

Mentoría: Coronavirus en Argentina y el mundo

Práctico 4: Aprendizaje Supervisado.

Matías Bettera Marcat

Lucas C. Didoné

Daniel Peralta

1. Práctico: Aprendizaje Supervisado

Objetivo:

Profundizar el trabajo realizado en el práctico anterior. Intentaremos mejorar los resultados iterando sobre la ingeniería de atributos, el modelado, y el análisis de la salida de los modelos.

2. Introducción

Se decide trabajar con el conjunto de datos propuesto por el Ministerio de Salud de Argentina (disponible en: <http://datos.salud.gob.ar/dataset/covid-19-casos-registrados-en-la-republica-argentina/archivo/fd657d02-a33a-498b-a91b-2ef1a68b8d16>), debido a que cuenta con variables que resultan de interés a la hora de seleccionar un target para intentar predecir empleando modelos de aprendizaje supervisado.

Además, este *dataset* es actualizado diariamente a las 20:00 hs. notificando CASOS COVID-19 registrados en el país con un corte del día a las 17:45 hs.

Las variables con las que cuenta el conjunto de datos son:

Título de la columna	Tipo de dato	Descripción
id_evento_caso	Número entero (integer)	Numero de caso
sexo	Texto (string)	Sexo
edad	Número entero (integer)	Edad
edad_años_meses	Texto (string)	Edad indicada en meses o años
residencia_pais_nombre	Texto (string)	País de residencia
residencia_provincia_nombre	Texto (string)	Provincia de residencia
residencia_departamento_nombre	Texto (string)	Departamento de residencia
carga_provincia_nombre	Texto (string)	Provincia de establecimiento de carga
fecha_inicio_sintomas	Fecha ISO-8601 (date)	Fecha de inicio de síntomas
fecha_apertura	Fecha ISO-8601 (date)	Fecha de apertura del caso
sepi_apertura	Número entero (integer)	Semana Epidemiológica de fecha de apertura
fecha_internacion	Fecha ISO-8601 (date)	Fecha de internación
cuidado_intensivo	Texto (string)	Indicación si estuvo en cuidado intensivo
fecha_cui_intensivo	Fecha ISO-8601 (date)	Fecha de ingreso a cuidado intensivo en el caso de corresponder
fallecido	Texto (string)	Indicación de fallecido
fecha_fallecimiento	Tiempo ISO-8601 (time)	Fecha de fallecimiento en el caso de corresponder
asistencia_respiratoria_mecanica	Texto (string)	Indicación si requirió asistencia respiratoria mecánica
carga_provincia_id	Número entero (integer)	Código de Provincia de carga
origen_financiamiento	Texto (string)	Origen de financiamiento
clasificacion	Texto (string)	Clasificación manual del registro
clasificacion_resumen	Texto (string)	Clasificación del caso
residencia_provincia_id	Número entero (integer)	Código de Provincia de residencia
fecha_diagnostico	Tiempo ISO-8601 (time)	Fecha de diagnóstico
residencia_departamento_id	Número entero (integer)	Código de Departamento de residencia
ultima_actualizacion	Fecha ISO-8601 (date)	Última actualización

Resulta de gran interés intentar poder predecir si un caso clasificado como portador de COVID-19 sobrevivirá o no a la enfermedad. A su vez, trataremos de predecir, dado un nuevo caso si este posee o no la enfermedad.

3. Limpieza, Pre-Procesado y Feature Engineer

Previo a la realización de esta etapa, se cargó el *dataset* utilizando Pandas y se analizaron los valores nulos y tipos de datos de cada una de las variables.

A continuación, se mencionarán el trabajo más importante realizado sobre cada una de las variables del conjunto de datos:

Grupo de variables	Variable	Trabajo realizado	Resultado
Sexo	sexo	<i>One Hot Encoder</i> y <i>renombré de columnas resultantes</i> .	Se mantiene la variable.
Edad	edad	Se pasan todas las edades a años, colocando un 0 en aquellos pacientes con edades expresadas en meses.	Se mantiene la variable edad, donde cada registro es un número entero.
	edad_año_meses	-	Se elimina la variable.
Lugares	residencia_pais_nombre	Se analizó la cantidad de casos por país y se detectó que el 95,89% de los registros corresponden a Argentina.	Se eliminaron los registros de personas no residentes en Argentina, posteriormente se eliminó la variable debido a que todos los registros pertenecen al mismo país.
	carga_provincia_nombre	Se contabilizó la cantidad de registros donde la provincia de residencia es igual a la de carga, se detectó que esto ocurre en el 82.84% de los casos. Se agregó una variable booleana llamada pcia_rec_eq_pcia_car que indica con un si la provincia de residencia es igual a la provincia de carga del dato.	Se agregó una nueva variable llamada pcia_rec_eq_pcia_car . Se eliminó la variable carga_provincia_nombre debido a que realizar un <i>One Hot Encoder</i> por cada provincia de carga y luego por cada provincia de residencia consideramos que implica agregar demasiada dimensionalidad a esa información.
	residencia_provincia_nombre	<i>One Hot Encoder</i> y <i>renombré de columnas resultantes</i> .	Se mantiene la variable.
	residencia_departamento_nombre	Se decide descartar ya que decidimos trabajar con la provincia como máximo nivel de granularidad a nivel geográfico.	Se elimina la variable.
Fechas	fecha_inicio_sintomas fecha_apertura fecha_internacion fecha_cui_intensivo fecha_diagnostico	<ul style="list-style-type: none"> Análisis de valores nulos por cada variable. Análisis de fechas iguales: se comparan todas las fechas entre sí para contabilizar la cantidad de registros por cada par de 	Se decide conservar la variable: fecha_di_dif_fecha_ap que representa la diferencia entre fecha_diagnostico y fecha_apertura ya que es la que menos tiene menos

		<p>variables que tiene el mismo valor.</p> <ul style="list-style-type: none"> • Creación de columnas: Se seleccionan fechas de a pares y se calcula la cantidad de días de diferencia entre esas dos fechas, estos valores se almacenan en una nueva columna. • Análisis de nulos en las nuevas columnas. 	<p>registros nulos de todas las nuevas variables creadas. Se eliminan las variables: fecha_inicio_sintomas, fecha_apertura, fecha_internacion, fecha_cui_intensivo y fecha_diagnostico debido a que se pretende eliminar la información temporal del <i>dataset</i> por la complejidad que implica el análisis de series temporales pero a su vez se desea mantener la información representativa de las fechas, es por este motivo que se decide conservar esta nueva variable que contiene un valor entero que representa la diferencia entre dos fechas importantes.</p> <p>Además, se eliminan todos los registros que tienen valores nulos en la variable: fecha_di_fecha_ap</p>
Semana Epidemiológica	sepi_apertura	Se analiza la distribución de los datos de esta variable.	Se conserva la variable.
Cuidado Intensivo	cuidado_intensivo	Binarización de la variable.	Se conserva la variable.
Asistencia Respiratoria Mecánica	asistencia_respiratoria_mecanica	Binarización de la variable.	Se conserva la variable.
Financiamiento	origen_financiamiento	Binarización y renombrado de la variable.	La variable resultante se llama: financiamiento_publico e indica con un 1 si el origen del financiamiento es público, de lo contrario la variable tendrá un 0.
Clasificación	clasificacion_resumen	Se conservan únicamente los registros que contengan valores confirmado o descartado en esta variable. Luego, se procede a la binarización y renombre de la misma.	La variable pasó a nombrarse covid_19_confirmado y tendrá un 1 en caso de ser positivo y un 0 si no lo es.
	clasificacion	Se analiza la distribución de los datos de esta variable	Se elimina la variable debido a que consideramos que con la información contenida en la nueva variable covid_19_confirmado contiene la información más densa y relevante de la clasificación.

Fallecido	fallecido	Binarización de la variable	Se conserva la variable.
VARIABLES ELIMINADAS			
VARIABLE		MOTIVO	
id_evento_caso		No resulta de utilidad para el análisis.	
carga_provincia_id		No resulta de utilidad para el análisis.	
residencia_provincia_id		No resulta de utilidad para el análisis.	
residencia_departamento_id		No resulta de utilidad para el análisis.	

Luego del proceso de Limpieza, Pre-Procesado y Feature Engineer mencionado, se procede a convertir cada variable en su tipo de dato correspondiente utilizando Pandas y posteriormente se eliminan los valores nulos que pudieron haber quedado.

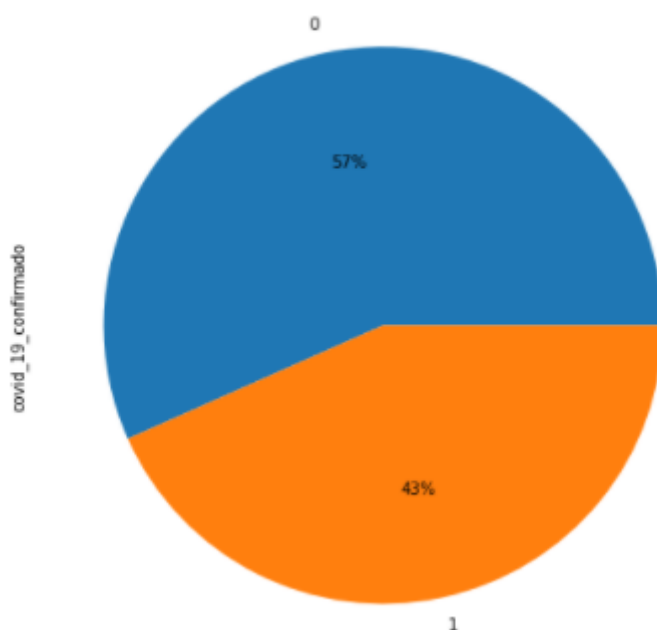
Finalmente se almacena el *dataset* resultante de aplicar todo el proceso mencionado.

4. Experimentación y evaluación

4.1. Experimento 1: COVID 19 – Confirmado

En este experimento se intentó predecir la variable covid_19_confirmado.

Análisis de balance de la variable target:



En la figura, se puede observar que la variable no presenta un desbalance de clases evidentes. Se puede observar que la clase mayoritaria es la 0, es decir aquellos casos que no presenten COVID – 19.

Debido a que la variable de salida no presenta un claro desbalance, se escoge la métrica *Accuracy* para evaluar el rendimiento del modelo por su interpretabilidad.

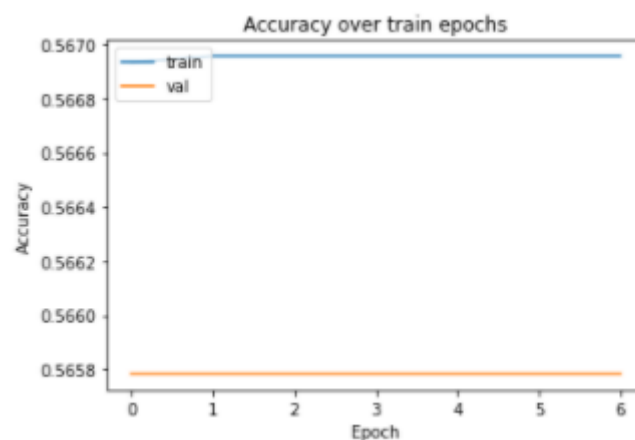
Modelo utilizado:

Como modelo predictor en este experimento se escogió una Red Neuronal (perceptrón multicapa) con la siguiente arquitectura:

Model: "sequential"		
Layer (type)	Output Shape	Param #
dense (Dense)	(None, 2048)	71680
activation (Activation)	(None, 2048)	0
dense_1 (Dense)	(None, 1024)	2098176
activation_1 (Activation)	(None, 1024)	0
dropout (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 1024)	1049600
activation_2 (Activation)	(None, 1024)	0
dropout_1 (Dropout)	(None, 1024)	0
dense_3 (Dense)	(None, 1024)	1049600
activation_3 (Activation)	(None, 1024)	0
dropout_2 (Dropout)	(None, 1024)	0
dense_4 (Dense)	(None, 512)	524800
activation_4 (Activation)	(None, 512)	0
dropout_3 (Dropout)	(None, 512)	0
dense_5 (Dense)	(None, 512)	262656
activation_5 (Activation)	(None, 512)	0
dropout_4 (Dropout)	(None, 512)	0
dense_6 (Dense)	(None, 256)	131328
activation_6 (Activation)	(None, 256)	0
dense_7 (Dense)	(None, 256)	65792
activation_7 (Activation)	(None, 256)	0
dense_8 (Dense)	(None, 128)	32896
activation_8 (Activation)	(None, 128)	0
dense_9 (Dense)	(None, 128)	16512
activation_9 (Activation)	(None, 128)	0
dense_10 (Dense)	(None, 2)	258
activation_10 (Activation)	(None, 2)	0
Total params: 5,303,298		
Trainable params: 5,303,298		
Non-trainable params: 0		

Resultados obtenidos:

El modelo finalizó su entrenamiento automáticamente luego de 7 épocas, debido a que no se detectó una mejora en la pérdida (*loss*) sobre el conjunto de validación durante 5 épocas sucesivas.



Se puede observar que durante el proceso de entrenamiento no se evidenció una mejora en la métrica Accuracy en ninguno de los dos conjuntos de datos.

Los máximos resultados en la métrica obtenidos fueron los siguientes:

- Accuracy train: 0.5669585
- Accuracy validation: 0.565784
- Accuracy test: 0.5562

Luego, se realizaron 10 predicciones sobre el conjunto de test, de las cuales 7 fueron exitosas y 3 erróneas.

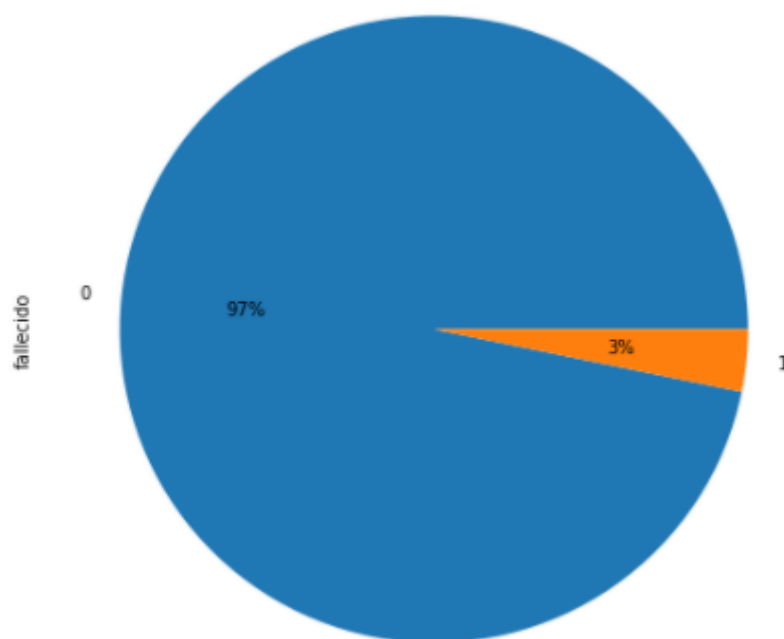
Conclusión:

Los resultados obtenidos luego de evaluar el modelo no resultan satisfactorios, sin embargo, esperábamos resultados que demuestren esa hipótesis ya que consideramos que no contamos con los datos suficientes para poder realizar esta clasificación correctamente.

4.2. Experimento 4.2: Sobrevive COVID-19: Neural Network Sin Bias

En este experimento se intentó predecir la variable fallecido (sobre aquellos datos que pertenezcan a casos clasificados positivos en la variable covid_19_confirmado).

Análisis de balance de la variable target:



En la figura, se puede observar que la variable presenta un desbalance de clases evidentes. Se puede observar que la clase mayoritaria es la 0, es decir aquellos casos que sobreviven.

Debido a que la variable de salida presenta un claro desbalance, se escoge la métrica *f1_score*, que representa la media armónica entre *Precision* y *Recall*.

Modelo utilizado:

Como modelo predictor en este experimento se escogió una Red Neuronal (perceptrón multicapa) con la siguiente arquitectura:

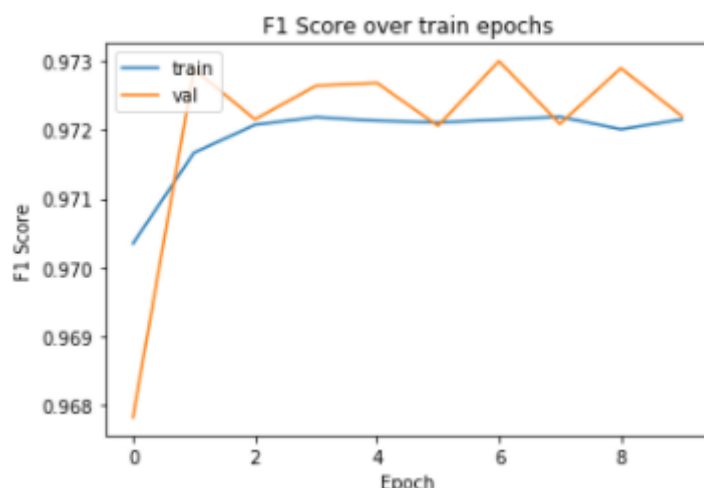
Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 2048)	71680
activation (Activation)	(None, 2048)	0
dense_1 (Dense)	(None, 1024)	2098176
activation_1 (Activation)	(None, 1024)	0
dropout (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 1024)	1049600
activation_2 (Activation)	(None, 1024)	0
dropout_1 (Dropout)	(None, 1024)	0
dense_3 (Dense)	(None, 1024)	1049600
activation_3 (Activation)	(None, 1024)	0
dropout_2 (Dropout)	(None, 1024)	0
dense_4 (Dense)	(None, 512)	524800
activation_4 (Activation)	(None, 512)	0
dropout_3 (Dropout)	(None, 512)	0
dense_5 (Dense)	(None, 512)	262656
activation_5 (Activation)	(None, 512)	0
dropout_4 (Dropout)	(None, 512)	0
dense_6 (Dense)	(None, 256)	131328
activation_6 (Activation)	(None, 256)	0
dense_7 (Dense)	(None, 256)	65792
activation_7 (Activation)	(None, 256)	0
dense_8 (Dense)	(None, 128)	32896
activation_8 (Activation)	(None, 128)	0
dense_9 (Dense)	(None, 128)	16512
activation_9 (Activation)	(None, 128)	0
dense_10 (Dense)	(None, 2)	258
activation_10 (Activation)	(None, 2)	0
Total params: 5,303,298		
Trainable params: 5,303,298		
Non-trainable params: 0		

Es importante destacar que en este modelo no se introdujo ningún sesgo (*bias*) que modele el desbalance existente entre las clases a la hora de inicializar los parámetros de la Red Neuronal.

Resultados obtenidos:

El modelo finalizó su entrenamiento automáticamente luego de 10 épocas, debido a que no se detectó una mejora en la pérdida (*loss*) sobre el conjunto de validación durante 5 épocas sucesivas.



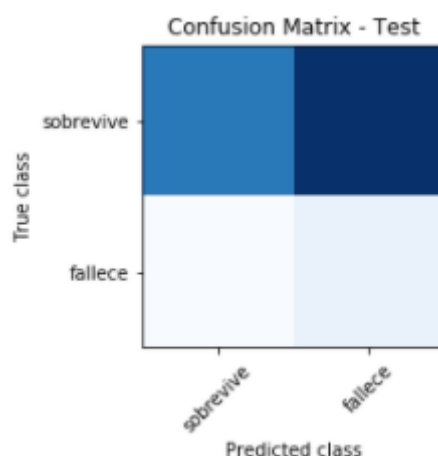
En el gráfico se puede observar la progresión del modelo, que, a pesar de no lograr estabilizarse completamente sobre el conjunto de datos de validación, alcanzó valores significativos de la métrica F1 Score.

Los máximos resultados en la métrica obtenidos fueron los siguientes:

- F1 Score train: 0.9721856
- F1 Score validation: 0.9729925
- F1 Score test: 0.9680

Luego, se realizaron 1000 predicciones sobre el conjunto de test, de las cuales el 44.1% fueron correctas. El porcentaje de ejemplos de la clase minoritaria sobre este conjunto de 1000 datos de test fue de 4.3%.

Al analizar la Matriz de Confusión de las predicciones sobre el conjunto de test, observamos lo siguiente:



Conclusión:

Si bien los valores obtenidos en la métrica son altos, lo que parece indicar que el modelo es capaz de generalizar correctamente, al analizar más en detalle las predicciones sobre el conjunto de test, podemos notar que el modelo suele equivocarse en las predicciones de casos en los que el paciente sobrevive, prediciendo que no es así (Falsos positivos).

Además, es posible notar que no suele acertar en los casos en los que la clase de salida esperada es “fallece”.

Este fenómeno podría explicar los altos resultados obtenidos en la métrica, puesto que el 97 % de los datos corresponden a la clase “sobrevive”, y según se observa en la matriz de confusión, es el caso en que mejor funciona el modelo (es decir el modelo predice correctamente la clase mayoritaria).

4.3. Experimento 4.3: Sobrevive COVID-19: Neural Network Con Bias

En este experimento, al igual que en el anterior (4.2), se intentó predecir la variable fallecido. Sin embargo, esta vez, se introdujo un sesgo (*bias*) que intentó modelar el desbalance existente entre las clases al momento de inicializar los parámetros en la capa de salida de la Red Neuronal.

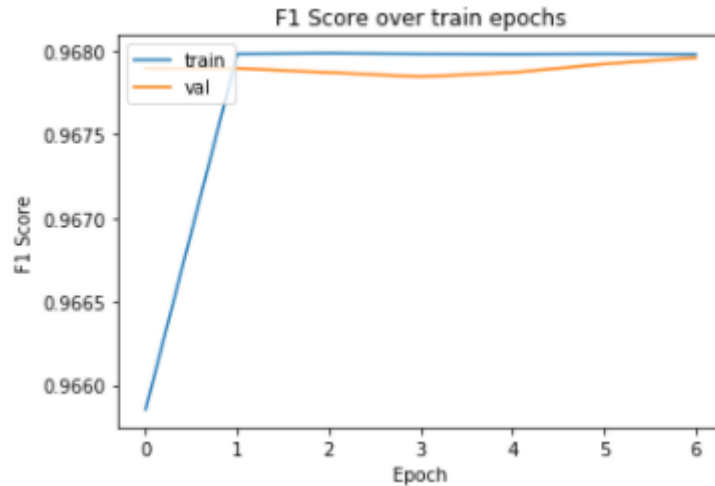
Modelo utilizado:

Como modelo predictor en este experimento se escogió una Red Neuronal (perceptrón multicapa) con la siguiente arquitectura:

Model: "sequential"		
Layer (type)	Output Shape	Param #
=====		
dense (Dense)	(None, 2048)	71680
dense_1 (Dense)	(None, 1024)	2098176
dropout (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 1024)	1049600
dropout_1 (Dropout)	(None, 1024)	0
dense_3 (Dense)	(None, 1024)	1049600
dropout_2 (Dropout)	(None, 1024)	0
dropout_3 (Dropout)	(None, 1024)	0
dense_4 (Dense)	(None, 512)	524800
dropout_4 (Dropout)	(None, 512)	0
dense_5 (Dense)	(None, 512)	262656
dropout_5 (Dropout)	(None, 512)	0
dense_6 (Dense)	(None, 256)	131328
dropout_6 (Dropout)	(None, 256)	0
dense_7 (Dense)	(None, 256)	65792
dropout_7 (Dropout)	(None, 256)	0
dense_8 (Dense)	(None, 128)	32896
dropout_8 (Dropout)	(None, 128)	0
dense_9 (Dense)	(None, 128)	16512
dropout_9 (Dropout)	(None, 128)	0
dense_10 (Dense)	(None, 2)	258
=====		
Total params: 5,303,298		
Trainable params: 5,303,298		
Non-trainable params: 0		

Resultados obtenidos:

El modelo finalizó su entrenamiento automáticamente luego de 7 épocas, debido a que no se detectó una mejora en la pérdida (*loss*) sobre el conjunto de validación durante 5 épocas sucesivas.



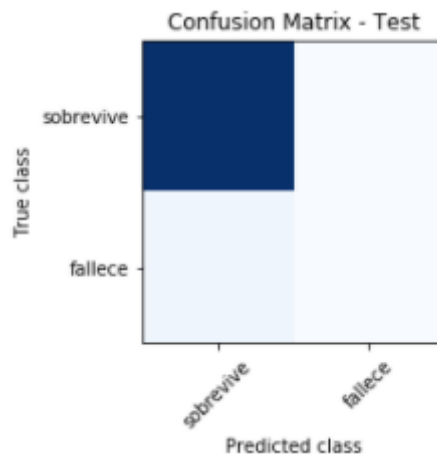
Al observar la gráfica, es posible notar que el modelo se encuentra más estable que en el experimento donde no se introdujo el sesgo, además la convergencia parece encontrarse más rápidamente (luego de un menor número de épocas de entrenamiento).

Los máximos resultados en la métrica obtenidos fueron los siguientes:

- F1 Score train: 0.96798587
- F1 Score validation: 0.96795857
- F1 Score test: 0.9679

Luego, se realizaron 1000 predicciones sobre el conjunto de test, de las cuales el 95.7% fueron correctas. El porcentaje de ejemplos de la clase minoritaria sobre este conjunto de 1000 datos de test fue de 4.3%.

Al analizar la Matriz de Confusión de las predicciones sobre el conjunto de test, observamos lo siguiente:



Conclusión:

En este experimento, al igual que en el anterior (4.2), se evidencian valores elevados en la métrica, sin embargo, al analizar la matriz de confusión, en este caso notamos una diferencia con respecto al experimento en el que no se introdujo el sesgo: desaparecen los casos de Falsos Positivos.

Sin embargo, el modelo sigue presentando el problema de predecir correctamente la clase mayoritaria pero no la minoritaria, lo que quizás explica los altos valores obtenidos en la métrica.

4.4. Experimento 4.4: Sobrevive COVID-19: RandomForest

En este experimento, al igual que en los anteriores (4.2 y 4.3), se intentó predecir la variable fallecido. Sin embargo, en este experimento se evaluó con las métricas: *Accuracy*, *Precision*, *Recall* y *F1-Score*.

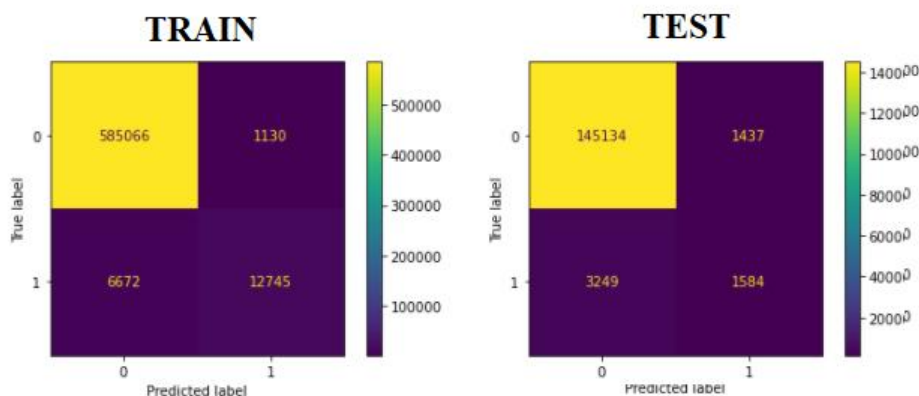
Modelo utilizado:

Como predictor en este experimento se escogió el modelo RandomForest.

Resultados obtenidos:

	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Train	0.98	0.9864750905682458	0.9871171853972751	0.9860752132345677
Test	0.96	0.9636189336173616	0.9690496948561464	0.9655744841765938

Matriz de Confusión:



Conclusión:

Al igual que en los experimentos anteriores, las métricas expresan valores altos, sin embargo, al analizar la Matriz de confusión es posible notar que en el conjunto de Train se producen Falsos Negativos, es decir predice 0 (sobrevive) cuando debería ser 1 (fallece). En el conjunto de test se evidencia el mismo problema.

Sin embargo, el número casos predichos de la clase de 1 (fallece), que pertenecen a esa clase, aumentó con respecto a los experimentos en los que se utilizaron Redes Neuronales.