UNIVERSIDAD DE ANTIOQUIA

FACULTAD DE INGENIERÍA
DEPARTAMENTO DE BIOINGENIERÍA
Inteligencia Artificial

Proyecto
Informe final

Nombres
Nataly Rodriguez Ateourtua
Juan Camilo Cataño Zuleta
Daniel Enrique López Yepes

Docente
Raul Ramos Pollan

Documentos
1.001.505.918
1.001.738.289
1.000.407.016

Medellín - Antioquia
2022

1. **Introducción.**

Problema predictivo a resolver:

La Organización Mundial de la Salud (OMS) debido al aumento exponencial del número de casos, caracterizó el 11 de marzo al COVID-19, causado por el SARS-CoV-2, como pandemia, siendo una amenaza en los sistemas de salud de todo el mundo, por este motivo desde el comienzo del brote, varios investigadores han estado recopilando datos de todos los países del mundo sobre múltiples indicadores que pueden ayudar a tomar mejores decisiones.

Debido a esta situación el sistema de salud se encontraba abrumado con posibles limitaciones en la realización de pruebas para la detección de SARS-CoV-2, donde probar cada caso sería poco práctico y los resultados de las pruebas podrían retrasarse, incluso si se estudia una pequeña población. Es por esto que en primera instancia se deseaba desarrollar un modelo que permitiera predecir por medio del primer dataset utilizado, los casos confirmados de COVID-19 entre los casos sospechosos durante una visita a la sala de urgencias, teniendo en cuenta que un caso se considera sospechoso basándose en los resultados de las pruebas de laboratorio comúnmente recolectadas. A pesar de tener el problema predictivo establecido, se presentó una serie de inconvenientes con el set de datos utilizado, ya que mucha de la información que contenía no era significativa, debido a que se tenía mucha información censurada y un gran número de datos inválidos/vacíos, donde al hacer una limpieza con el método .dropna() el dataset quedaba completamente vacío. Ante esta problemática se decide cambiar el set de datos, siguiendo la situación del COVID-19, implementando el repositorio creado en colaboración con la Universidad de Oxford, el cual permitió tener datos actualizados de todos los países, con el fin de realizar un seguimiento ante la problemática ocurrida de la pandemia. En este nuevo set el problema predictivo a resolver, se centró en primera instancia, en intentar conocer por medio de un modelo de regresión la cantidad de nuevas muertes presentada por COVID 19 en países como Argentina, seleccionando este lugar debido a que era la región que presentaba mayor número de datos. Posteriormente, al ver que un modelo de regresión no sería tan óptimo, se estableció un modelo de clasificación, el cual presenta como objetivo predecir el estado actual de la pandemia en base a una clasificación de 3 niveles, para esto el algoritmo debe tener un vector con distintos parámetros y este podrá establecer según las características dadas el nivel de peligro que se predice.

Dataset que se va a utilizar:

El set de datos utilizado en el proyecto, presenta información de la enfermedad del COVID-19, recopilada en diferentes regiones del mundo. El dataset está compuesto por un conjunto de datos .csv, el cual se caracteriza por presentar un tamaño de 166326 filas y 67 columnas, en las cuales se tiene información de la situación de la pandemia en una región en específico, como es el caso del total de muertes, total de vacunados, pacientes hospitalizados, entre otros. En los datos más relevantes se encuentra los siguientes:

- location: Contiene el país a donde está asociado el registro
- date: Contiene la fecha del registro
- total_cases: Contiene el conteo de casos positivos totales hasta el momento
- new_cases: Contiene la información de nuevos casos de Covid reportados.
- total_deaths: Contiene la información de las muertes causadas por el covid hasta ese día.

- new_deaths: Contiene la información de las nuevas muertes reportadas en esa fecha.
- icu_patients: Contiene el conteo de pacientes en UCI hasta ese día.
- hosp_patients: Contiene el conteo de pacientes que han sido hospitalizados hasta el momento.
- people_vaccinated: Contiene el conteo de personas que han sido vacunadas hasta el momento.
- new_vaccinations: Contiene el número de nuevos vacunados ese día.

También se establecieron otras columnas que brindan información respecto a la tasa de contagio, edad de las personas, mortalidad acumulativa, desarrollo humano, etc.

El dataset presenta como variable target (variable objetivo), para el primer modelo la variable new_death. En el caso de los otros modelos diseñados, la variable objetivo pertenece a la clasificación de riesgo, la cual es una variable sintética, creada con el fin de predecir el estado de la pandemia. Es de resaltar que entre los tipos de datos se cuenta con variables de tipo floats (flotante), objects (objeto) e integer (entero).

Para evaluar el algoritmo de aprendizaje diseñado, se implementa una serie de métricas, donde se tiene en cuenta para el caso de modelo de regresión, el error medio absoluto, el cual es una media del valor absoluto de los errores, caracterizándose por ser la métrica más fácil de comprender ya que es el promedio de los errores. También se implementó el error cuadrático medio, y la R cuadrática. Para el error cuadrático medio se presenta la media del error cuadrático. Es más popular que el error de Media absoluto ya que hace foco en grandes errores, debido a que el término cuadrático tiene errores más grandes que van aumentando su valor. La R cuadrática por otro lado, es una medida popular para darle precisión al modelo. Representa cuán cerca están los datos de la línea de regresión ajustada. Mientras más alto el R-cuadrático, mejor se encontrará ajustado el modelo respecto de los datos. El puntaje mejor posible es 1.0 y puede tomar valores negativos. [1]

Para los modelos de clasificación se utilizaron la matriz de confusión, la cual establece una matriz que describe el rendimiento completo del modelo, presentando 4 términos relevantes:

- Verdaderos positivos : los casos en los que predijimos SÍ y el resultado real también fue SÍ.
- Verdaderos negativos : los casos en los que predijimos NO y el resultado real fue NO.
- Falsos positivos : los casos en los que predijimos SÍ y el resultado real fue NO.
- Falsos negativos : los casos en los que predijimos NO y el resultado real fue SÍ. [1]

La matriz de confusión también puede ser vista de la siguiente manera:

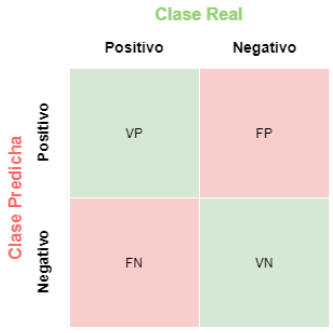


Imagen 1. Matriz de confusión

También se analizaron métricas como el F1-score, el cual hace referencia a la media armónica entre la precisión y la recuperación. El rango para esta métrica es [0, 1] y permite saber qué tan preciso es el clasificador (cuántas instancias clasifica correctamente), así como qué tan robusto es (no pierde una cantidad significativa de instancias). Otra métrica implementada es el recall y la precisión, donde para el recall se tiene en cuenta el número de resultados positivos correctos dividido por el número de todas las muestras relevantes (todas las muestras que deberían haber sido identificadas como positivas). En otra instancia para la precisión se establece el número de resultados positivos correctos dividido por el número de resultados positivos previstos por el clasificador. [1]

Teniendo en cuenta el desempeño deseable del modelo, se establece que si no presenta una exactitud del 95% y una precisión del 90%, no vale la pena ponerlo en producción, debido a la poca fiabilidad de este. Para poder establecer un modelo capaz de determinar el estado de la pandemia con base en unas características específicas, es necesario determinar que los datos con los que se está construyendo sean útiles y aportan al sistema, siendo de suma importancia analizar el set con el que se cuenta. Por otra parte, cuando se elabore el modelo y se entrene de manera adecuada, se espera que sea capaz de determinar con la mayor precisión y exactitud posible, haciendo uso exclusivo de la información que se le suministre.

2. Exploración descriptiva del dataset

Nuestro dataset sigue la línea del covid-19, que contiene información de esta enfermedad recopilada alrededor de algunos países del mundo. Algunas de las columnas que nos llamaron la atención fueron las siguientes:

- location: Contiene el país a donde está asociado el registro
- date: Contiene la fecha del registro
- total_cases: Contiene el conteo de casos positivos totales hasta el momento
- new_cases: Contiene la información de nuevos casos de Covid reportados.
- total_deaths: Contiene la información de las muertes causadas por el covid hasta ese día.
- new_deaths: Contiene la información de las nuevas muertes reportadas en esa fecha.
- icu_patients: Contiene el conteo de pacientes en UCI hasta ese día.
- hosp_patients: Contiene el conteo de pacientes que han sido hospitalizados hasta el momento.
- people_vaccinated: Contiene el conteo de personas que han sido vacunadas hasta el momento.
- new_vaccinations: Contiene el número de nuevos vacunados ese día.

Teniendo en cuenta las columnas de interés, se realizó un análisis descriptivo del set de datos, donde se calculó la media, la mediana y la desviación estándar para el total de muertes, casos covid-19, y para los nuevos casos. Además, se generó un diagrama de barras y uno de bigotes para ver la dispersión de los datos, esto se hizo para las variables mencionadas anteriormente. Al continuar con la exploración del dataset, se encontró que Argentina y México tienen la mayor cantidad de información, al presentar una cantidad de datos de 795.

Posteriormente, se hizo un análisis estadístico, comenzando con la entropía de shannon, la cual permite conocer cómo es la cantidad de información que contiene un arreglo en cuestión,

con el objetivo de identificar si la información que contiene determinada columna es variada, llegando a las siguientes conclusiones:

- La cantidad de información contenida en los datos respecto a muertes es mucho más variada que la información contenida respecto a contagios, debido a que la entropía de Shannon entrega un valor mayor.
- Hay un comportamiento no esperado entre pacientes en UCI, pacientes hospitalizados y nuevos vacunados, pues se está obteniendo el mismo valor de entropía. Esto podría indicarnos que las personas encargadas de realizar el registro obviaron estas columnas y reportaron información muy similar entre ellas, ya que no es normal que dos o más columnas entreguen un valor de entropía similar, por este motivo se estudia este caso posteriormente con otro tipo de pruebas.
- El reporte de personas vacunadas es el campo con menor información de entre todos, esto nos indica que el comportamiento de esto podría ser fácilmente explicado mediante una curva paramétrica, pues sus cambios no son muy significativos en comparación con otros campos.
- La columna de expectativa de vida tiene una entropía de Shannon de magnitud significativa, lo cual puede indicar que la información contenida en esta columna es supremamente variada. Esto se puede corroborar desde la teoría, pues al ser el Covid-19 una enfermedad nueva no era posible establecer una expectativa de vida igual.

También se realizó la entropía de permutación, la cual es una medida que puede determinar la complejidad en series temporales, basándose en la comparación de sus valores vecinos.

Desde la teoría se nos indica que presenta grandes ventajas frente a otros parámetros: rapidez, robustez, simplicidad de cálculo e invarianza con respecto a transformaciones no lineales.
Esta entropía normalmente se usa también en el análisis de señales como EEG, ECG y EMG. Los parámetros normalmente usados son: Tao=5 y D=3, los cuales se implementaron en este proyecto.
De esta prueba se puede llegar a las siguientes conclusiones:

- Aparecen nuevos comportamientos dentro del dataset que complementan lo encontrado con la entropía de Shannon. Por ejemplo: A pesar de que total_cases posee una mayor cantidad de información determinada por la entropía de Shannon, su complejidad para ser estimada es mucho menor en comparación con los casos nuevos, donde ocurre todo lo contrario, es decir, hay menor cantidad de información pero mayor complejidad para ser estimada.
- El parámetro de expectativa de vida parece ser sumamente fácil de estimar. Esto en parte también comprueba lo vivido en la pandemia, pues la expectativa de vida por ejemplo para los adultos mayores era muy baja debido a enfermedades preexistentes y desgaste corporal.

Posteriormente, se realizó una prueba de normalidad y de U mann-whitney. En el caso de la prueba de normalidad se implementa la prueba de Kolmogorov-Smirnov, ya que se tienen más de 300 datos. Esta prueba tiene las siguientes condiciones:

- H0 o Hipótesis nula: Los datos siguen una distribución normal.
- H1 o Hipótesis alternativa: Los datos no siguen una distribución normal.

Para este tipo de pruebas lo normal es establecer un intervalo de confianza del 95%; en este caso se implementó este intervalo con el objetivo de ser lo más precisos posible.
De esta prueba se observa que todos los resultados entregaron un valor de P=0.0, este valor es menor a 0.05 que es el valor del alfa con el que se está trabajando. El hecho de que el pVal sea menor al alfa quiere decir que se rechaza la hipótesis nula, lo que confirma la teoría de que los datos no siguen una distribución normal conocida.

Para el caso de la prueba U maan- Whitney, se caracteriza por ser una prueba que se aplica para distribuciones que no son paramétricas, es decir: No siguen una distribución normal en específico.Esta prueba tiene las siguientes condiciones:

- H0 o Hipótesis nula: Los datos pertenecen a una misma distribución.
- H1 o Hipótesis alternativa: Los datos pertenecen a distribuciones diferentes.

Se implementó este test, con el fin de verificar si las columnas que tienen un mismo valor de entropía de Shannon pertenecen a distribuciones distintas o si por el contrario, poseen los mismos valores.
De los resultados obtenidos se concluye que los valores que entrega el pVal son inferiores al 0.05, lo cual nos lleva a rechazar la hipótesis nula y con ello se establece que la información proviene de distribuciones diferentes. El resultado de esta prueba demuestra que hay independencia entre las columnas analizadas.

Finalmente, se realizó un análisis del covid-19 en Argentina, este se hizo con ayuda de gráficos, donde se estudió principalmente el total de muertes, casos de muertes recientes y las muertes por millón.

3. Iteraciones de desarrollo.

Preprocesado de datos

Durante la elaboración de los modelos se llegó a dos tipos de preprocesados distintos, pues dadas las necesidad (si clasificación o regresión) se debía procesar la información de manera distinta.

El preprocesado para regresión consistió en completar los campos que presentaban información NaN, para eso se decide eliminar las filas que no tuvieran el dato que se esperaba predecir ("new_deaths"), para posteriormente completar los datos faltantes teniendo en cuenta el promedio.
Posteriormente se establecen gráficos de dispersión y una matriz de correlación que permite establecer una relación entre pares de atributos en comparación con el objetivo y para cada atributo en comparación con otros atributos, con el fin de establecer cuáles características podrían generar mejores resultados en el modelo de regresión.
Es de resaltar que se crea un atributo sintético con el fin de establecer una relación entre algunos atributos de entrada y la salida de interés.

El preprocesado para clasificación tuvo diferentes etapas, las cuales fueron:

1. **Rellenado de información faltante:**
   En esta etapa lo que se realizó fue llenar todos los campos con información NaN con ceros. Inicialmente se planteó realizar un .dropna(), sin embargo se perdía información relacionada con los primeros días de la pandemia, por lo cual se preferiría rellenar de ceros la información no disponible y asumir que ese día en concreto no hubo muestreo en esas variables como tal.

2. **Establecer el sistema de clasificación:**
   Lo siguiente que se realizó fue establecer cómo sería el sistema de clasificación, este constaría de 3 niveles los cuales son:
   - **Riesgo elevado (3):** Donde se indica que el estado de la pandemia es altamente peligroso.
   - **Riesgo moderado (2):** Donde indica que el estado de la pandemia es peligroso pero que no representa un riesgo elevado, sin embargo es pertinente realizar monitoreos y prepararse para posibles escaladas en el riesgo.
   - **Riesgo bajo (1):** Donde indica que el estado de la pandemia no es sustancialmente peligroso y se cuentan con las herramientas para combatirlo eficientemente.

   A cada nivel se le asignó un número del 1 al 3 que representa el nivel de riesgo, siendo 1 un riesgo bajo y 3 un riesgo elevado.

3. **Establecer cómo se alimenta el sistema de clasificación:**
   Lo siguiente que se realizó fue idear en qué se iba a basar el sistema de clasificación global, cómo se iba a alimentar y cómo se generaría este coeficiente.

   Se pensó entonces definir a la variable de clasificación general en base a 3 variables de clasificación parciales que seguirán un comportamiento similar al global.

   Las variables de clasificación parciales, su definición matemática y su métrica de evaluación son las siguientes:
   - **Porcentaje de nuevos contagios:**
     Esta variable cuantifica cómo son los nuevos contagios en relación a la población total del país.

     Matemáticamente se define como:

     $$Métrica\ porcentual = \frac{Nuevos\ casos}{Población} \times 100 \times Factor\ de\ escalado$$

     Los nuevos casos corresponden al valor de la columna *new_cases* y la población corresponde al valor de la columna *population*. Se usó un factor de escalado de 1000 con el fin de amplificar el valor de la métrica y así poder realizar una clasificación más óptima.

     Luego de obtener el valor porcentual de la métrica el siguiente paso fue entonces clasificarla en 3 niveles, de forma similar a la métrica global de

clasificación, sin embargo aquí se usaron umbrales para establecer estos límites, por ello entonces la clasificación fue la siguiente:

- **Riesgo elevado (3):** Si el valor de la métrica porcentual es mayor al 20%.
- **Riesgo moderado (2):** Si el valor de la métrica porcentual es mayor al 5% pero menor o igual al 20%.
- **Riesgo bajo (1):** Si el valor de la métrica porcentual es menor o igual al 5%

De esta forma entonces se definió el valor que tomaría este criterio, siendo estos 1,2 y 3 dependiendo del nivel de amenaza.

- **Porcentaje de mortalidad:**
  Esta variable busca cuantificar qué tan mortal está siendo la enfermedad, es por ello que usar información de las nuevas muertes y los nuevos casos. Matemáticamente se define como:

$$Métrica\ porcentual = \frac{Nuevas\ muertes}{Nuevos\ casos} \times 100$$

Las nuevas muertes corresponden al valor de la columna *new_deaths* y los nuevos casos corresponden al valor de la columna *new_cases*. En este caso no se usó un factor de escalado debido a que la métrica porcentual era muy diciente por sí misma, es decir, ya presentaba valor en el intervalo de 1 a 100 y no en decimales de 0.

Luego de obtener el valor porcentual de la métrica el siguiente paso fue entonces clasificarla en 3 niveles, de forma similar a la métrica global de clasificación, sin embargo aquí se usaron umbrales para establecer estos límites, por ello entonces la clasificación fue la siguiente:

- **Riesgo elevado (3):** Si el valor de la métrica porcentual es mayor al 10%.
- **Riesgo moderado (2):** Si el valor de la métrica porcentual es mayor al 3% pero menor o igual al 10%.
- **Riesgo bajo (1):** Si el valor de la métrica porcentual es menor o igual al 3%

De esta forma entonces se definió el valor que tomaría este criterio, siendo estos 1,2 y 3 dependiendo del nivel de amenaza.

- **Porcentaje de UCI:**
  Esta variable busca cuantificar cómo es la relación entre las admisiones semanales a UCI con los nuevos casos. Matemáticamente se define como:

$$Métrica\ porcentual = \frac{Admisiones\ semanales\ a\ UCI}{Nuevos\ casos} \times 100$$

Las admisiones semanales a UCI corresponden al valor de la columna *weekly_icu_admissions* y los nuevos casos corresponden al valor de la

columna *new_cases*. De forma similar a la métrica anterior aquí no se utilizó un factor de escalado.

Luego de obtener el valor porcentual de la métrica el siguiente paso fue entonces clasificarla en 3 niveles, de forma similar a la métrica global de clasificación, sin embargo aquí se usaron umbrales para establecer estos límites, por ello entonces la clasificación fue la siguiente:

- **Riesgo elevado (3):** Si el valor de la métrica porcentual es mayor al 15%.
- **Riesgo moderado (2):** Si el valor de la métrica porcentual es mayor al 5% pero menor o igual al 15%.
- **Riesgo bajo (1):** Si el valor de la métrica porcentual es menor o igual al 5%

De esta forma entonces se definió el valor que tomaría este criterio, siendo estos 1,2 y 3 dependiendo del nivel de amenaza.

4. **Cálculo del coeficiente de clasificación general:**
   Finalmente, se estableció que el coeficiente de clasificación general de cada registro estaría dado por el promedio de los coeficientes de clasificación parciales, luego se aplicará un redondeo con cero decimales, de forma que los valores resultantes únicamente fueran enteros pertenecientes al intervalo [1,3].

   Este coeficiente indicaría entonces qué tan peligroso es el estado de la pandemia, pues a nivel profundo se alimenta con valores recopilados y de acceso a los sistemas de salud de cada país.

Modelos supervisados

1. ***Primer modelo: Regresión lineal:***
   La regresión lineal es un algoritmo de aprendizaje supervisado que se utiliza en Machine Learning y en estadística, en la cual se presenta una aproximación para modelar la relación entre una variable escalar dependiente “y” y una o más variables explicativas nombradas con “X”. [2]

   En este caso se presenta como variable dependiente, la cantidad de nuevas muertes en el país de Argentina ("new_deaths"). Como variables explicativas se tiene en cuenta: "total_cases", "new_cases", "total_deaths", "people_vaccinated_per_hundred", "people_fully_vaccinated_per_hundred", "total_vaccinations","people_vaccinated", "people_fully_vaccinated", "new_cases_per_million" y un atributo sintetico establecido como "death_rate".

   Es de resaltar que para realizar el modelo, se divide el set de datos, en entrenamiento y test, los cuales en primera instancia se separan utilizando un 80% para entrenamiento y un 20% para test o validación. En segundo lugar se dividen los datos implementando la función bootstrap, la cual es una técnica de remuestreo que consiste en extraer repetidamente muestras de los datos de origen con reemplazo, para esto se tiene una división de las muestras del 30% para los datos de test.

2. ***Segundo modelo: Árboles de decisión***

   Los árboles de decisión son algoritmos estadísticos o técnicas de machine learning que nos permiten la construcción de modelos predictivos de analítica de datos para el Big Data basados en su clasificación según ciertas características o propiedades, o en la regresión mediante la relación entre distintas variables para predecir el valor de otra [3].

   En este modelo se crearon variables auxiliares de clasificación, las cuales son las siguientes: Porcentaje de casos, porcentaje de muertes, porcentaje de nuevos casos, porcentaje de UCI, porcentaje de mortalidad, porcentaje de inmunización, peso de los parámetros. Posteriormente, se construyó el dataset de procesamiento con los datos mencionados anteriormente, asignando una nueva variable, la cual será la ponderación de estos valores. Luego, se añade esta columna al dataset principal.

   Para la realización del modelo, Debido al gran volumen de datos, el set se va a dividir en 3 partes, de la siguiente forma:

   - Set de entrenamiento: 70%
   - Set de prueba: 30%

   Estos valores serán separados aleatoriamente por software, de esta forma se busca eliminar posibles problemas como el sesgo. Fue necesario tomar datos de países aleatorios para poder realizar el entrenamiento debido a que Argentina no llegó a los niveles críticos. Se van a tomar 500 datos de cada tipo, es decir, 500 con clasificación de 1, 500 con clasificación de 2 y 500 con clasificación de 3.

3. ***Tercer modelo: Perceptrón multicapa***

   El perceptrón multicapa es básicamente una red neuronal artificial, que como su nombre lo indica, se compone de múltiples capas, cada una con un nivel específico de neuronas. Este tipo de modelo busca resolver problemas cuya clasificación no sea específicamente lineal.[4]

   De forma similar al anterior modelo, se crearon variables de clasificación sintéticas, estas fueron: Porcentaje de nuevos casos, Porcentaje de mortalidad y Porcentaje de UCI, variables mencionadas anteriormente.

   De igual forma se dividió el set de forma aleatoria en 70% entrenamiento y 30% datos para prueba. El set inicialmente buscaba contener información únicamente de Argentina (Que es el foco en los algoritmos de regresión), sin embargo los datos disponibles indican que Argentina no posee los 3 niveles de riesgo, es por ello que es ineficiente usarlos para entrenamiento y prueba, por ello hubo que tomar datos de forma aleatoria, generando entonces un set de 5000 datos con nivel 1, 5000 con nivel 2 y 5000 con nivel 3, set que sería entonces dividido en los valores anteriormente mencionados (70% entrenamiento y 30% prueba).

   Este algoritmo hace uso de las columnas *new_cases*, *new_deaths*, *weekly_icu_admissions* y *population*.

4. ***Cuarto modelo: Máquinas de soporte vectorial***

   Las máquinas de soporte vectorial (SVM) son una serie de algoritmos de aprendizaje supervisado, enfocados usualmente con problemas que involucran clasificación y regresión. Se alimenta de datos etiquetados en clases, con el objetivo de construir un modelo capaz de predecir la clase de una nueva muestra. Una SVM presenta los datos como puntos en el espacio, separando las clases por medio de hiperplanos que se definen mediante vectores (de aquí el nombre), con los cuales buscan la correspondencia del valor con un grupo o no. [5]

   Este modelo fue realizado siguiendo las mismas indicaciones del modelo 3 (Perceptrón multicapa), pues se usaron los mismos datos, las mismas columnas y se dividieron los datos de forma similar.

5. ***Quinto modelo: regresión logística***

   La regresión logística es un método estadístico que se usa para resolver problemas de clasificación binaria, donde el resultado solo puede ser de naturaleza dicotómica, o sea, solo puede tomar dos valores posibles. Por ejemplo, se puede utilizar para detectar la probabilidad de que ocurra un evento [6].

   En este modelo se crearon variables auxiliares de clasificación, las cuales son las siguientes: Porcentaje de casos, porcentaje de muertes, porcentaje de nuevos casos, porcentaje de UCI, porcentaje de mortalidad, porcentaje de inmunización, peso de los parámetros. Posteriormente, se construyó el dataset de procesamiento con los datos mencionados anteriormente, asignando una nueva variable, la cual será la ponderación de estos valores. Luego, se añade esta columna al dataset principal.

   Para la realización del modelo, Debido al gran volumen de datos, el set se va a dividir en 3 partes, de la siguiente forma:

   - Set de entrenamiento: 70%
   - Set de prueba: 30%

   Estos valores serán separados aleatoriamente por software, de esta forma se busca eliminar posibles problemas como el sesgo. Fue necesario tomar datos de países aleatorios para poder realizar el entrenamiento debido a que Argentina no llegó a los niveles críticos. Se van a tomar 500 datos de cada tipo, es decir, 500 con clasificación de 1, 500 con clasificación de 2 y 500 con clasificación de 3.

Resultados, métricas y curvas de aprendizaje.

1. ***Primer modelo: Regresión lineal:***

   Estableciendo la división de los datos utilizando un 80% para entrenamiento y un 20% para test o validación, se obtienen las métricas presentadas en la tabla 1.

Tabla 1: Métricas obtenidas al dividir el set de datos sin utilizar la función bootstrap

Error medio absoluto	Suma residual de los cuadrados (MSE)	R2- score	Promedio de validación
439.35	627055.12	-0.22	-80.9822

Utilizando la función bootstrap para la división de los datos implementando, se registra lo evidenciado en la tabla 2.

Tabla 2: Métricas obtenidas al dividir el set de datos al implementar la función bootstrap

Error medio absoluto	Suma residual de los cuadrados (MSE)	R2- score
86.1925	12642.2865	0.187

Debido al valor de las métricas obtenidas se puede evidenciar que se tienen mejores resultados al utilizar para la división de los datos de entrenamiento y de prueba la función bootstrap, ya que presenta errores más bajos y un valor de R2-score de valor mayor y positivo, siguiendo esto se realizó una curva de aprendizaje de dicho modelo, obteniendo la imagen 2.

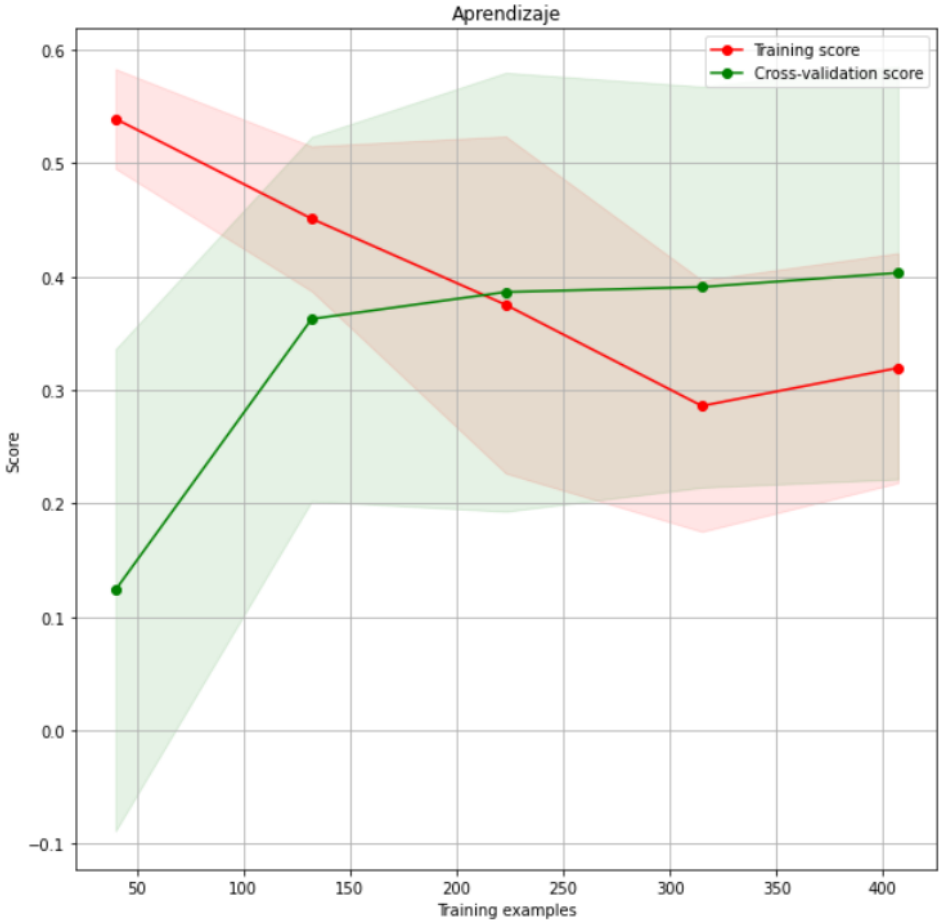


Imagen 2. Curva de aprendizaje del primer modelo

Las curvas de aprendizaje permiten conocer la estabilidad del modelo. En el modelo anterior se puede evidenciar un caso de bias o sesgo, ya que los modelos no tienden a presentar una convergencia adecuada, es decir los datos no presentan una mejora en los datos de validación del modelo utilizado y los datos de entrenamiento tienden a disminuir, pero un valor de aproximadamente 0.28 presentan un aumento.

2. *Segundo modelo: Árboles de decisión*

Estableciendo la división de los datos utilizando un 70% para entrenamiento y un 30% para test o validación, se obtienen las métricas presentadas en la tabla 3.

Tabla 3: Métricas obtenidas al dividir el set de datos

Recall	f1-score	precisión	accuracy
0.94	0.94	0.96	0.966

Se puede evidenciar un alto recall, por lo tanto nuestro algoritmo detecta bien la clase, pero también incluye muestras de otras clases. Además, contamos con una precisión del 96%, lo que se considera aceptable. Finalmente, el f1-score es de gran utilidad dado que, en este caso tenemos una distribución de clases desigual, por lo tanto con un porcentaje de 94%, nos indica una buena división.

La curva de aprendizaje para este modelo nos indica lo contrario, donde podemos observar un sesgo, lo cual nos indica que el modelo hizo suposiciones sobre los datos de entrenamiento. Esto conduce a una simplificación excesiva del modelo y puede provocar un error elevado tanto en los conjuntos de entrenamiento como de prueba. Sin embargo, esto también hace que el modelo sea más rápido de aprender y fácil de entender.

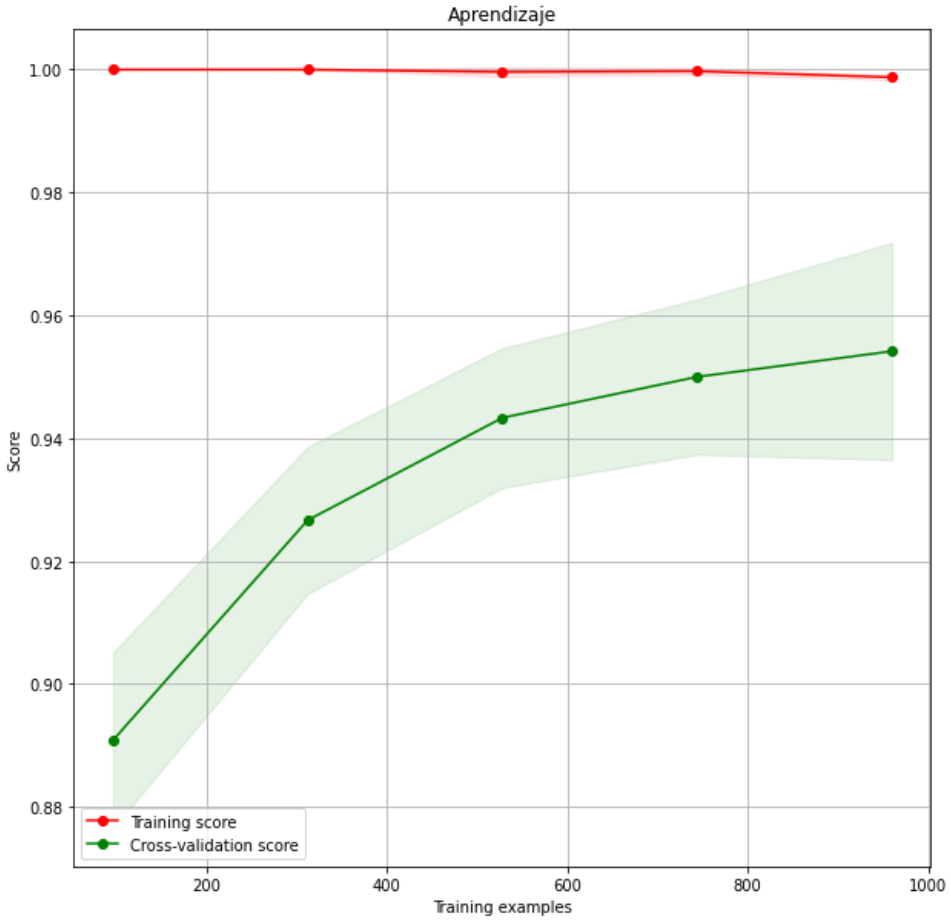


Imagen 3. Curva de aprendizaje del segundo modelo.

3. ***Tercer modelo:***

Se generó una iteración, donde el algoritmo iría evaluando la cantidad de neuronas, de 1 a 30 neuronas, de forma similar con el número de capas, siendo esta de 1 a 30 capas, construyendo así el siguiente dataset:

	Capas	Neuronas	Precision train	Recall train	F1-Score train	Precision test	Recall test	F1-Score test
0	1.0	1.0	0.110159	0.333333	0.165593	0.113333	0.333333	0.169154
1	1.0	2.0	0.115360	0.333333	0.171402	0.102450	0.333333	0.156729
2	1.0	3.0	0.448693	0.337243	0.179131	0.435783	0.335430	0.160896
3	1.0	4.0	0.106308	0.324536	0.160154	0.116331	0.327044	0.171617
4	1.0	5.0	0.109872	0.312200	0.162541	0.110070	0.307190	0.162069
...	...	...	...	...	...	...	...	...
895	30.0	26.0	0.587983	0.587491	0.587710	0.572375	0.572533	0.572450
896	30.0	27.0	0.609330	0.597691	0.600515	0.594232	0.590399	0.591041
897	30.0	28.0	0.654902	0.622683	0.628873	0.642383	0.627107	0.630581
898	30.0	29.0	0.639290	0.599778	0.600795	0.661744	0.632361	0.632208
899	30.0	30.0	0.586399	0.570081	0.575467	0.592911	0.583307	0.585051

900 rows × 8 columns

Imagen 4. Dataset resultado del tercer modelo.

De aquí se seleccionó de forma artificial el mejor modelo, buscando que los valores de cada métrica fueran los máximos. Con ello se llega entonces al siguiente resultado:

	Capas	Neuronas	Precision train	Recall train	F1-Score train	Precision test	Recall test	F1-Score test
863	29.0	24.0	0.661835	0.602895	0.607093	0.653537	0.618005	0.618055

Imagen 5. Mejor configuración del modelo 3.

Indicando entonces que el mejor modelo de este conjunto fue aquel con 29 capas, cada una con 24 neuronas.

Las matrices de confusión resultantes son las siguientes:

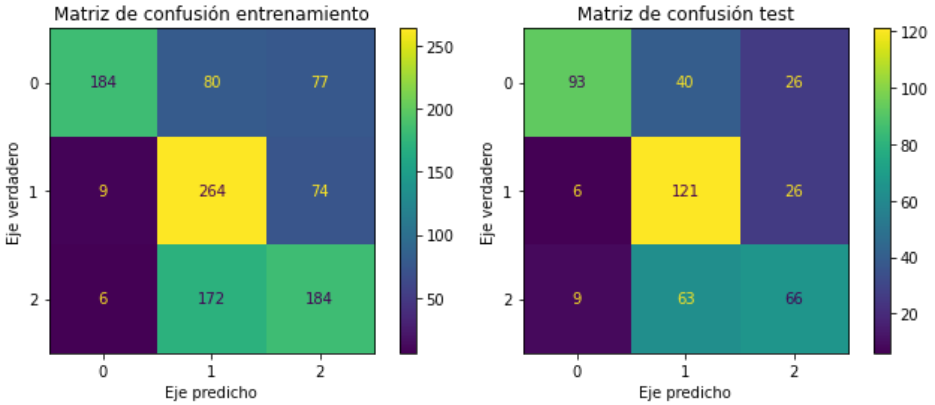


Imagen 6 y 7. Matrices de confusión del modelo 3.

En estas matrices se muestra que el algoritmo predice de forma eficiente el menor nivel (aquí etiquetado como 0), y que es capaz de separar el nivel de riesgo intermedio (aquí etiquetado como 1); sin embargo, sufre más cuando los datos están entre un riesgo moderado y elevado, y esto era de esperarse debido a que los valores

están muy cerca entre sí, por lo tanto el perceptrón multicapa no generó un buen modelo para este problema.

La curva de aprendizaje fue la siguiente:

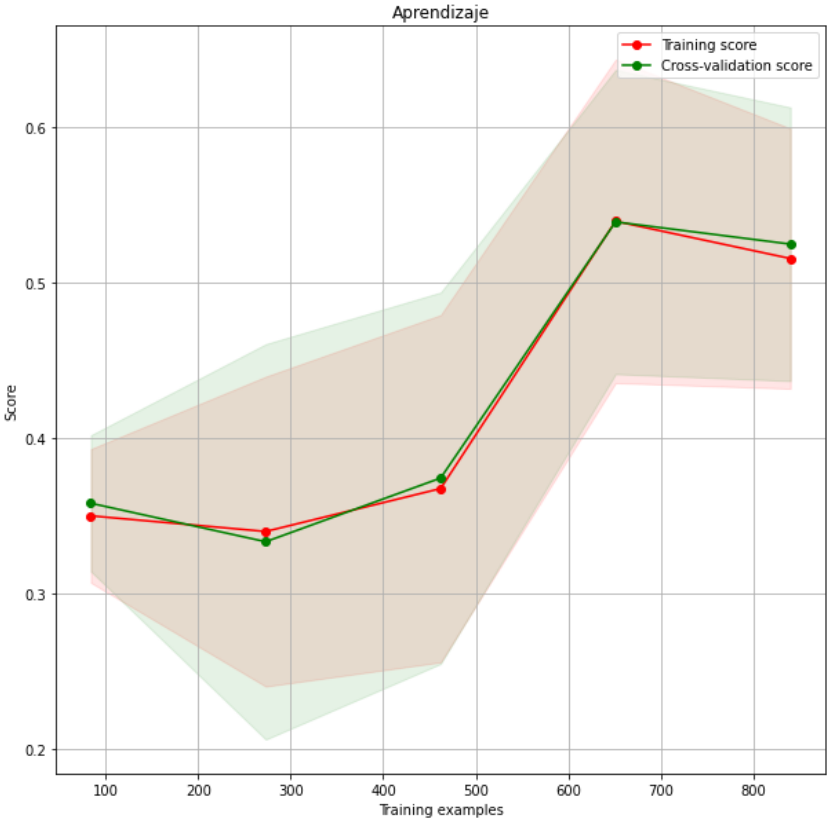


Imagen 8. Curva de aprendizaje del tercer modelo.

La curva de aprendizaje nos muestra una pequeña convergencia entre algunos puntos, sin embargo, presenta un sesgo. Es de resaltar que en la parte final de la curva se observa que los valores tienden a disminuir.

4. ***Cuarto modelo:***

Se generó una iteración, donde el algoritmo iría evaluando el valor de C, de 1.0 a 5.0 con saltos de 0.1, y variando el tipo de gamma como *auto* o como *scale*. El dataset resultante es el siguiente:

	C	Gamma	Precision train	Recall train	F1-Score train	Precision test	Recall test	F1-Score test
0	1.0	auto	0.998981	0.999066	0.999022	0.737209	0.415091	0.310287
1	1.0	scale	0.279627	0.395454	0.318536	0.274613	0.427537	0.328006
2	1.1	auto	0.998981	0.999066	0.999022	0.729031	0.418332	0.319325
3	1.1	scale	0.283988	0.401056	0.322630	0.278075	0.432199	0.331469
4	1.2	auto	0.998981	0.999066	0.999022	0.729031	0.418332	0.319325
...	...	...	...	...	...	...	...	...
77	4.8	scale	0.288409	0.406659	0.326713	0.288678	0.446185	0.341832
78	4.9	auto	1.000000	1.000000	1.000000	0.729031	0.418332	0.319325
79	4.9	scale	0.288409	0.406659	0.326713	0.288678	0.446185	0.341832
80	5.0	auto	1.000000	1.000000	1.000000	0.729031	0.418332	0.319325
81	5.0	scale	0.288409	0.406659	0.326713	0.288678	0.446185	0.341832

82 rows × 8 columns

Imagen 9. Dataset de configuraciones del cuarto modelo.

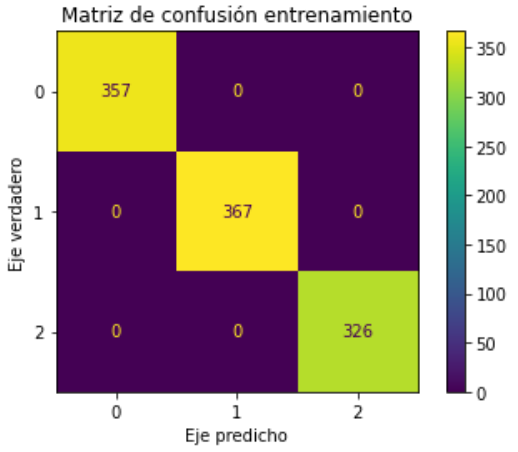
De forma similar al modelo anterior, se extrajo el mejor modelo de forma sintética, arrojando entonces el siguiente resultado:

	C	Gamma	Precision train	Recall train	F1-Score train	Precision test	Recall test	F1-Score test
6	1.3	auto	1.0	1.0	1.0	0.729031	0.418332	0.319325

Imagen 10. Mejor configuración de parámetros del cuarto modelo.

Significa entonces que el mejor modelo es aquel con un valor de C=1.3 y un Gamma calculado automáticamente.

Las matrices de confusión son las siguientes:



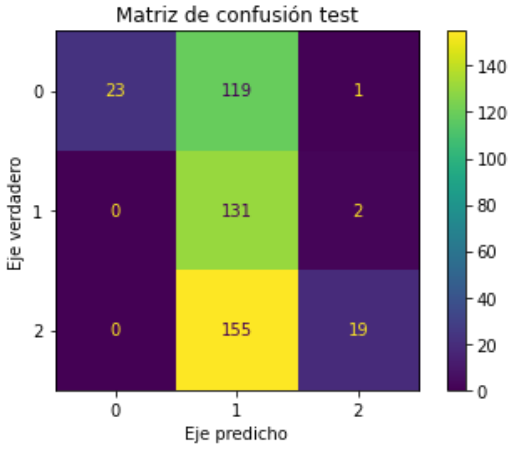


Imagen 11 y 12. Matrices de confusión del modelo 4.

Como se observa, parece que el modelo se aprende de memoria los datos pues la matriz de confusión de entrenamiento es perfecta, sin embargo, al aplicar los datos de test, se evidencia que el algoritmo realmente no está siendo capaz de clasificar correctamente los datos, y que se encuentra sesgado, pues la gran parte de sus clasificaciones las hace como un riesgo intermedio (reflejado en la matriz con el valor 1 en los ejes). Al igual que con el anterior modelo, esto puede evidenciar que la diferencia entre los distintos grupos no es tan fácilmente discriminable por un algoritmo de este tipo.

La curva de aprendizaje que se obtiene es la siguiente:

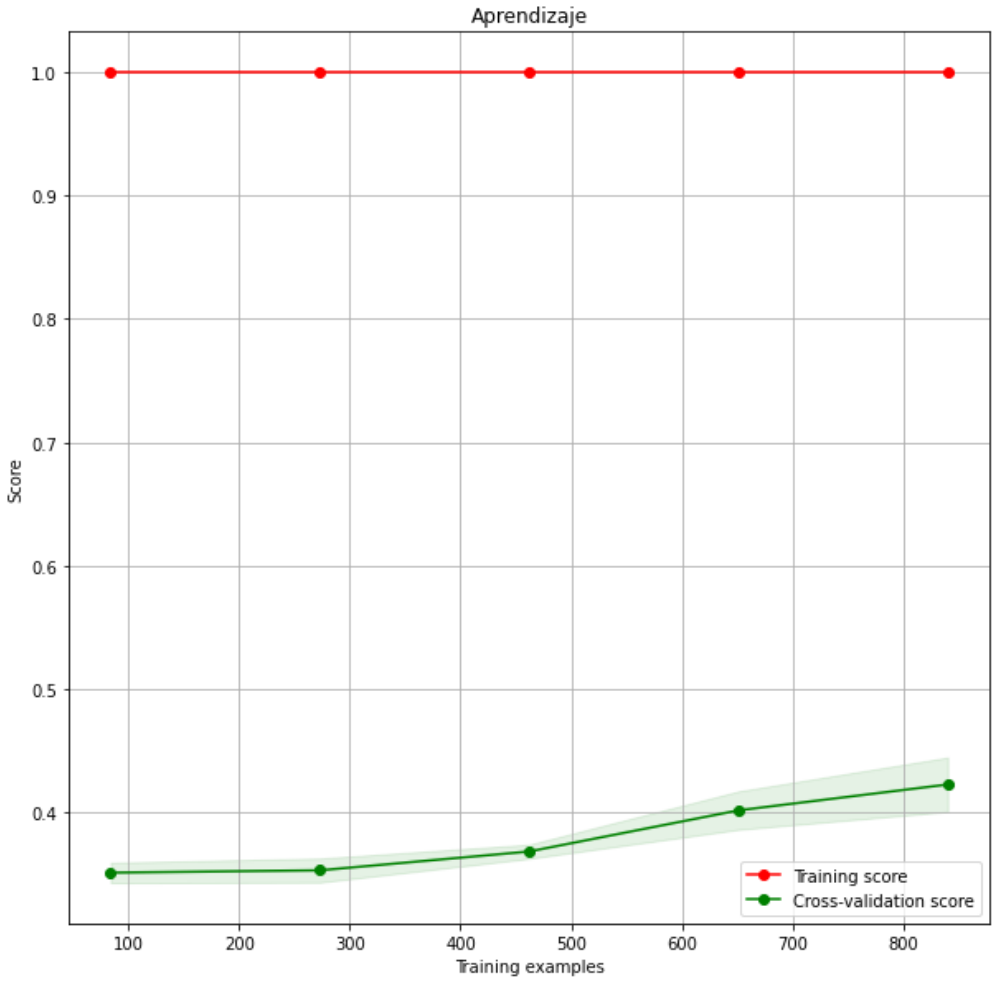


Imagen 13. Curva de aprendizaje del cuarto modelo.

La curva de aprendizaje para este modelo nos indica un sesgo, por lo que el modelo hizo suposiciones sobre los datos de entrenamiento. Esto conduce a una simplificación excesiva del modelo y puede provocar un error elevado tanto en los conjuntos de entrenamiento como de prueba. Sin embargo, esto también hace que el modelo sea más rápido de aprender y fácil de entender.

5. ***Quinto modelo: Regresión logística***

Estableciendo la división de los datos utilizando un 70% para entrenamiento y un 30% para test o validación, se obtienen las métricas presentadas en la tabla 4.

Tabla 4: Métricas obtenidas al dividir el set de datos

Recall	f1-score	precisión	accuracy
0	0.54	0.37	0.3767

Se puede evidenciar un recall de 0, lo cual sugiere que el modelo no fue capaz de identificar. Además, contamos con una precisión del 37%, por tanto nuestro modelo

no detecta las clases muy bien. Finalmente, el f1-score está por debajo del 60%, lo que nos quiere decir que el modelo no es aceptable.

Al igual que lo visto en el segundo modelo, presenta un sesgo, sin embargo, se puede apreciar un pequeño punto de convergencia.

Curva de aprendizaje:

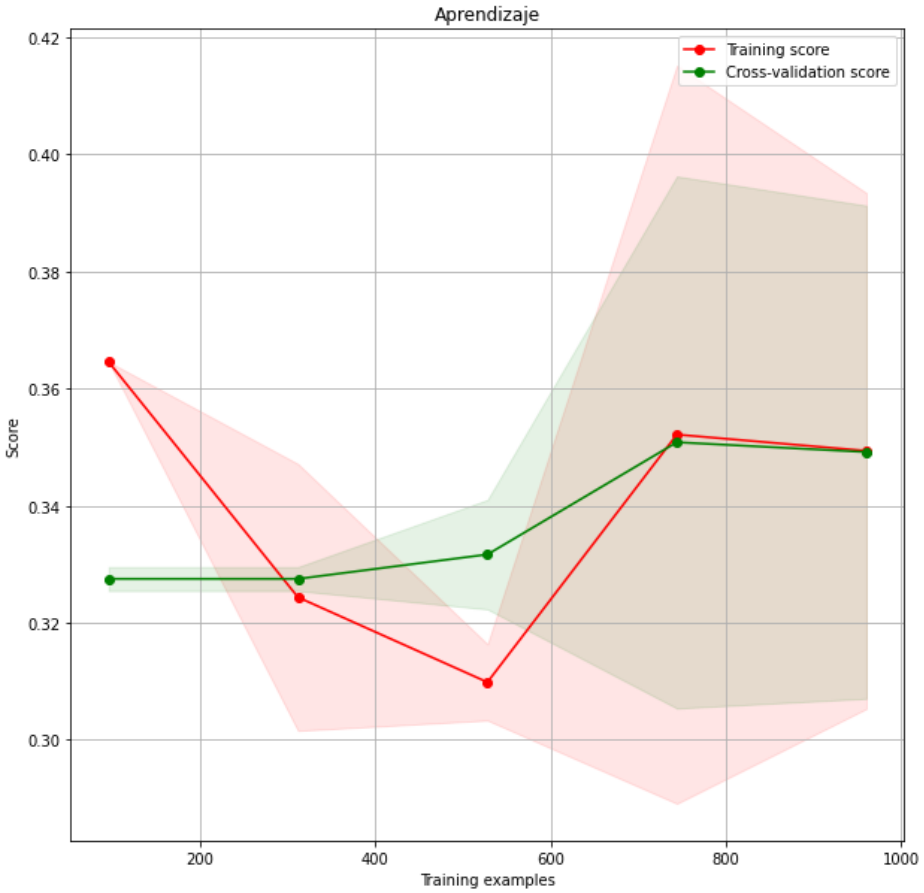


Imagen 14. Curva de aprendizaje del quinto modelo.

Al igual que lo visto en el segundo modelo, presenta un sesgo, sin embargo, se puede apreciar un pequeño punto de convergencia.

4. Retos y consideraciones de despliegue

El potencial de la inteligencia artificial, aunque enorme, está limitado por una serie de factores.Algunos son técnicos y otros muy humanos. Es por ello que durante nuestro proyecto encontramos algunos limitantes como lo son los recursos necesarios para ejecutar la predicción al momento de procesar una gran cantidad de datos, va siempre por delante de las capacidades del hardware, por lo que siempre nos encontraremos con un importante cuello de botella.

Se encontró que no todos los problemas son fácilmente clasificables, pues los datos se encuentran mezclados y lo que podría suponer una tarea medianamente fácil de clasificación para un humano, puede convertirse en una tarea compleja incluso para una máquina.

Para desplegar un algoritmo de este tipo es necesario validar pertinentemente los modelos, de forma que los resultados que arrojen sean consecuentes con los reales, pues de esto supone entonces la fiabilidad y confianza del sistema en cuestión.

5. Conclusiones

- El aprendizaje automático es un campo de la inteligencia artificial que se encarga del desarrollo de algoritmos que pueden aprender de los datos y hacer predicciones sobre ellos, siendo de gran ayuda en diferentes situaciones como es el caso de la emergencia ocasionada por el covid 19, donde se implementaron múltiples indicadores con el fin de ayudar a tomar mejores decisiones.
- Para realizar predicciones precisas, es importante tener datos de alta calidad que sean representativos, siendo relevante establecer divisiones de datos de entrenamiento y datos de test, implementando funciones como la técnica de remuestreo bootstrap, que utiliza el muestreo aleatorio, con el objetivo de evitar problemas de overfitting y mejorar la estabilidad de los algoritmos de aprendizaje automático.
- Los árboles de decisión son algoritmos fácilmente entendibles y que funcionan bien en modelos analíticos basados en clasificación o regresión para obtener resultados a un problema.
- El perceptrón multicapa es uno de los modelos que más busca comprender cómo es el comportamiento del sistema, más que simplificarlo.
- Las máquinas de soporte vectorial tienden a simplificar el problema, construyendo hiperplanos que a simple vista parecen perfectos, pero que cuando se prueban con datos de prueba reflejan sobreentrenamiento, sesgo y predicciones incorrectas.
- Las curvas de aprendizaje son una excelente herramienta de diagnóstico para determinar el sesgo y la varianza en un algoritmo de machine learning supervisado.

Bibliografía:

1. Bajaj, Aayush. “Performance Metrics in Machine Learning [Complete Guide] - neptune.ai.” Neptune.ai, 21 July 2022, https://neptune.ai/blog/performance-metrics-in-machine-learning-complete-guide. Accessed 9 November 2022.
2. “Regresión Lineal.” Aprende Machine Learning, 13 May 2018, https://www.aprendemachinelearning.com/tag/regresion-lineal/. Accessed 9 November 2022.
3. Unir, V. (2021, 19 octubre). Árboles de decisión: en qué consisten y aplicación en Big Data. UNIR. https://www.unir.net/ingenieria/revista/arboles-de-decision/ . Accessed 9 November 2022.
4. Multicapa Sobreaprendizaje Perceptron Neuronas. (s. f.). https://web.archive.org/web/20140714231842/http://www.lab.inf.uc3m.es/%7Ea0080630/redes-de-neuronas/perceptron-multicapa.html
5. Mendoza, J. (2020, 23 noviembre). ¿Qué es un modelo SVM? https://estadisticamente.com/que-es-un-modelo-svm/
6. Buitrago, B. (2021, 15 diciembre). Regresión Logística I — Machine Learning - iWannaBeDataDriven. Medium. https://medium.com/iwannabedatadriven/regresi%C3%B3n-log%C3%ADstica-i-machine-learning-84ffe9d6be15. Accessed 9 November 2022.