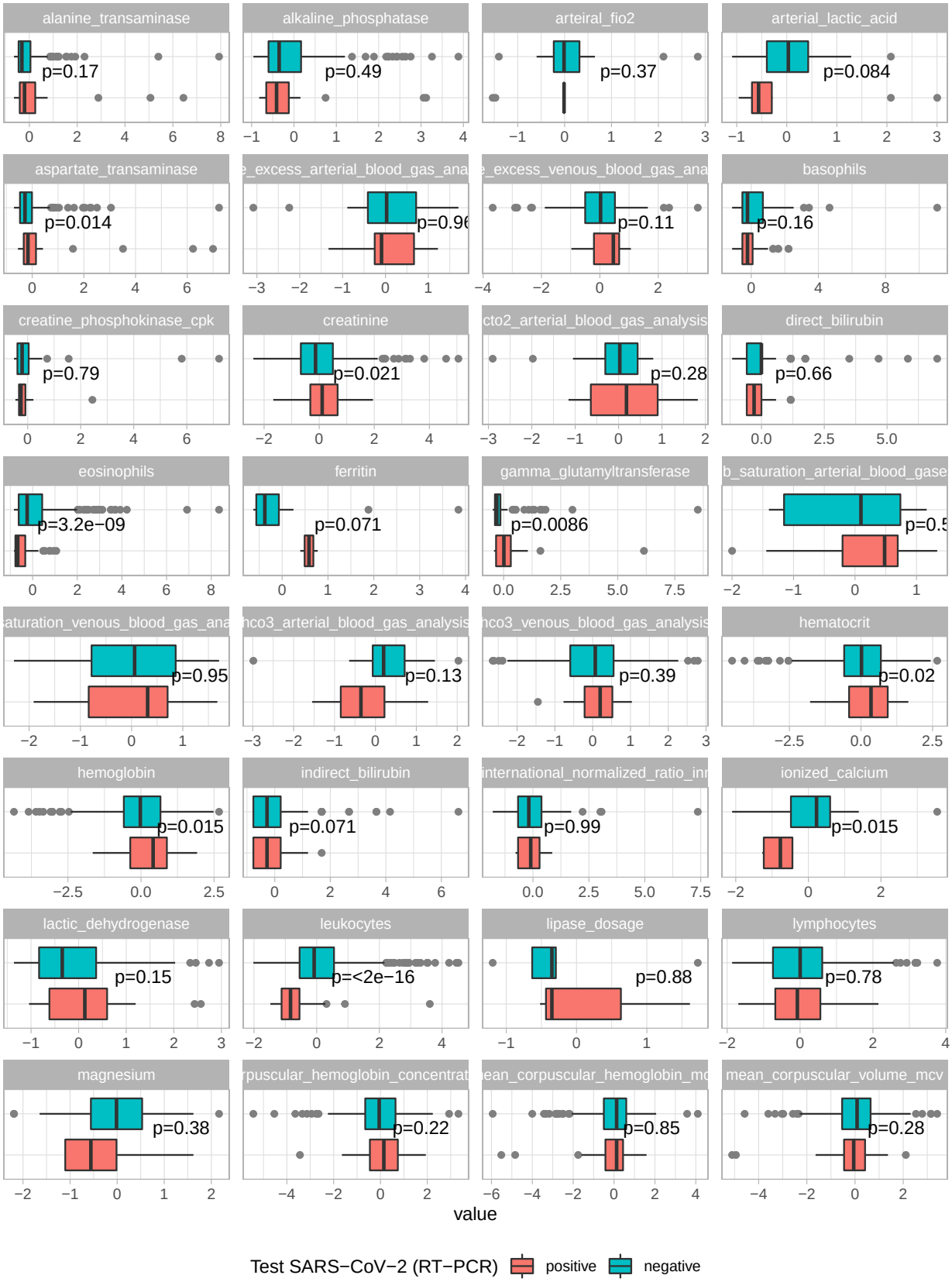

Figura 5: *Diagramas de caja de las variables numéricas según positividad del test SARS-CoV-2 (RT-PCR). En el test de Wilcoxon de las medias de los grupos Positivo y Negativo tiene significancia estadística si p<=0.05.*

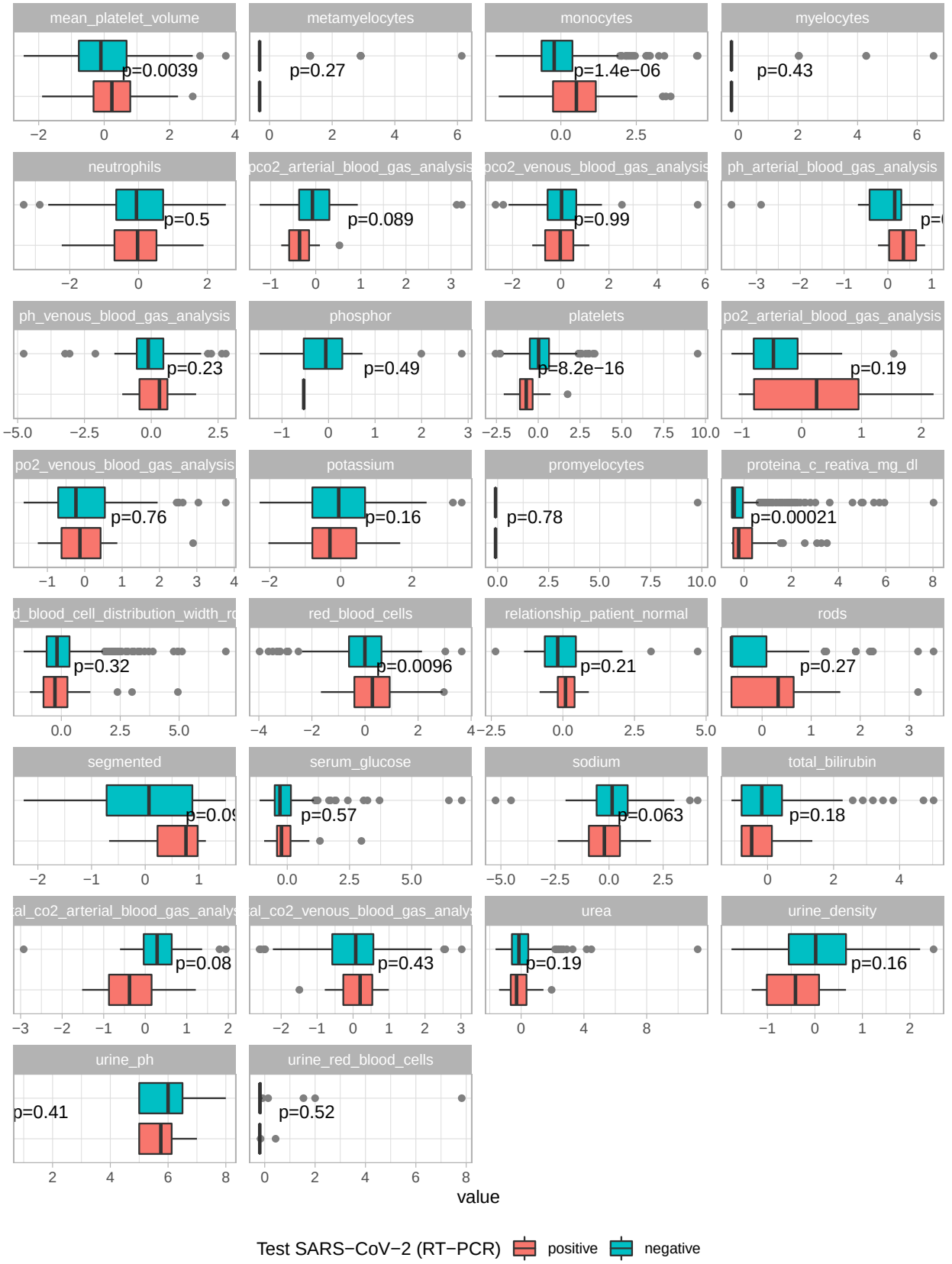

Figura 6: *Diagramas de caja de las variables numéricas según positividad del test SARS-CoV-2 (RT-PCR). En el test de Wilcoxon de las medias de los grupos Positivo y Negativo tiene significancia estadística si p<=0.05.*

3.4 Variables categóricas

En las variables categóricas finales son considerablemente más frecuentes sus valores “sanos”, es decir, “no detectado” o “negativo”.

Tabla 3: *Niveles de las variables categóricas en el dataset preperocesado.*

Variable	Nro. Niveles	Obs. por nivel
adenovirus	2	not: 1339, det: 13
bordetella_pertussis	2	not: 1350, det: 2
chlamydophila_pneumoniae	2	not: 1343, det: 9
coronavirus_hku1	2	not: 1332, det: 20
coronavirus229e	2	not: 1343, det: 9
coronavirusnl63	2	not: 1307, det: 45
coronavirusoc43	2	not: 1344, det: 8
disease_detected	2	0: 4884, 1: 760
inf_a_h1n1_2009	2	not: 1254, det: 98
influenza_a	2	not: 1336, det: 18
influenza_a_rapid_test	2	neg: 768, pos: 52
influenza_b	2	not: 1277, det: 77
influenza_b_rapid_test	2	neg: 771, pos: 49
metapneumovirus	2	not: 1338, det: 14
parainfluenza_1	2	not: 1349, det: 3
parainfluenza_3	2	not: 1342, det: 10
parainfluenza_4	2	not: 1333, det: 19
respiratory_syncytial_virus	2	not: 1302, det: 52
rhinovirus_enterovirus	2	not: 973, det: 379
strepto_a	2	neg: 297, pos: 34
urine_aspect	2	0: 61, 1: 9
urine_color	2	0: 68, 1: 2
urine_crystals	2	0: 65, 1: 5
urine_hemoglobin	2	abs: 53, pre: 16
urine_leukocytes	2	1: 61, 0: 9

La proporción de *sars_cov_2_exam_result* = “positivo” parecería tener, en mayor o menor medida, diferencias entre los distintos niveles de las variables categóricas, salvo para “coronavirus229e”, “coronavirusnl63.” En particular, la variable que fue creada como resumen de la presencia de algún test positivo para otra infección (“disease_detected”), aparentaría estar relacionada con menor proporción de *sars_cov_2_exam_result* = “positivo” en caso valer 1 (alguna infección detectada).

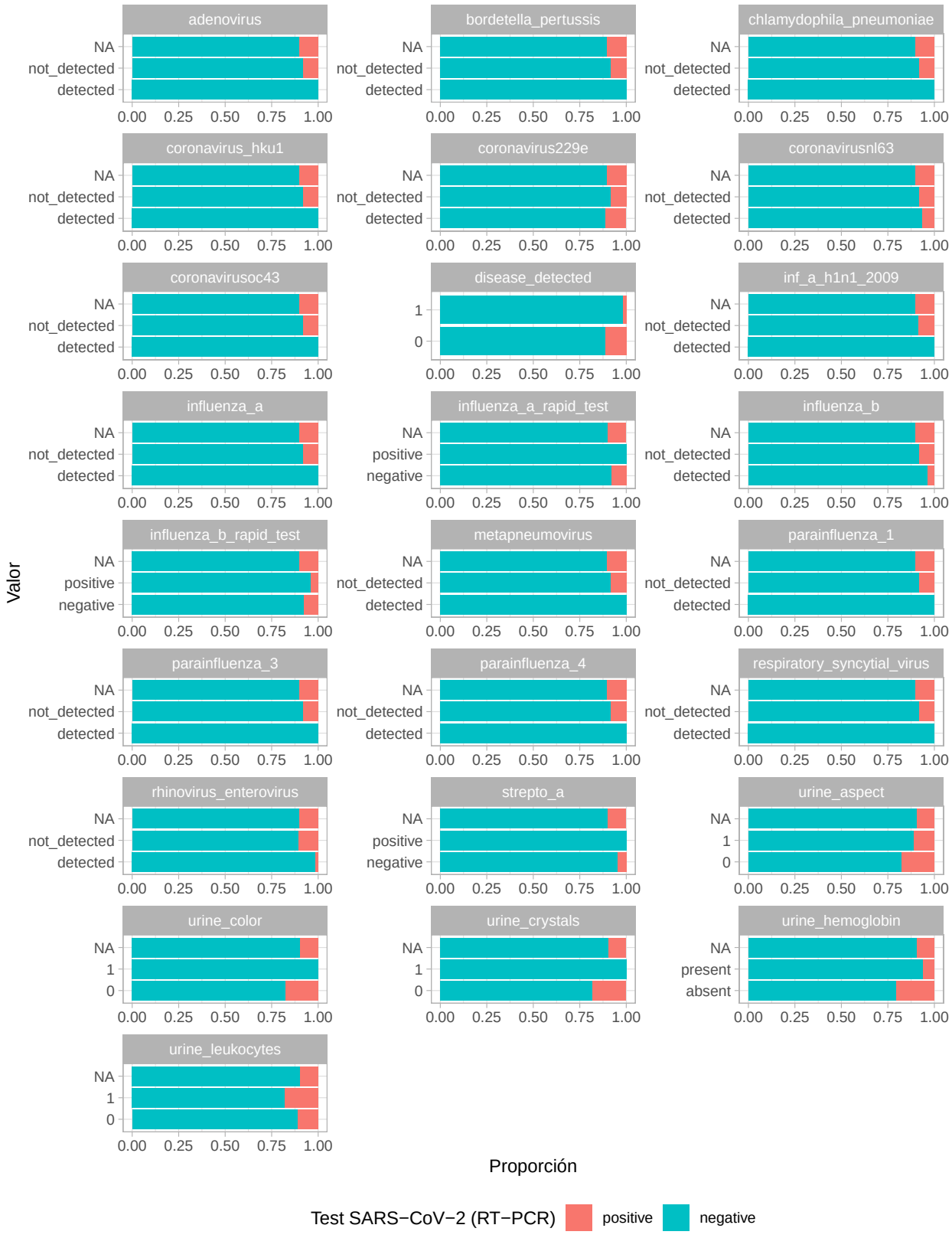

Figura 7: *Positividad del test SARS-CoV-2 (RT-PCR) según nivel de la variable categórica.*

3.5 Datos faltantes

Lo primero que se observa es que la cantidad de valores faltantes es extremadamente alta para todas las variables salvo para la edad y para la variable creada “disease_detected”, las cuales están completas.

Por otra parte, se evidencian conjuntos de variables con porcentajes de faltantes iguales o muy similares. Esto se aprecia claramente para los parámetros del hemograma, del análisis de orina y la pesquisa para otras infecciones. El fenómeno se debe, en el caso del hemograma y la orina, a que las determinaciones se hacen o todas o ninguna según si el médico solicitó o no análisis de sangre u orina.

Los análisis para detectar otras infecciones son el conjunto con menos porcentaje de faltantes, seguido por el hemograma. Los análisis de orina se encuentran ausentes en prácticamente todos los registros al igual que los análisis sobre sangre arterial; posiblemente, en parte, porque estas muestras son más difíciles de obtener.

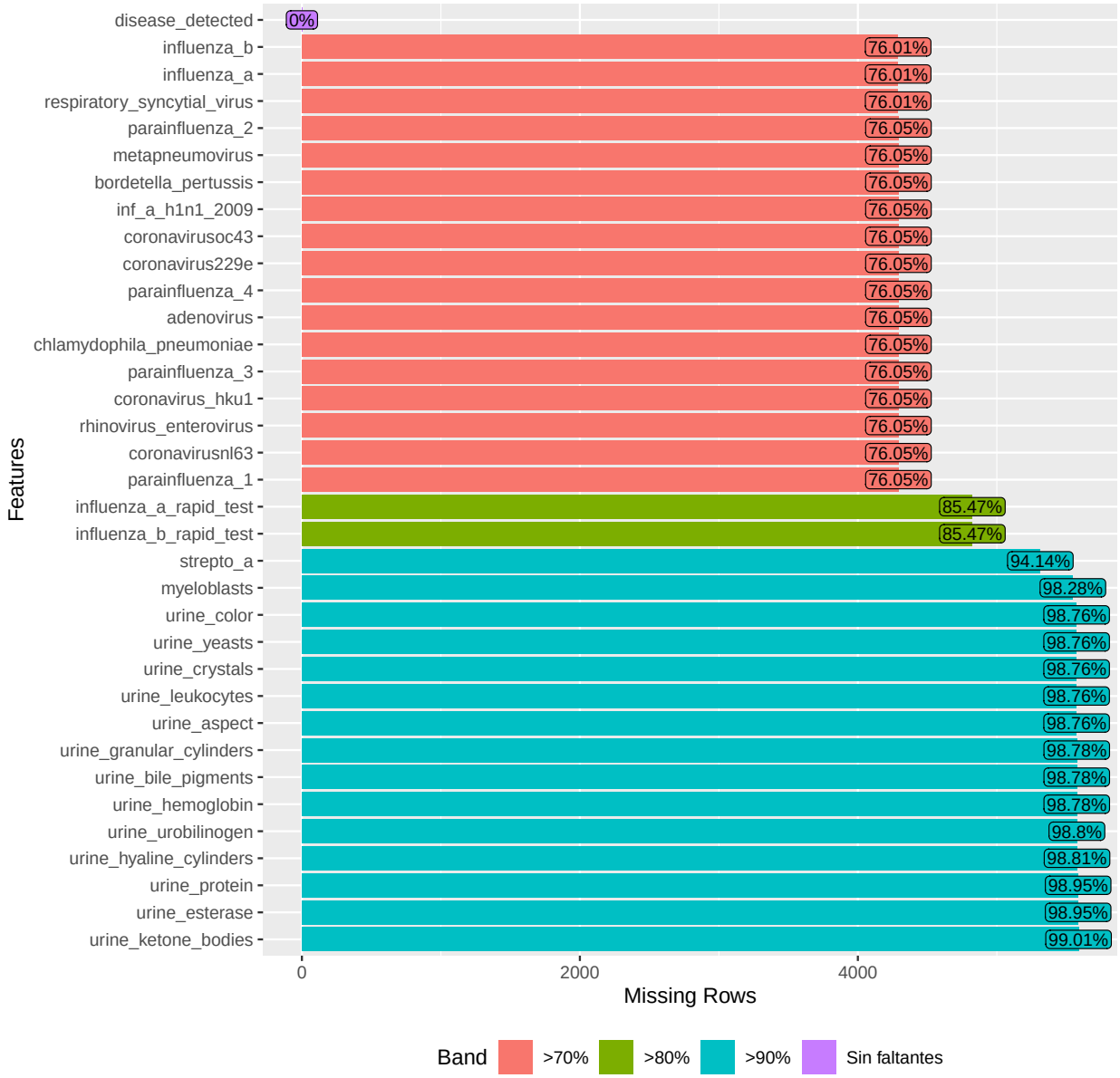

Figura 8: *Porcentaje de valores faltantes en las variables categóricas.*

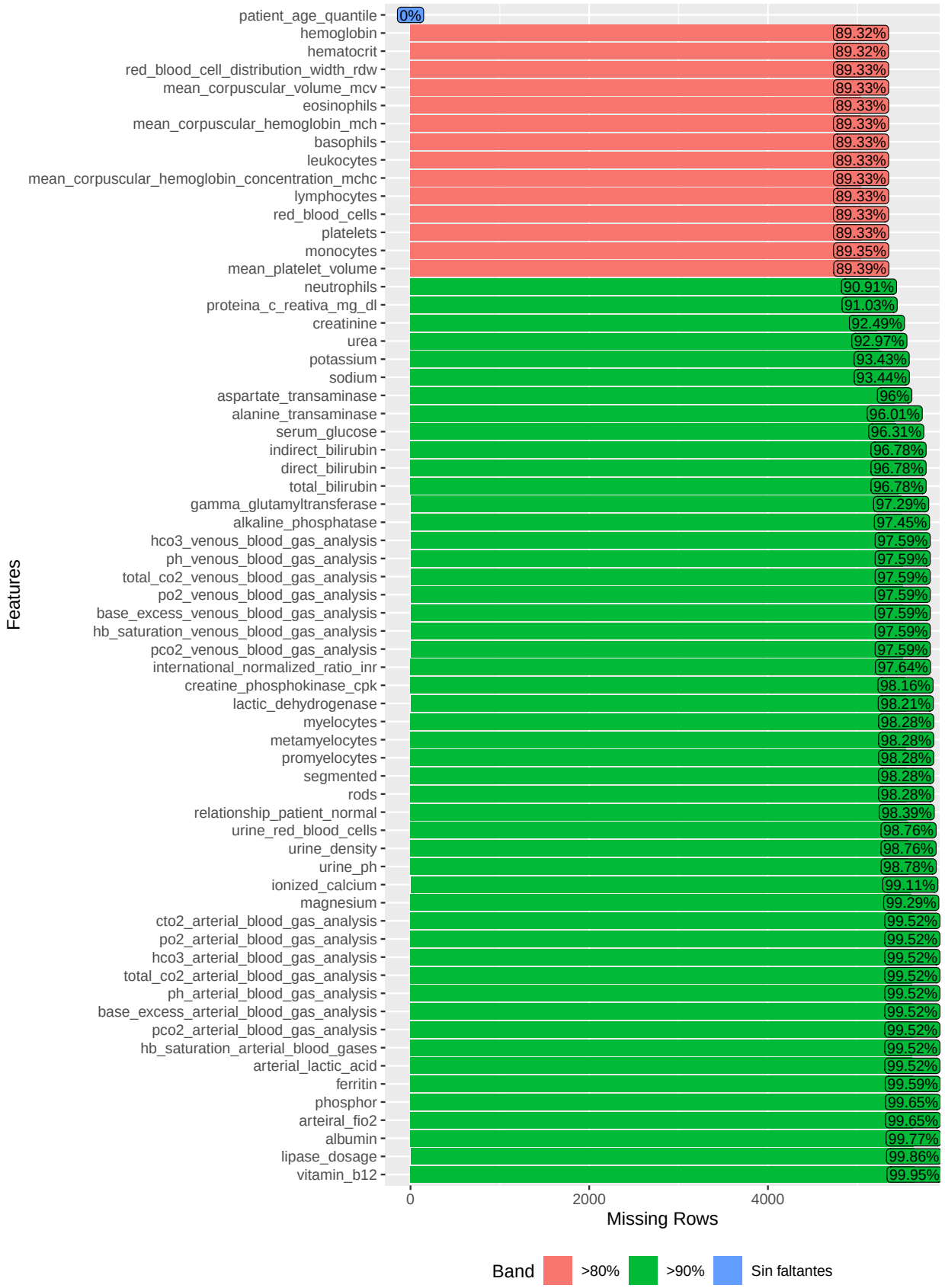

Figura 9: *Porcentaje de valores faltantes en las variables numéricas.*

4 Comparación y selección de modelos

4.1 Experimentos

A continuación se muestran los resultados de las estimaciones de ROC-AUC a través de los experimentos realizados. Los diagramas de caja representan los 10 valores de las métricas obtenidas en 10 iteraciones de validación cruzada. En las secciones siguientes se analizan estos resultados.

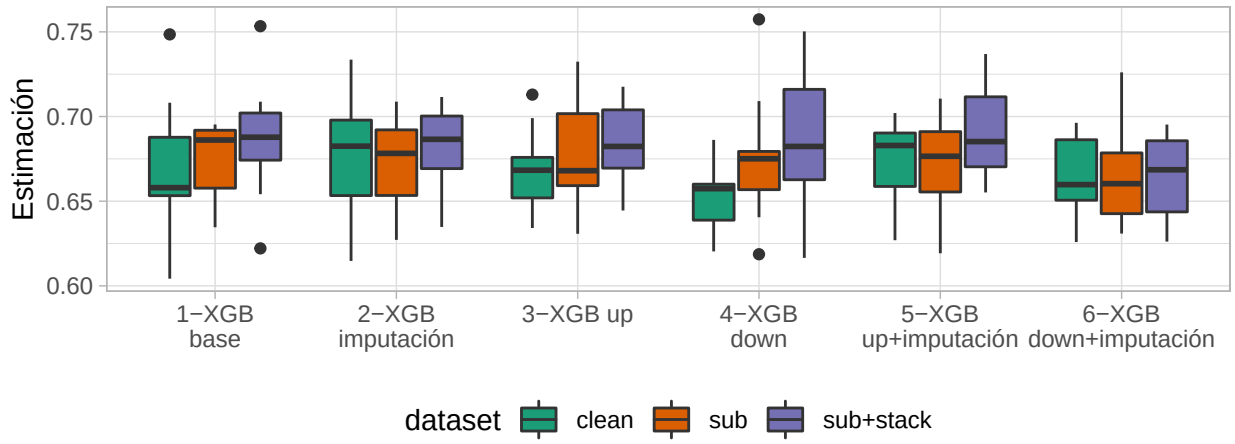

Figura 10: *Estimación de ROC-AUC para los distintos experimentos.*

4.2 Efecto de la reducción de atributos

Entrenar únicamente con los 20 atributos más importantes de un ajuste mediante XGBoost (ver sección 2.1.4), no parece empeorar las estimaciones de ROC-AUC de un XGBoost posterior. Por otra parte, significó una mejora considerable de los tiempos de procesamiento respecto al dataset con todos los atributos. Se decide, entonces, la construcción del *stacking* a partir del dataset reducido en pos, no solo de mejores tiempos de procesamiento, sino también de la obtención de un clasificador más simple y más generalizable.

4.3 Construcción del modelo para *stacking*

Sobre el dataset con los 20 atributos más importantes (ver sección 2.1.4), se ajusta un RPart cuyos parámetros fueron obtenidos mediante optimización bayesiana con validación cruzada de 10 iteraciones. Su ROC-AUC estimada (0.65) es inferior a cualquiera de los ajustes obtenidos con XGBoost.

En el ajuste final del RPart se utiliza únicamente el conjunto de entrenamiento con el objetivo de intentar reducir el *data leakage* en el entrenamiento del XGBoost posterior, alimentado con estas probabilidades. No es deseable que un modelo aprenda con información que, a los efectos de la estimación de performance, es considerada nueva (es decir, el conjunto de prueba) porque podría entregar una estimación optimista y resultar en un modelo subóptimo (Kaufman et al. 2012). Finalmente, la columna de las predicciones de RPart se adjunta al dataset con los 20 atributos.

4.4 Efecto de la imputación de valores faltantes

Al comparar las métricas ROC-AUC de los modelos sin y con imputación (modelos 1 y 2), se observa mucha superposición entre las cajas del mismo dataset (mismo color) y muy poca variación entre sus medianas y

sus valores medios. Lo anterior hace pensar que a simple vista no existe una diferencia real (no azarosa) en las métricas debida a la imputación de faltantes. Se evalúa de manera más objetiva con el análisis de las probabilidades posteriores.

En los contrastes de las probabilidades posteriores se considera que 1 punto porcentual representa una diferencia sustancial entre las estimaciones de performance de los modelos. La columna pract_equiv = 1 significa que toda el área de la distribución de las diferencias se encuentra dentro del rango [-1%, 1%]. Por su parte, pract_pos y pract_neg indican la proporción del área a la derecha o a la izquierda del rango.

Para los 3 datasets, resulta que la imputación tiene mayor probabilidad de ser equivalente (pract_equiv) que de representar una mejora de 1 punto porcentual en ROC-AUC (pract_neg).

Tabla 4: *Contrastes de las distribuciones de probabilidades posteriores de las diferencias en ROC-AUC entre modelos con y sin imputación de faltantes.*

dataset	contrast	pract_pos	pract_equiv	pract_neg
clean	1-XGB base vs 2-XGB imputación	0.03	0.61	0.36
sub	1-XGB base vs 2-XGB imputación	0.05	0.95	0.01
stack	1-XGB base vs 2-XGB imputación	0.14	0.83	0.02

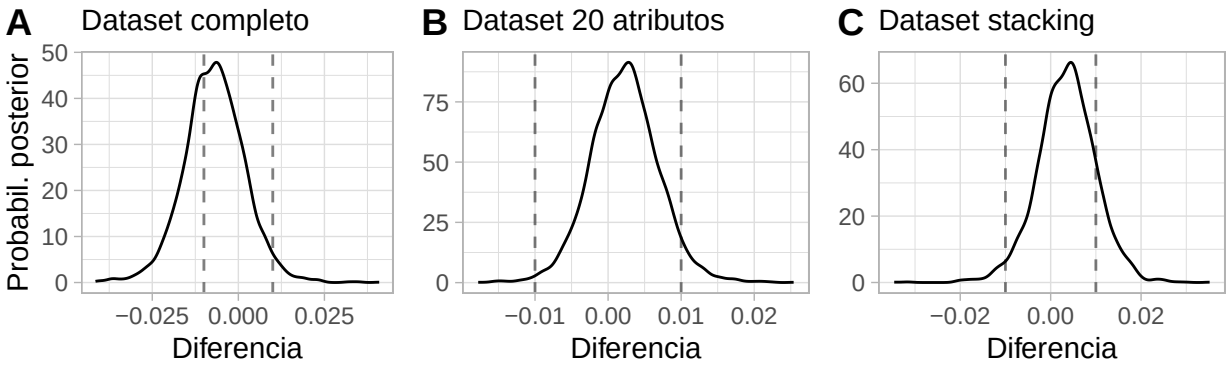

Figura 11: *Distribuciones de probabilidades posteriores de las diferencias en ROC-AUC entre modelos con y sin imputación de faltantes.*

4.5 Estrategias para combatir el desbalance de clases

Observando los diagramas de caja, parecería ser que el *stacking* de RPart mejora ligeramente el ROC-AUC tanto del dataset completo como de aquel reducido en atributos. Los constrastes de las probabilidades posteriores permiten establecer si esta mejora es real. A simple vista, no parecería ser posible afirmar lo mismo respecto al *upsampling* de la clase minoritaria (modelo 3) ni al *downsampling* de la clase mayoritaria (modelo 4).

En el caso del dataset con todos los atributos, solo el *stacking* tiene una probabilidad clara de aportar 1 punto porcentual extra a ROC-AUC (P = 0.99) respecto a no hacer este tipo de balanceo de clases. Sobre el dataset completo en atributos, el *downsampling* muestra el efecto contrario: tiene probabilidades de resultar en valores más bajos en las métricas (P = 0.80).

Tabla 5: *Contrastes de las distribuciones de probabilidades posteriores entre modelos con y sin balanceo de clases en el dataset completo.*

contrast	pract_pos	pract_equiv	pract_neg
1-XGB base_clean vs 1-XGB base_sub+stack	0.00	0.01	0.99
1-XGB base_clean vs 3-XGB up_clean	0.15	0.76	0.09
1-XGB base_clean vs 4-XGB down_clean	0.80	0.20	0.00

Para el dataset reducido en atributos, el *stacking* tiene una probabilidad de 0.95 de representar una mejora en ROC-AUC. En este caso, el *upsampling* y el *downsampling* tienen más probabilidad de entregar un resultado equivalente en ROC-AUC respecto a no balancear (P = 0.77).

Tabla 6: *Contrastes de las distribuciones de probabilidades posteriores entre modelos con y sin balanceo de clases en el dataset con 20 atributos.*

contrast	pract_pos	pract_equiv	pract_neg
1-XGB base_sub vs 1-XGB base_sub+stack	0.00	0.05	0.95
1-XGB base_sub vs 3-XGB up_sub	0.10	0.77	0.13
1-XGB base_sub vs 4-XGB down_sub	0.12	0.77	0.11

Al *stacking* no le representa ninguna mejora en ROC-AUC, la aplicación de *upsampling* o *downsampling*. El ensemble se considera una estrategia de balanceo de clases (Han, Pei, and Kamber 2011) así que posiblemente esto explique el resultado obtenido.

Tabla 7: *Contrastes de las distribuciones de probabilidades posteriores entre modelos con y sin balanceo de clases en el dataset con stacking.*

contrast	pract_pos	pract_equiv	pract_neg
1-XGB base_sub+stack vs 3-XGB up_sub+stack	0.25	0.64	0.11
1-XGB base_sub+stack vs 4-XGB down_sub+stack	0.23	0.63	0.14

4.6 Efecto de la acumulación de técnicas

La acumulación de técnicas, es decir, *upsampling* más imputación (modelo 5) o *downsampling* más imputación (modelo 6), no mejoran las métricas de los clasificadores.

Tabla 8: *Contrastes de las distribuciones de probabilidades posteriores entre modelos con y sin acumulación de técnicas.*

dataset	contrast	pract_pos	pract_equiv	pract_neg
Completo	1-XGB base vs 5-XGB up+imputación	0.04	0.69	0.26
Completo	1-XGB base vs 6-XGB down+imputación	0.22	0.72	0.06
Reducido	1-XGB base vs 5-XGB up+imputación	0.16	0.78	0.06
Reducido	1-XGB base vs 6-XGB down+imputación	0.38	0.60	0.02
Stacking	1-XGB base vs 5-XGB up+imputación	0.08	0.62	0.30
Stacking	1-XGB base vs 6-XGB down+imputación	0.86	0.14	0.00

4.7 Selección del modelo

La única técnica que resulta en mejoras objetivas en la estimación de ROC-AUC es el *stacking* de un ajuste de RPart con uno de XGBoost. Por ende, se elige ese enfoque para el ajuste del modelo final.

5 Modelo final

Los hiperparámetros optimizados del clasificador que consiste únicamente de un *stacking* son los siguientes:

```
## Boosted Tree Model Specification (classification)
##
## Main Arguments:
##   mtry = 17
##   trees = 1053
##   min_n = 2
##   tree_depth = 1
##   learn_rate = 0.0170270451495629
##   loss_reduction = 0.668829610197898
##   sample_size = 0.691625455598212
##
## Computational engine: xgboost
```

La curva ROC del modelo elegido muestra un comportamiento equilibrado y con una forma similar al modelo "base". En el cuadrante superior derecho del gráfico de la curva ROC (donde se encuentran los valores altos de sensibilidad que se buscan para nuestro modelo), hay una buena curvatura.

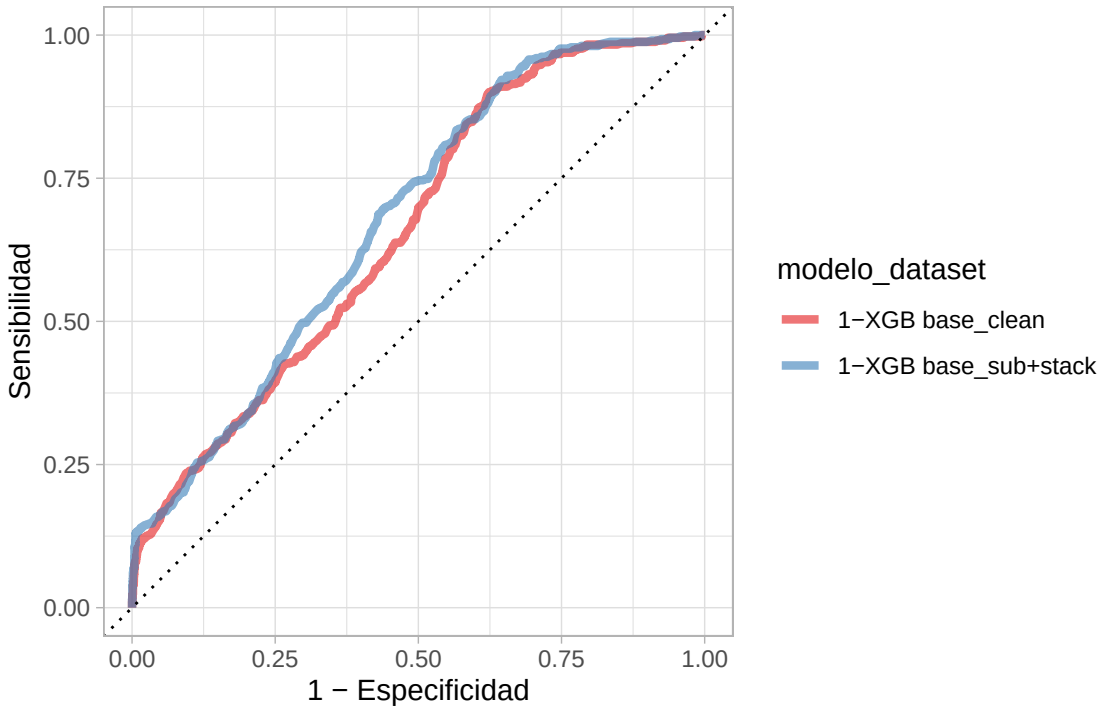

Figura 12: *Curva ROC del modelo base y del modelo elegido.*

5.1 Ajuste del umbral de clasificación

Se debe ajustar el punto de corte que determina la clasificación dura para las clases “Test COVID = Positivo” y “Test COVID = Negativo” que permitan alcanzar una Sensibilidad alta. Es deseable contar con la menor cantidad posible de falsos negativos sin perder demasiados verdaderos negativos. Para ello, se elige el nuevo umbral como aquel que maximiza el índice J.

Como se puede ver, las probabilidades predichas promedio a través de los 10 conjuntos de la validación cruzada del set de entrenamiento se encuentran en gran medida concentradas por debajo de 0.2. El punto de corte que maximiza el índice J es 0.093. Al aplicarlo al conjunto de entrenamiento resulta en una Sensibilidad estimada satisfactoria para el objetivo del presente trabajo.

La distribución de probabilidades en el conjunto de prueba es muy similar a la que resultó del set de entrenamiento, a partir de la cual se eligió el umbral de clasificación. Consiguientemente, se considera que el nuevo umbral es aplicable a la predicción de clases en datos nuevos.

En la siguiente sección se analizan en detalle los resultados de Sensibilidad y Especificidad en el conjunto de testeo como así también el resto de las métricas.

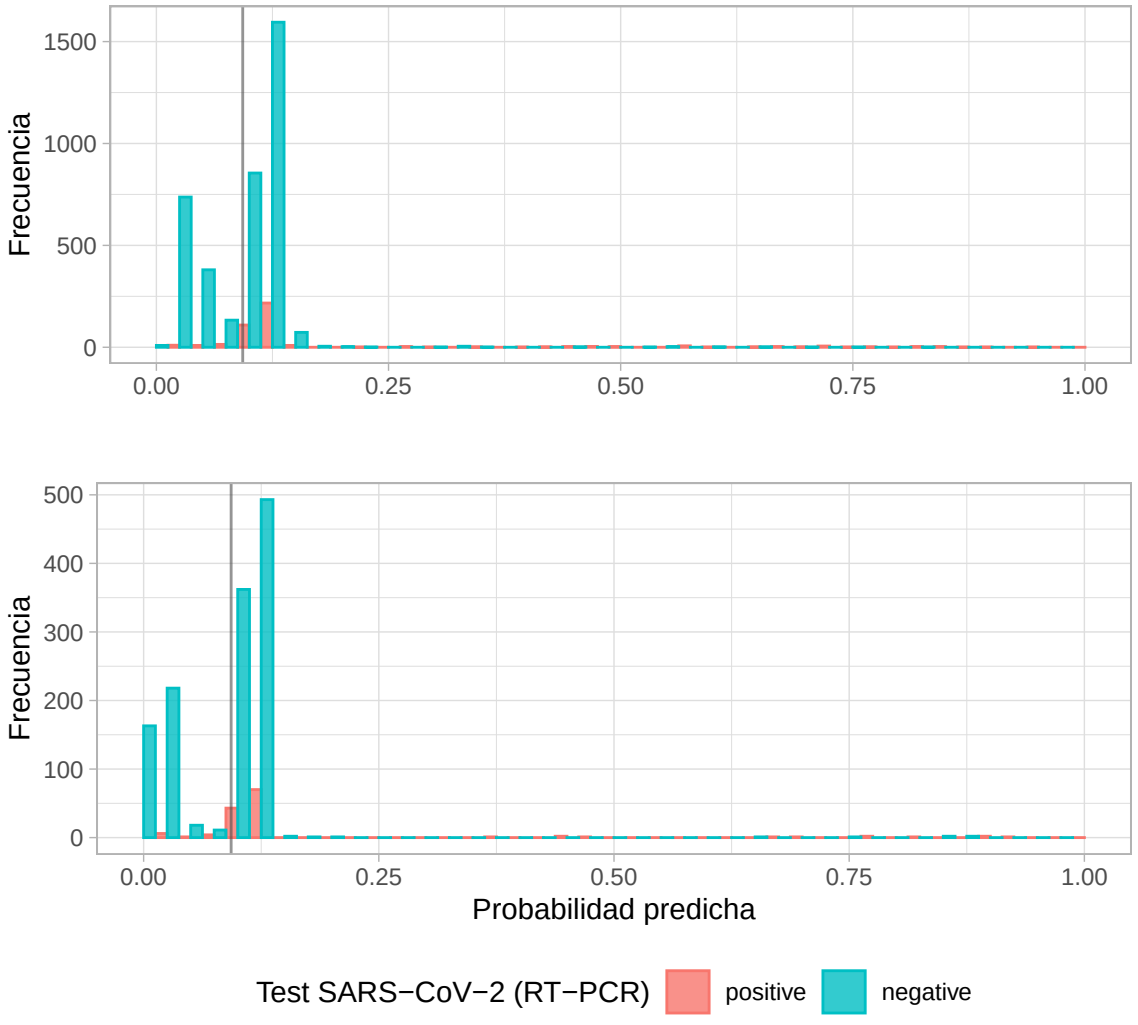

Figura 13: *Histograma de probabilidades en el set de entrenamiento (arriba) y testeo (abajo). La línea vertical corresponde al umbral de clasificación ajustado.*

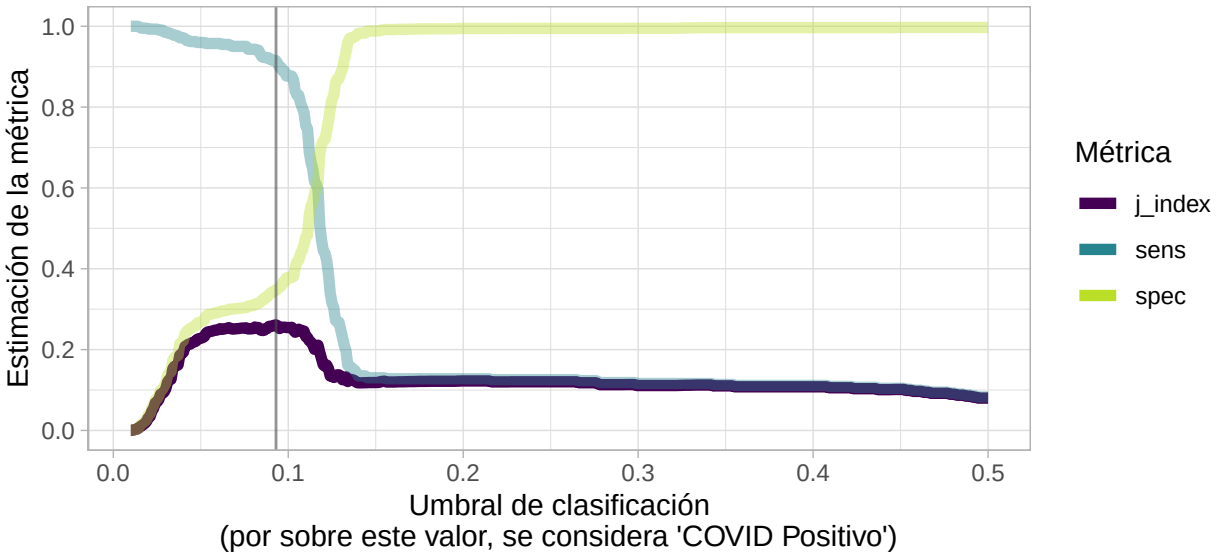

Figura 14: *Balanceo de la performance por variación del umbral. La línea vertical corresponde al máximo del índice J.*

Tabla 9: *Métricas luego de ajustar el umbral.*

Metrica	Estimacion
sens	0.915
spec	0.346
j_index	0.261

5.2 Evaluación

Para la evaluación final de modelo, se realiza un último ajuste en todo el conjunto de entrenamiento utilizando los parámetros obtenidos por validación cruzada de 10 iteraciones. Luego, se calcula su performance en datos nuevos del conjunto de prueba (1/4 del conjunto total) para comparar con las estimaciones en el conjunto de entrenamiento.

Tabla 10: *Comparación del ROC-AUC estimado por validación cruzada en el conjunto de entenamiento*

Métrica	Valor medio (conj. entrenamiento)	Desvío estándar (conj. entrenamiento)	Valor final (conj. prueba)
roc_auc	0.687	0.035	0.664

El valor de ROC-AUC en datos nuevos se se encuentra dentro del rango del valor medio estimado ± desvío estándar. Además, cae ligeramente por debajo de la “caja” estimada, es decir, dentro del rango estimado que forman Q1 y Q3.

La performance del modelo construido resulta considerablemente inferior a aquella alcanzada en Banerjee et al. (2020). Se cree que esto se debe, principalmente, a que este modelo tiene dificultad para clasificar a los pacientes cuyos valores del hemograma están ausentes (aproximadamente el 90%). En cambio en Banerjee et al. (2020) se hizo el ajuste solo sobre el relativamente pequeño conjunto de pacientes que presentan datos en el hemograma (n = 598).

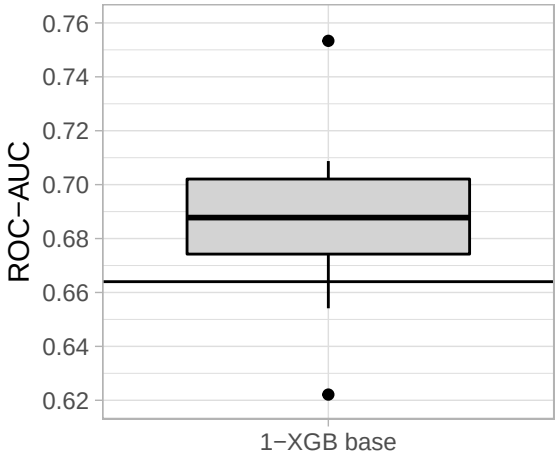

Figura 15: *Diagrama de caja de los valores de ROC-AUC en las 10 iteraciones de validación cruzada para el modelo con stacking. La línea representa la ROC-AUC en el conjunto de testeo.*

En relación a las métricas sobre predicciones de clase, en el conjunto de prueba el modelo:

- Identificó correctamente al 91.9% de los hisopados positivos. Se esperaba 91.5% (sensibilidad).
- Identificó correctamente al 40.2% de los hisopados negativos. Se esperaba 39.4% (especificidad).
- Individuos con una clasificación positiva tienen un 12.6% de probabilidades de tener un hisopado positivo. Se esperaba 13.4%. PPV
- Individuos con una clasificación negativa tienen un 97.4% de probabilidades de tener un hisopado negativo. Se esperaba 97.3%. NPV

Los resultados del modelo en el conjunto de prueba son muy similares a los estimados, validando las estimaciones de performance realizadas.

Tabla 11: *Comparación de las estimaciones (conj. entrenamiento) y valores finales (conj. prueba) de todas las métricas de interés.*

Métrica	Valor final (conj. prueba)	Estimación (conj. entrenamiento)
sens	0.919	0.915
spec	0.322	0.346
j_index	0.241	0.261
ppv	0.126	0.134
npv	0.974	0.973

5.3 Análisis cualitativo

¿Cómo se hubiera procedido en cada caso si se hubiera usado el modelo para decidir realizar o no el test de COVID-19 en pacientes nuevos? La matriz de confusión muestra los aciertos y desaciertos de cada clasificación.

Para los 989 pacientes clasificados como positivos por el modelo, se mantendría la sospecha que motivó la aplicación del modelo y se haría el test PCR. Esto no representa variación en los costos respecto a no aplicarlo.

Los 422 individuos clasificados como negativos por el modelo representan un ahorro **bruto** en el costo total de tests de COVID-19. Sin embargo, 411 de ellos serían realmente negativos y 11 serían, en realidad,

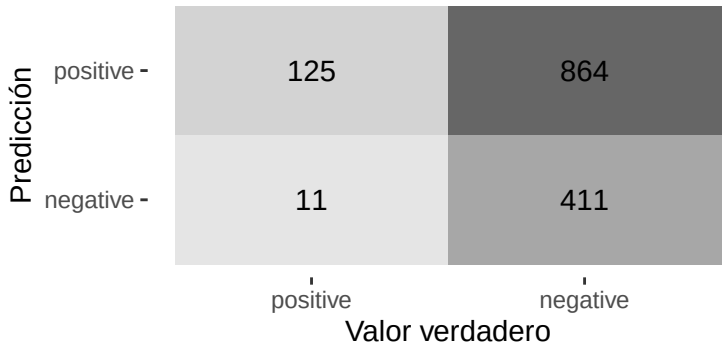

Figura 16: *Matriz de confusión en el conjunto de testeo.*

positivos. Entonces, al ahorro bruto de los 422 tests habría que restarle el costo del impacto de los falsos negativos. Esto es, el costo de necesitar tests adicionales para encontrar el diagnóstico o el costo de un paciente que podría irse a su casa y contagiar a más personas o el costo de que el cuadro se agrave por falta de diagnóstico temprano, entre otros. Por otra parte, existe un costo humano (sufrimiento, estrés, etc); más difícilmente cuantificable. Queda claro por qué un clasificador basado en la clase mayoritaria sería una mala idea.

Como la realización de un análisis de costos y riesgos excede ampliamente el alcance de este trabajo, se opta simplemente por no contabilizar a los falsos negativos como ahorro. En definitiva, se ahorraron 411 de los 1275 negativos que se podrían haber ahorrado. Esto es un 32.2% de ahorro, es decir, la Especificidad en el conjunto de prueba. Entonces, el modelo promete un ahorro aproximado del 34.6%, que es su Especificidad estimada en el conjunto de entrenamiento. Pero este porcentaje de ahorro es sólo sobre los negativos, no sobre el total.

Realmente, el ahorro sería de 29.1%. Esto es VN/Total. Acá entra en juego la prevalencia de casos positivos ya que los verdaderamente positivos nunca se van a poder evitar hacer (salvo que se tenga un modelo perfecto). El ahorro máximo teórico para este dataset es 90.1% que es la proporción de negativos.

Repitiendo el análisis anterior en el conjunto de prueba para así obtener una estimación “nominal” del modelo, resulta que **el ahorro estimado es del 30.9%**.

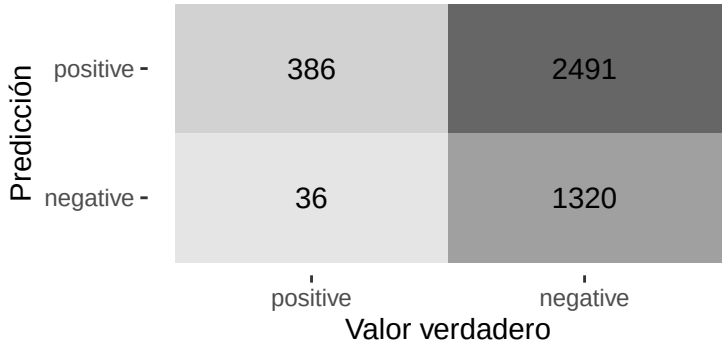

Figura 17: *Matriz de confusión en el conjunto de entrenamiento.*

5.4 Importancia de variables en el modelo final

En la importancia de variables se pone de manifiesto la contribución que tiene al modelo final el input de las probabilidades de un modelo anterior, resultando en la variable más importante del modelo. Ocupa esa

posición porque resume en una única variable mucha información sobre el problema.

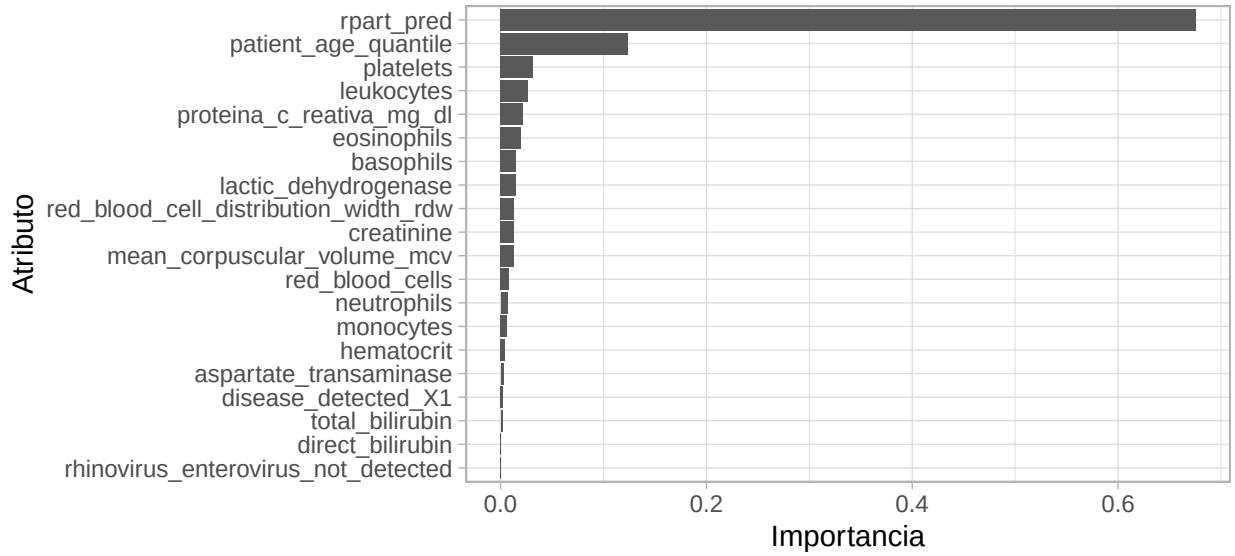

Figura 18: *Atributos que resultan más importantes en el ajuste final de XGBoost con hiperparámetros ajustados y stacking.*

6 Conclusiones

Fue posible la realización de un clasificador del resultado del test RT-PCR de SARS-CoV-2 a partir de análisis clínicos de laboratorio de pacientes sospechosos de COVID-19 pertenecientes al Hospital Israelita Albert Einstein (Brasil) utilizando XGBoost, una implementación de *gradient boosting*.

Ninguna de las técnicas aplicadas sobre las distintas variantes del dataset apuntadas específicamente a lidiar con datos faltantes o con el desbalance de clases otorgaron una mejora en la estimación del ROC-AUC del clasificador. Esto confirma la robustez de XGBoost a los datasets con tales características. Sí se obtuvo una mejora en el ROC-AUC al hacer un *stacking* de clasificadores, incluso realizado un subconjunto de 20 atributos del dataset original.

Principalmente el hemograma, pero también algunos parámetros hepáticos y clínicos, resultaron los más determinantes en la discriminación de presencia de SARS-CoV-2. Se destaca también el rol del atributo fabricado que indica presencia de alguno de los tantos virus que forman parte del diagnóstico diferencial de COVID-19.

Resultaría útil contar con información sobre la historia clínica de los pacientes para analizar si la presencia de enfermedades de base se encuentra relacionada o no con la alteración de ciertos parámetros de laboratorio y si aporta valor agregado al modelo.

Además, sería interesante evaluar si un ajuste de XGBoost usando solo los pacientes con hemograma completo entrega una mejor performance y si esta se acerca a la de los clasificadores de Banerjee et al. (2020) (redes neuronales, *random forest* y glmnet). También se podría analizar reducir el dataset a aquellos registros que tienen completo alguno de los parámetros que resultaron importantes en este trabajo (más allá del hemograma) y luego aplicarles una imputación más precisa que la media y moda, como lo es el *bagging*.

A futuro, se podría profundizar el análisis del impacto en los costos de una posible implementación en atención sanitaria. En ese caso, sería también importante caracterizar si este modelo tiene aplicabilidad para las distintas tipologías de pacientes (ambulatorios e internados) y si sirve tanto como detector temprano

como tardío. Aunque sobre este último punto se coincide con Banerjee et al. (2020) en que la mayor utilidad está en el *screening* temprano.

7 Referencias

Banerjee, Abhirup, Surajit Ray, Bart Vorselaars, Joanne Kitson, Michail Mamalakis, Simonne Weeks, Mark Baker, and Louise S Mackenzie. 2020. "Use of Machine Learning and Artificial Intelligence to Predict Sars-Cov-2 Infection from Full Blood Counts in a Population." *International Immunopharmacology* 86: 106705.

Blumenthal, David, Elizabeth J Fowler, Melinda Abrams, and Sara R Collins. 2020. "Covid-19—Implications for the Health Care System."

Chen, Tianqi, and Carlos Guestrin. 2016. "Xgboost: A Scalable Tree Boosting System." In *Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, 785–94.

Chen, Tianqi, Tong He, Michaël Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, and Ignacio Cano [aut] Kailong Chen [aut] Rory Mitchell [aut]. 2020. *Xgboost: Extreme Gradient Boosting*. https://CRAN.R-project.org/package=xgboost.

Chen, Tianqi, Tong He, Michaël Benesty, and Yuan Tang. 2020. "Understand Your Dataset with Xgboost." 2020. https://cran.r-project.org/web/packages/xgboost/vignettes/discoverYourData.html.

Cramer, Maria. 2020. "Covid-19 Testing Is in Short Supply. Should You Still Get a Test? - the New York Times." July 2020. https://www.nytimes.com/2020/07/31/health/coronavirus-covid-testing.html.

Data4u, Sao Paulo, E. Hospital Israelita Albert Einstein. 2020. "Diagnosis of Covid-19 and Its Clinical Spectrum." https://www.kaggle.com/einsteindata4u/covid19/version/7.

Deloitte. 2020. "Impact of the Covid-19 Pandemic on Healthcare Systems?" June 2020. https://www2.deloitte.com/fr/fr/pages/covid-insights/articles/impact-covid19-healthcare-systems.html.

Friedman, Jerome H. 2001. "Greedy Function Approximation: A Gradient Boosting Machine." *Annals of Statistics*, 1189–1232.

———. 2002. "Stochastic Gradient Boosting." *Computational Statistics & Data Analysis* 38 (4): 367–78.

García, Salvador, Julián Luengo, and Francisco Herrera. 2015. *Data Preprocessing in Data Mining*. Springer.

Han, Jiawei, Jian Pei, and Micheline Kamber. 2011. *Data Mining: Concepts and Techniques*. Elsevier.

Hong, Ki Ho, Sang Won Lee, Taek Soo Kim, Hee Jae Huh, Jaehyeon Lee, So Yeon Kim, Jae-Sun Park, et al. 2020. "Guidelines for Laboratory Diagnosis of Coronavirus Disease 2019 (Covid-19) in Korea." *Ann Lab Med* 40 (5): 351–60.

Kaufman, Shachar, Saharon Rosset, Claudia Perlich, and Ori Stitelman. 2012. "Leakage in Data Mining: Formulation, Detection, and Avoidance." *ACM Transactions on Knowledge Discovery from Data (TKDD)* 6 (4): 1–21.

Kuhn, Max. 2020. *Tidyposterior: Bayesian Analysis to Compare Models Using Resampling Statistics*. https://CRAN.R-project.org/package=tidyposterior.

Kuhn, Max, and Kjell Johnson. 2019. *Feature Engineering and Selection: A Practical Approach for Predictive Models*. CRC Press.

Kuhn, Max, and Davis Vaughan. 2020. *Probably: Tools for Post-Processing Class Probability Estimates*. https://CRAN.R-project.org/package=probably.

Kuhn, Max, and Hadley Wickham. 2020. *Tidymodels: Easily Install and Load the 'Tidymodels' Packages*. https://CRAN.R-project.org/package=tidymodels.

Lippi, Giuseppe, and Mario Plebani. 2020a. “A Modern and Pragmatic Definition of Laboratory Medicine.” *Clinical Chemistry and Laboratory Medicine (CCLM)* 1 (ahead-of-print).

———. 2020b. “Laboratory Abnormalities in Patients with Covid-2019 Infection.” *Clinical Chemistry and Laboratory Medicine (CCLM)* 58 (7): 1131–4.

MedlinePlus. n.d. “Pruebas Médicas.” https://medlineplus.gov/spanish/pruebas-de-laboratorio/.

Millan, Carolina, Andrea Navarro, and Stephan Kueffner. 2020. “Test Scarcity Makes Tracking Coronavirus in Latin America Harder - Bloomberg.” April 2020. https://www.bloomberg.com/news/articles/2020-04-08/test-scarcity-makes-tracking-virus-in-latin-america-a-crapshoot.

OMS, Organización Mundial de la Salud. 2020a. “Preguntas Sobre Los Nuevos Coronavirus.” April 2020. https://www.who.int/es/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/q-a-coronaviruses.

———. 2020b. “Cronología de La Respuesta de La Oms a La Covid-19.” June 2020. https://www.who.int/es/news-room/detail/29-06-2020-covidtimeline.

Pfeiffer, Sacha, Meg Anderson, and Barbara Van Woerkom. 2020. “Why Is There a Coronavirus Test Shortage? One Reason: We Don’t Have Enough Swabs : NPR.” NPR, USA. May 2020. https://www.npr.org/2020/05/12/853930147/despite-early-warnings-u-s-took-months-to-expand-swab-production-for-covid-19-te.

Therneau, Terry, Beth Atkinson, and Brian Ripley. 2019. *Rpart: Recursive Partitioning and Regression Trees*. https://CRAN.R-project.org/package=rpart.

Wickham, Hadley. 2019. *Tidyverse: Easily Install and Load the ’Tidyverse’*. https://CRAN.R-project.org/package=tidyverse.

Wolpert, David H. 1992. “Stacked Generalization.” *Neural Networks* 5 (2): 241–59.

Youden, William J. 1950. “Index for Rating Diagnostic Tests.” *Cancer* 3 (1): 32–35.

8 Apéndice

8.1 Columnas con todos los datos vacíos

x
mycoplasma_pneumoniae
urine_sugar
partial_thromboplastin_time_ptt
prothrombin_time_pt_activity
d_dimer

8.2 Hiperparámetros del modelo para hacer reducción de atributos

```
## Boosted Tree Model Specification (classification)
##
## Main Arguments:
##   mtry = 46
##   trees = 441
##   min_n = 10
##   tree_depth = 3
##   learn_rate = 0.0177239233007729
```

```
##   loss_reduction = 0.987814261220396
##   sample_size = 0.998033098296444
##
## Computational engine: xgboost
```

8.3 Estadística descriptiva de las variables numéricas

	Mean	Std.Dev	Skewness	Kurtosis
alanine_transaminase	0.00	1.00	4.90	29.31
albumin	0.00	1.04	-0.23	-0.01
alkaline_phosphatase	0.00	1.00	1.95	3.08
arteiral_fio2	0.00	1.03	1.05	1.49
arterial_lactic_acid	0.00	1.02	1.47	1.34
aspartate_transaminase	0.00	1.00	4.81	27.81
base_excess_arterial_blood_gas_analysis	0.00	1.02	-1.18	1.56
base_excess_venous_blood_gas_analysis	0.00	1.00	-0.43	1.85
basophils	0.00	1.00	2.91	24.74
creatine_phosphokinase_cpk	0.00	1.00	5.57	33.73
creatinine	0.00	1.00	0.89	2.93
cto2_arterial_blood_gas_analysis	0.00	1.02	-0.79	0.80
direct_bilirubin	0.00	1.00	3.81	20.04
eosinophils	0.00	1.00	2.75	12.97
ferritin	0.00	1.02	2.52	6.26
gamma_glutamyltransferase	0.00	1.00	5.83	40.85
hb_saturation_arterial_blood_gases	0.00	1.02	-0.42	-1.28
hb_saturation_venous_blood_gas_analysis	0.00	1.00	-0.17	-1.03
hco3_arterial_blood_gas_analysis	0.00	1.02	-0.56	1.00
hco3_venous_blood_gas_analysis	0.00	1.00	-0.20	0.79
hematocrit	0.00	1.00	-0.73	1.41
hemoglobin	0.00	1.00	-0.62	1.15
indirect_bilirubin	0.00	1.00	2.83	12.04
international_normalized_ratio_inr	0.00	1.00	3.47	20.75
ionized_calcium	0.00	1.01	0.57	1.32
lactic_dehydrogenase	0.00	1.00	1.10	0.55
leukocytes	0.00	1.00	1.42	2.90
lipase_dosage	0.00	1.07	0.73	-1.29
lymphocytes	0.00	1.00	0.47	0.14
magnesium	0.00	1.01	-0.04	-0.31
mean_corpuscular_hemoglobin_concentration_mchc	0.00	1.00	-0.52	2.27
mean_corpuscular_hemoglobin_mch	0.00	1.00	-1.12	5.03
mean_corpuscular_volume_mcv	0.00	1.00	-0.68	2.96
mean_platelet_volume	0.00	1.00	0.42	-0.03
metamyelocytes	0.00	1.01	3.68	15.09
monocytes	0.00	1.00	0.95	1.94
myelocytes	0.00	1.01	4.70	22.95
neutrophils	0.00	1.00	-0.02	-0.19
patient_age_quantile	9.32	5.78	0.03	-1.21
pco2_arterial_blood_gas_analysis	0.00	1.02	2.14	4.21
pco2_venous_blood_gas_analysis	0.00	1.00	1.05	6.50
ph_arterial_blood_gas_analysis	0.00	1.02	-2.33	5.09
ph_venous_blood_gas_analysis	0.00	1.00	-0.68	4.08
phosphor	0.00	1.03	1.15	1.20

	Mean	Std.Dev	Skewness	Kurtosis
platelets	0.00	1.00	1.79	13.32
po2_arterial_blood_gas_analysis	0.00	1.02	0.87	-0.40
po2_venous_blood_gas_analysis	0.00	1.00	1.18	1.54
potassium	0.00	1.00	0.40	-0.04
promyelocytes	0.00	1.01	9.55	90.06
proteina_c_reativa_mg_dl	0.00	1.00	3.63	17.21
red_blood_cell_distribution_width_rdw	0.00	1.00	2.36	9.06
red_blood_cells	0.00	1.00	-0.37	1.25
relationship_patient_normal	0.00	1.01	1.37	4.56
rods	0.00	1.01	1.68	2.03
segmented	0.00	1.01	-0.48	-0.87
serum_glucose	0.00	1.00	3.78	19.69
sodium	0.00	1.00	-0.35	3.25
total_bilirubin	0.00	1.00	2.31	6.91
total_co2_arterial_blood_gas_analysis	0.00	1.02	-0.48	0.85
total_co2_venous_blood_gas_analysis	0.00	1.00	-0.16	0.83
urea	0.00	1.00	4.34	40.39
urine_density	0.00	1.01	0.25	-0.37
urine_ph	5.95	0.87	0.30	-1.17
urine_red_blood_cells	0.00	1.01	6.85	49.26
vitamin_b12	0.00	1.22	-0.35	-2.33

8.4 Hiperparámetros del modelo para hacer el *stacking*

```
## Decision Tree Model Specification (classification)
##
## Main Arguments:
##   cost_complexity = 2.95236962938079e-10
##   tree_depth = 7
##   min_n = 12
##
## Computational engine: rpart
```

8.5 Métricas de los experimentos

modelo_dataset	roc_auc
1-XGB base_clean	0.669
1-XGB base_sub	0.675
1-XGB base_sub+stack	0.687
2-XGB imputación_clean	0.676
2-XGB imputación_sub	0.673
2-XGB imputación_sub+stack	0.683
3-XGB up_clean	0.668
3-XGB up_sub	0.676
3-XGB up_sub+stack	0.684
4-XGB down_clean	0.652
4-XGB down_sub	0.675
4-XGB down_sub+stack	0.685
5-XGB up+imputación_clean	0.674

modelo_dataset	roc_auc
5-XGB up+imputación_sub	0.673
5-XGB up+imputación_sub+stack	0.691
6-XGB down+imputación_clean	0.665
6-XGB down+imputación_sub	0.667
6-XGB down+imputación_sub+stack	0.665