



**SEP**  
SECRETARÍA DE  
EDUCACIÓN PÚBLICA



Culiacán Sin. 30/05/2020

# INTELIGENCIA ARTIFICIAL

## UNIDAD5: -REDES NEURONALES

TEMA:  
PROYECTO FINAL REDES NEURONALES ENE-JUL 2020.



INSTITUTO  
TECNOLÓGICO  
DE CULIACÁN

INTEGRANTES:  
AGUIRRE CASTRO ANDRÉS  
CORTEZ ESPARZA EDUARDO ALEJANDRO  
DIAZ VILLEGAS FRANCISCO JAVIER  
REYES PEÑA PEDRO CESAR  
ING. SISTEMAS | HORA 09:00 A 10:00 | GRUPO: EB02  
PROFESORA: NORA ESMERALDA CANCELA GARCIA



## **DESCRIPCIÓN DEL PROYECTO**

Investigar algún framework, paquete o librería para la construcción de Redes Neuronales. Describir sus características y realizar algún ejemplo simple con los datos que actualmente se tienen de COVID-19. Puede ser por ejemplo a partir de la edad, sexo, país, o región, enfermedades o padecimientos crónicos, diabetes, hipertensión u obesidad, determinar el riesgo de muerte. El secretario de salud de Sinaloa publica diariamente los detalles de los fallecimientos, con esto se puede hacer el ejercicio.

El objetivo será aterrizar los conceptos analizados en clase de las redes neuronales y tener nuestro primer contacto con una herramienta que me permita su creación. También analizaremos la importancia de los datos para que la red se entrene y los resultados que obtenemos al probarla.

## **EL OBJETIVO DE LA RED**

### **¿Cuál es el principal objetivo?**

El principal objetivo de este proyecto fue hacer un análisis y sobre cuáles son las características que tiene una persona para tener riesgo de muerte por el virus COVID-19 que a la fecha nos mantiene en aislamiento, en un ejemplo simple si una persona cuenta con problemas de salud cual sería el porcentaje de sufrir muerte por padecer de este virus contando con una edad muy adulta o teniendo enfermedades como la diabetes entre otras.

### **¿Qué vamos a intentar predecir?**

El dato que intentamos predecir es si el individuo muere o no en base a su historial de datos, esto lo hicimos gracias a una recopilación de datos sobre el COVID-19 proporcionado por el gobierno de México, estos datos fueron analizados con detenimiento en lo particular nos llamó la atención una columna que decía; “**fecha de defunción**” la cual es la variable que indica si la persona murió o no, para mejorar esto lo que hicimos fue remplazar el tipo de esta columna para que fuera

más fácil de manipular en nuestra red neuronal, originalmente este dato era en formato fecha lo que hicimos fue crear una columna nueva a partir de esta que se llamaba “Muerte” la cual tomamos a partir de la fecha si la fecha era “9999-99-99” significaba que el sujeto no murió y la variable era igual a 0 y si la fecha correspondía a una real significaba que el sujeto había muerto la variable era 1, ahora bien con base a las otras variables podremos tratar de predecir esta columna en nuevos casos de gente que padezca el virus.

### **¿Cuáles son las características objetivo?**

Las características que utilizamos para el objetivo fueron datos como la edad, sexo, si el paciente sufre de diabetes, si sufre de una enfermedad cardiaca, tiene embarazo, sufre de obesidad, cuenta con asma, sufre de hipertensión, tabaquismo, etc.

### **¿Cuáles son los datos de entrada?**

Como se dijo previamente los datos de entrada provienen de un csv proporcionado por el gobierno de México que a la fecha de que se realizó el proyecto cuenta con aproximadamente 254794 registros, los cuales usaremos para alimentar a nuestra red neuronal, las columnas o variables que se leen de entrada sobre los pacientes son los siguientes.

ORIGEN, EDAD, SECTOR, ENTIDAD\_UM, SEXO, ENTIDAD\_NAC, ENTIDAD\_RES, MUNICIPIO\_RES, TIPO\_PACIENTE, INTUBADO, NEUMONIA, NACIONALIDAD, EMBARAZO, HABLA LENGUA\_INDIG, DIABETES, EPOC, ASMA, INMUSUPR, HIPERTENSION, OTRA\_COM, CARDIOVASCULAR, OBESIDAD, RENAL\_CRONICA, TABAQUISMO, MUERTO

## ¿Están disponibles?

¡Claro que están disponibles! actualmente se encuentran en el siguiente portal.

Datos Abiertos

SALUD



La Secretaría de Salud es la dependencia del Poder Ejecutivo que se encarga primordialmente de la prevención de enfermedades y promoción de la salud

Conjunto de datos

Actividad

Grupos

Reportar

Información referente a casos COVID-19 en México

Datos y Recursos

Bases de datos COVID-19

Información del Sistema de Vigilancia Epidemiológica de Enfermedades Respiratoria Viral, que informan las 475 unidades monitoras de enfermedad...

Descargar

Más información

Metadatos

Campo	Valor
-------	-------

Puede acceder a ellos rápidamente desde la siguiente liga:

<https://datos.gob.mx/busca/dataset/informacion-referente-a-casos-covid-19-en-mexico>

## ¿A qué clase de problema nos enfrentamos?



- **¿Qué es exactamente el coronavirus?**

Los coronavirus, llamados así porque tienen forma de corona, son una familia amplia de virus comunes en muchas especies de animales. Varios coronavirus pueden infectar a las personas, según los CDC. En su mayoría, estas cepas causan síntomas similares al resfriado, pero pueden a veces desarrollarse en enfermedades más complejas del tracto respiratorio inferior, como neumonía o bronquitis.

En raras ocasiones, los coronavirus en los animales pueden evolucionar y propagarse entre los humanos, como se vio con el MERS y el SARS. Los funcionarios de salud hacen referencia al virus en el centro de este último brote como a un nuevo (novel) coronavirus, ya que es algo que no habían visto antes.

- **¿Cómo se está propagando el coronavirus?**

El virus se transmite primordialmente de persona a persona por las gotitas de respiración de la tos y los estornudos de alguien infectado. Según los CDC, esas gotitas pueden caer en la boca o nariz de las personas que se encuentran cerca o pueden inhalarse en los pulmones. Puede que sea posible contraer el COVID-19 al tocar una superficie u objeto contaminado y luego tocarse la boca, nariz u ojos, "pero el virus se puede propagar de muchas otras formas", informan los CDC.

Sin embargo, se cree que las personas son más contagiosas cuando están enfermas con los síntomas del virus, dicen los CDC. Según un estudio basado en la población de Islandia, por ejemplo, el 43% de los participantes que dieron positivo al virus indicaron no tener síntomas cuando se les administró la prueba. De manera similar, según una encuesta basada en la población de Indiana, el 45% de los casos positivos no tenían síntomas al momento de realizarse la prueba, y alrededor del 18% de las personas a bordo del crucero Diamond Princess nunca desarrollaron síntomas de una infección por coronavirus.

- **¿Qué pueden hacer los adultos mayores para reducir el riesgo de enfermarse?**

Los adultos mayores y las personas con trastornos crónicos de salud, como diabetes y enfermedades cardíacas y pulmonares, son más propensos que las personas jóvenes y saludables a experimentar síntomas graves de la enfermedad causada por el coronavirus (COVID-19). Las personas con sistemas inmunes comprometidos y aquellas que viven en hogares de ancianos o centros de cuidado a largo plazo también tienen un mayor riesgo de sufrir complicaciones por la COVID-19.

Además de las enfermedades crónicas, el riesgo de muerte es mayor entre los adultos mayores. Según estadísticas de los Centros para el Control y la Prevención de Enfermedades (CDC), 8 de cada 10 de las muertes reportadas han sido de adultos de 65 años o mayores.

- **¿Pueden los niños o los adolescentes contraer la COVID-19?**

Las investigaciones indican que los niños y los adolescentes tienen las mismas probabilidades de infectarse que cualquier otro grupo de edad y pueden propagar la enfermedad.

Las pruebas hasta la fecha sugieren que los niños y los adultos jóvenes tienen menos probabilidades de desarrollar una enfermedad grave, pero con todo se pueden dar casos graves en estos grupos de edad.

Los niños y los adultos deben seguir las mismas pautas de cuarentena y aislamiento si existe el riesgo de que hayan estado expuestos o si presentan síntomas. Es particularmente importante que los niños eviten el contacto con personas mayores y con otras personas que corran el riesgo de contraer una enfermedad más grave.

- **¿Cuáles son los síntomas?**

Según los CDC, los pacientes con COVID-19 han indicado síntomas parecidos a los de otras enfermedades respiratorias, como:

- ✓ Fiebre leve a severa
- ✓ Tos
- ✓ Dificultad respiratoria

“Los escalofríos, el dolor muscular, el dolor de garganta y la pérdida reciente de sabor u olfato también pueden ser síntomas de una infección por coronavirus. Se han detectado otros síntomas menos comunes como los síntomas gastrointestinales: náuseas, vómitos o diarrea”, dicen los CDC. Erupciones en la piel u otras lesiones también pueden ser señales del virus.

Los CDC les solicitan a aquellas personas con síntomas que llamen a su proveedor de cuidados de salud o a su departamento de salud local para instrucciones sobre qué hacer antes de acudir a un centro y transmitir gérmenes a otras personas. Quienes se sienten enfermos y no están seguros de sus síntomas pueden también verificar la guía interactiva de los CDC, que ofrece consejos sobre la atención médica adecuada.

Expertos de la salud advierten que una persona debe buscar atención médica inmediata si experimenta cualquiera de los siguientes síntomas: dolor o presión en el pecho, desorientación o confusión, un cierto color azul en la cara o en los labios, dificultad para respirar o respiración entrecortada. Muchos pacientes con severas complicaciones a causa del virus pueden desarrollar neumonía y requerir un ventilador para poder respirar.

Los doctores pueden ordenar a un paciente realizarse una prueba de la COVID-19 si sospecha que la persona tiene síntomas clínicos. El 18 de marzo se firmó una ley destinada a garantizar que esta prueba puede realizarse sin costo adicional.

Los CDC han publicado una serie de consejos (en inglés) a seguir si una persona se ha infectado con el virus.

- **¿Es posible contraer el coronavirus a partir de una persona asintomática?**

Según los últimos informes, es probable que se den contagios por coronavirus a partir de personas infectadas que aún no hayan presentado síntomas. Sin embargo, sobre la base de los datos disponibles, sabemos que las personas con síntomas de la enfermedad son las principales responsables de su propagación.

- **¿Cuánto tiempo transcurre entre la exposición a la COVID-19 y el desarrollo de síntomas?**

El tiempo que transcurre entre la exposición a la COVID-19 y el momento en que comienzan los síntomas suele ser de alrededor de cinco o seis días, pero puede variar entre 1 y 14 días.

- **¿Es peligrosa la infección por coronavirus SARS-CoV-2?**

Al igual que otras infecciones del aparato respiratorio, la infección por coronavirus puede causar síntomas leves como dolor de garganta, tos y fiebre. Sin embargo, en algunas personas puede adoptar un curso grave y originar neumonía y problemas de respiración. La infección por coronavirus (COVID-19) puede ser mortal.

No se sabe exactamente cuántas personas han sido infectadas con el coronavirus y no presentan ningún tipo de síntomas. Por esta razón, es muy difícil estimar la mortalidad de COVID-19.

La mortalidad entre todos los enfermos en China fue del 2,3 %. En el grupo de 70-79 años de edad fue del 8 % y de  $\geq 80$  años del 14,8 %. La mortalidad en los casos de curso crítico fue del 49 %. La mortalidad entre los enfermos infectados con SARS-CoV-2 que tenían enfermedades crónicas fue mayor: el 10,5 % en personas con enfermedades crónicas del sistema circulatorio, el 6,3 % con enfermedades crónicas del tracto respiratorio, el 6 % con hipertensión arterial y el 5,6 % con enfermedades neoplásicas.

La mortalidad en Italia es de un 7 %. El 40 % de los fallecidos por COVID-19 tenía más de 79 años.

Para comparar, en función de la estación, mueren 0,1-0,5 % de los enfermos de gripe.



- **¿Quién enferma debido al coronavirus SARS-CoV-2?**

La infección por el coronavirus SARS-CoV-2 puede afectar a personas de cualquier edad. El análisis de las incidencias indica que su curso es más severo en las personas mayores con enfermedades crónicas (Por ejemplo: asma, diabetes, enfermedades cardíacas) o inmunodeficiencias. Esto también se aplica a los casos mortales.

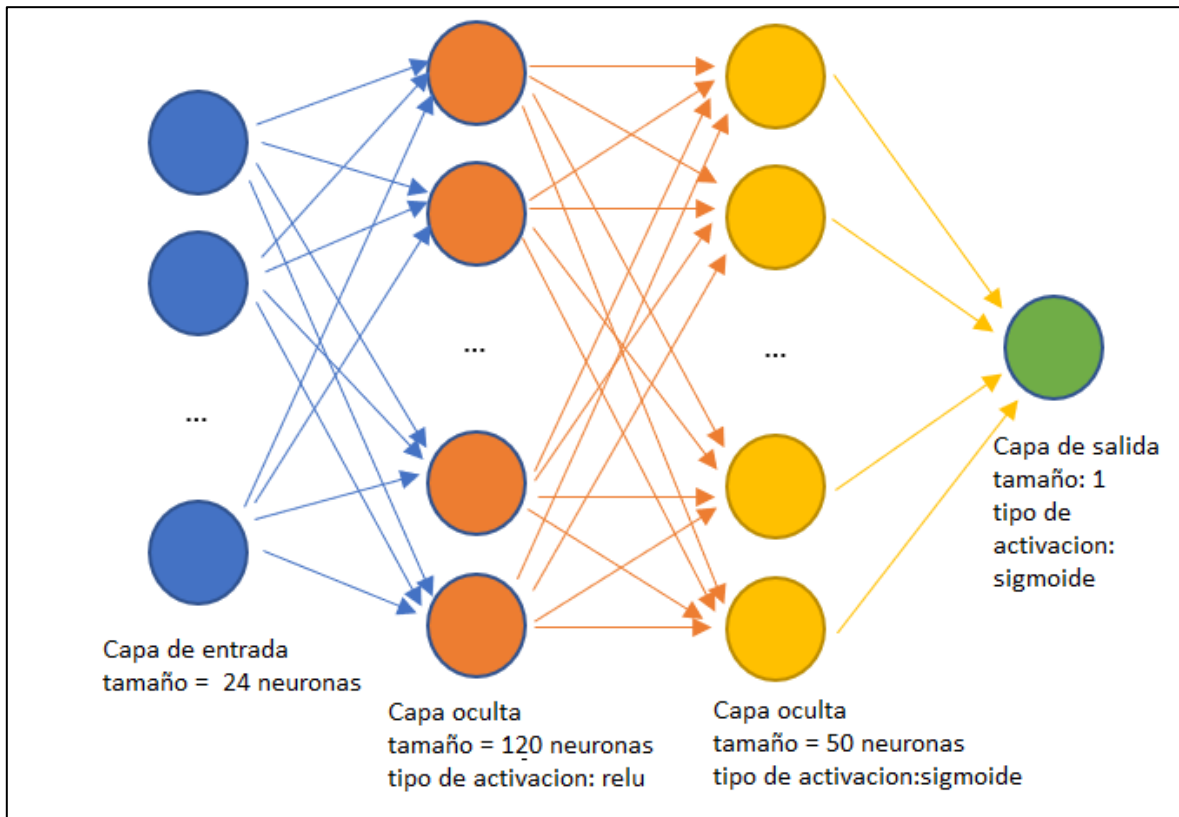
Los hombres se enferman con más frecuencia que las mujeres. Los niños se enferman muy raramente (un 1-2 % de todos los casos).

- **¿Cómo se va a medir la variable objetivo?**

La variable objetivo decidimos medirla solamente con las variables o datos vitales como las enfermedades crónicas que posee un paciente de COVID-19 y su edad.

Esto se hizo gracias a lo previamente dicho en el anterior punto, en el que supimos a qué tipo de problema nos estamos enfrentando, decidimos hacerlo de esta forma para que las probabilidades o predicciones sean lo más acercadas a lo que se dice sobre esta enfermedad de que a mayor sea el número de enfermedades crónicas o personas de bajas defensas, puedan morir debido a contraer este virus.

## EL DISEÑO DE LA RED



Para nuestra red neuronal se optó por este diseño, se realizaron varios borradores de diseños en el transcurso del desarrollo del proyecto hasta que se hizo uno con 7 capas, pero decidimos optar por uno más sencillo ya que conforme a más capas que agregáramos y cambiáramos su función de activación el modelo perdía precisión, el modelo que dejamos a continuación consta de 4 capas básicas, una capa de entrada en la que llegan los datos que están en nuestro archivo para alimentar nuestra red,

Una capa oculta de activación relu con un tamaño de 120, la justificación por la que se asignó dicho tamaño fue la gran cantidad de datos que posee nuestro documento, lo hicimos así pensando que en que nuestra red debe pensar más los datos de entrada y además usamos la activación relu porque encontramos que era mucho más precisa que la función sigmoide.

Después hicimos una nueva capa oculta de tipo sigmoide con tamaño de 50 para variar un poco.

Por último, se creó una capa de salida de una sola neurona de activación sigmoide que representa el estado final de nuestra red.

## **INFORMACIÓN DEL FRAMEWORK**

### **KERAS**



Keras es una biblioteca de Redes Neuronales de Código Abierto escrita en Python. Es capaz de ejecutarse sobre **TensorFlow, Microsoft Cognitive Toolkit o Theano.**

Está especialmente diseñada para posibilitar la experimentación en más o menos poco tiempo con redes de Aprendizaje Profundo. Sus fuertes se centran en ser amigable para el usuario, modular y extensible.

Inicialmente fue desarrollada como parte de los esfuerzos de investigación del proyecto ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System)

Su autor principal y mantenedor ha sido el **ingeniero de Google François Chollet.**

En 2017, el equipo de TensorFlow de Google decidió ofrecer soporte a Keras en la biblioteca de core de TensorFlow.

Chollet explica que Keras ha sido concebido para actuar como una interfaz en lugar de ser una framework de Machine Learning Standalone. Ofrece un conjunto de abstracciones más intuitivas y de alto nivel haciendo más sencillo el desarrollo de modelos de aprendizaje profundo independientemente del backend computacional utilizado.

Microsoft añadió un backend en CNTK a Keras también, disponible desde la CNTK v2.0

### **Características**

- Keras contiene varias implementaciones de los bloques constructivos de las redes neuronales como por ejemplo los layers, funciones objetivo, funciones de activación, optimizadores matemáticos.
- Su código está alojado en GitHub y existen foros y un canal de Slack de soporte.
- Además del soporte para las redes neuronales estándar, Keras ofrece soporte para las Redes Neuronales Convolucionales y para las Redes Neuronales Recurrentes.
- Keras permite generar modelos de Deep Learning en teléfonos inteligentes tanto sobre iOS como sobre Android, sobre una Java Virtual Machine o sobre web. También permite el uso de entrenamiento distribuido de modelos de aprendizaje profundo en clusters de Graphics Processing Units (GPU) y Tensor processing units (TPU).
- Con la llegada de TensorFlow 2.0, se puede utilizar la API de Keras para definir modelos y ejecutarlos de forma "eager". Con esto se consigue facilitar el desarrollo de modelos, posibilitar la depuración de modelos utilizando herramientas estándar, y simplificar la definición de modelos dinámicos utilizando estructuras de control.

## ¿Qué es TensorFlow?



TensorFlow es una biblioteca de software de código abierto para computación numérica, que utiliza gráficos de flujo de datos. Los nodos en las gráficas representan operaciones matemáticas, mientras que los bordes de las gráficas representan las matrices de datos multidimensionales (tensores) comunicadas entre ellos.

TensorFlow es una gran plataforma para construir y entrenar redes neuronales, que permiten detectar y descifrar patrones y correlaciones, análogos al aprendizaje y razonamiento usados por los humanos.

La arquitectura flexible de TensorFlow le permite implementar el cálculo a una o más CPU o GPU en equipos de escritorio, servidores o dispositivos móviles con una sola API. TensorFlow fue desarrollado originalmente por investigadores e ingenieros que trabajaban en el equipo de Google Brain Team, dentro del departamento de investigación de Machine Intelligence, con el propósito de llevar a cabo el aprendizaje automático y la investigación de redes neuronales profundas.

Sin embargo, el sistema es lo suficientemente general como para ser aplicable a una amplia variedad de otros dominios igualmente.

## **ALGUNOS OTROS PAQUETES O UTILERÍAS USADAS EN LA ELABORACIÓN DEL PROYECTO FUERON:**

- **Anaconda**

Anaconda es un distribución libre y abierta de los lenguajes Python y R, utilizada en ciencia de datos, y aprendizaje automático (machine learning). Esto incluye procesamiento de grandes volúmenes de información, análisis predictivo y cómputos científicos. Está orientado a simplificar el despliegue y administración de los paquetes de software.

Las diferentes versiones de los paquetes se administran mediante el sistema de gestión de paquetes conda, el cual lo hace bastante sencillo de instalar, correr, y actualizar software de ciencia de datos y aprendizaje automático como ser Scikit-team, TensorFlow y SciPy.

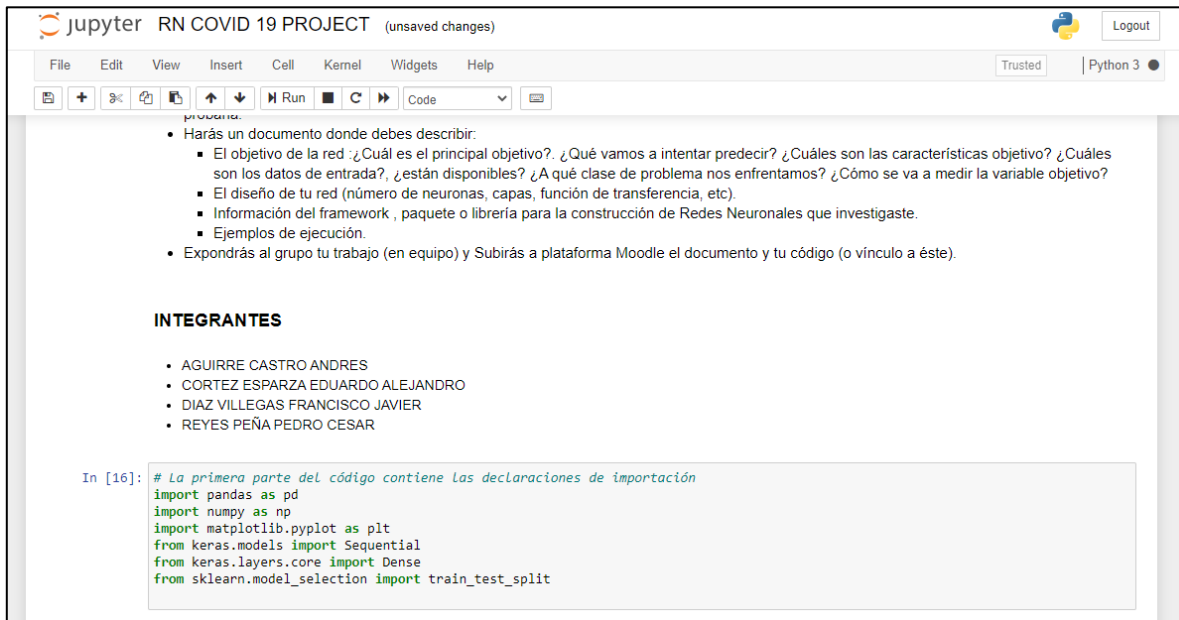
La distribución Anaconda es utilizada por 6 millones de usuarios e incluye más de 250 paquetes de ciencia de datos válidos para Windows, Linux y MacOS.

- **Jupyter Notebook**

Jupyter Notebook es una aplicación cliente-servidor lanzada en 2015 por la organización sin ánimo de lucro Proyecto Jupyter. Permite crear y compartir documentos web en formato JSON que siguen un esquema versionado y una lista ordenada de celdas de entrada y de salida. Estas celdas albergan, entre otras cosas, código, texto (en formato Markdown), fórmulas matemáticas y ecuaciones, o también contenido multimedia (Rich Media). El programa se ejecuta desde la aplicación web cliente que funciona en cualquier navegador estándar. El requisito previo es instalar y ejecutar en el sistema el servidor Jupyter Notebook. Los documentos creados en Jupyter pueden exportarse, entre otros formatos, a HTML, PDF, Markdown o Python y también pueden compartirse con otros usuarios por correo electrónico, utilizando Dropbox o GitHub o mediante el visor integrado de Jupyter Notebook.

## IMPLEMENTACION DE LA RED.

Captura de la libreta de jupyter con el programa en funcionamiento



### Código fuente

# La primera parte del código contiene las declaraciones de importación

```
import pandas as pd
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
from keras.models import Sequential
```

```
from keras.layers.core import Dense
```

```
from sklearn.model_selection import train_test_split
```

```
TARGET_VARIABLE = "MUERTO"
```

```
TRAIN_TEST_SPLIT=0.75
```

```
HIDDEN_LAYER_SIZE=120
```

```
raw_data = pd.read_csv("COVID19DATAMEXICO.csv")
```

```
mask = np.random.rand(len(raw_data)) < TRAIN_TEST_SPLIT
```

```
tr_dataset = raw_data[mask]
```

```
te_dataset = raw_data[mask]
```

```
tr_data = np.array(raw_data.drop(TARGET_VARIABLE, axis=1))
```

```

tr_labels = np.array(raw_data[[TARGET_VARIABLE]])
te_data = np.array(te_dataset.drop(TARGET_VARIABLE, axis=1))
te_labels = np.array(te_dataset[[TARGET_VARIABLE]])
ffnn = Sequential()
## Agregamos una capa de activacion relu y la capa de entrada
ffnn.add(Dense(HIDDEN_LAYER_SIZE, input_shape=(24,), activation="relu"))
## Agregamos otra capa de activacion sigmoide
ffnn.add(Dense(50, activation="sigmoid"))
## Agregamos una capa de activacion relu y la capa de entrada
ffnn.add(Dense(HIDDEN_LAYER_SIZE, input_shape=(24,), activation="relu"))
## Agregamos otra capa de activacion sigmoide
ffnn.add(Dense(50, activation="sigmoid"))

# La Sigüente línea tomará el tamaño de la capa oculta de la capa anterior (Keras
lo hace automáticamente) y
# creará una capa de salida con una neurona.

ffnn.add(Dense(1, activation="sigmoid"))
# En seguida se compila el modelo, lo que significa que ensamblará todas las
demás cosas que Python necesita
# a partir de lo que hemos especificado
# Al compilar el modelo se mandan llamar las librerías en el backend, en nuestro
caso Tensorflow
# loss --> función de pérdida o loss function, que es utilizada para evaluar los
pesos.
# optimizer --> define el optimizador para los pesos
# metrics--> reporta la precisión de la clasificación en este caso se usa accuracy

ffnn.compile(loss="mean_squared_error", optimizer="sgd", metrics=["accuracy"])

```



```
# entrena la red neuronal con los datos--> tr_data, indicado las etiquetas -->
tr_labels, durante 50 épocas,
# tomando dos muestras en lote.
# verbose = 1 significa que imprimirá la precisión y la pérdida después
# de cada época de entrenamiento.
# batch_size --> número de instances que son evaluadas antes de que los pesos
sean actualizados en la red neuronal.
```

```
hist = ffnn.fit(tr_data, tr_labels, epochs=25, batch_size=1, verbose=1)
## Desplegara una grafica sobre la funcion de perdida
plt.plot(hist.history['loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Val'], loc='upper right')
plt.show()
# evalua el modelo con los datos de prueba
metrics = ffnn.evaluate(te_data, te_labels, verbose=1)
# se imprimen metricas del modelo, en este caso se imprime la certeza del modelo
print("%s: %.2f%%" % (ffnn.metrics_names[1], metrics[1]*100))
new_data = np.array(pd.read_csv("new_data.csv"))
results = ffnn.predict(new_data)
print(results)
```

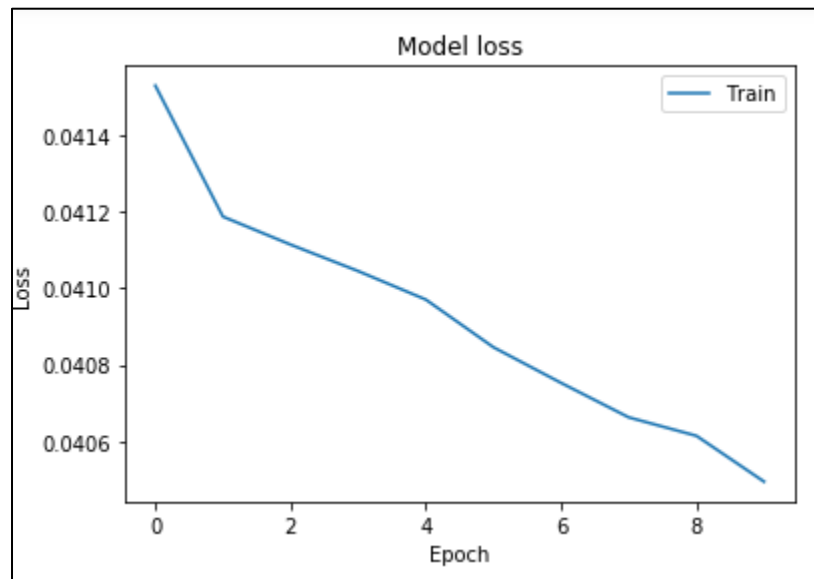
## EJEMPLOS DE EJECUCIÓN.

Realizamos diferentes entrenamientos con menos o con más épocas, para ver cómo afecta al entrenamiento de nuestra red, luego que se obtuvo el modelo con mejor entrenamiento y mayor número de épocas, decidimos guardarlo en un archivo para poder correrlo y realizar varias predicciones con nuevos individuos a predecir su porcentaje de riesgo de muerte.

### Ejecución #1

Entrenamiento de la red neuronal de 10 épocas

```
Epoch 1/10
254794/254794 [=====] - 253s 994us/step - loss: 0.0415 - accuracy: 0.9485
Epoch 2/10
254794/254794 [=====] - 242s 949us/step - loss: 0.0412 - accuracy: 0.9485
Epoch 3/10
254794/254794 [=====] - 244s 959us/step - loss: 0.0411 - accuracy: 0.9485
Epoch 4/10
254794/254794 [=====] - 242s 949us/step - loss: 0.0410 - accuracy: 0.9485
Epoch 5/10
254794/254794 [=====] - 248s 973us/step - loss: 0.0410 - accuracy: 0.9485
Epoch 6/10
254794/254794 [=====] - 259s 1ms/step - loss: 0.0408 - accuracy: 0.9484
Epoch 7/10
254794/254794 [=====] - 253s 992us/step - loss: 0.0408 - accuracy: 0.9485
Epoch 8/10
254794/254794 [=====] - 270s 1ms/step - loss: 0.0407 - accuracy: 0.9485
Epoch 9/10
254794/254794 [=====] - 253s 994us/step - loss: 0.0406 - accuracy: 0.9485
Epoch 10/10
254794/254794 [=====] - 256s 1ms/step - loss: 0.0405 - accuracy: 0.9486
```



	ORIGEN	EDAD	SECTOR	ENTIDAD_UM	SEXO	ENTIDAD_NAC	ENTIDAD_RES	MUNICIPIO_RES	TIPO_PACIENTE	INTUBADO	...	DIABETES	EPOC
0	2	60	3	2	1	2	2	2	2	1	...	1	1
1	2	82	2	12	2	12	12	1	2	2	...	2	1
2	2	70	2	14	1	14	14	120	1	97	...	1	1
3	2	52	4	27	1	27	27	4	1	97	...	1	1
4	2	45	4	27	1	27	27	4	1	97	...	1	2
5	2	25	4	21	1	21	21	114	1	97	...	2	1

ASMA	INMUSUPR	HIPERTENSION	OTRA_COM	CARDIOVASCULAR	OBESIDAD	RENAL_CRONICA	TABAQUISMO
1	1	2	1	1	1	1	1
2	2	1	2	1	2	1	1
1	1	1	2	2	2	2	2
1	1	1	2	2	1	2	2
2	2	2	2	2	2	2	2
1	2	1	1	1	1	2	2

### Datos predichos:

```
[[0.27340662]
 [0.29580906]
 [0.02746334]
 [0.00710292]
 [0.00461532]
 [0.00259812]]
```

### Comentarios:

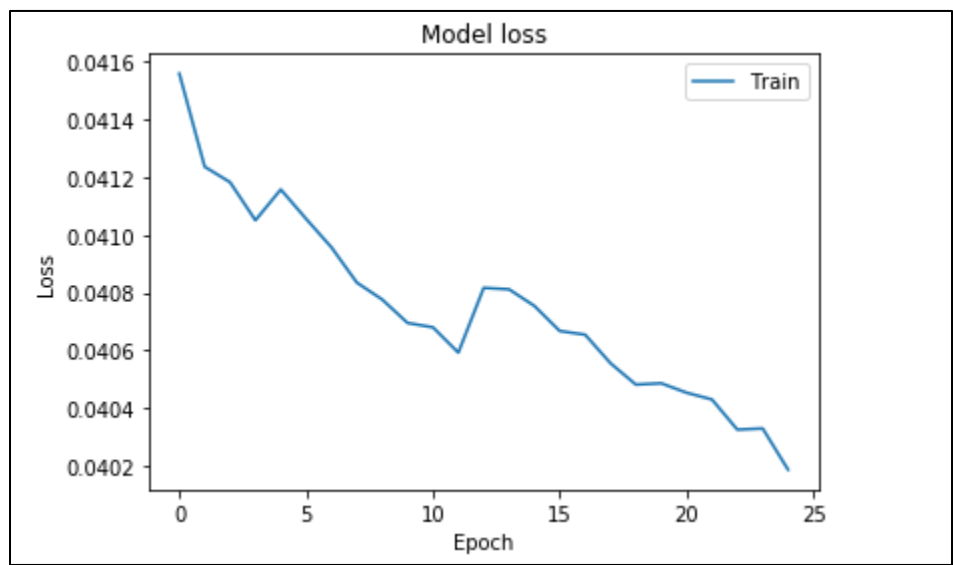
Podemos apreciar que, pese a que el modelo no se entrenó lo suficiente si llego a darnos resultados acercados a los que esperábamos, lo comprobamos con los primeros dos pacientes al primero de ellos le dimos 60 años y le adjuntamos que padece varios tipos de enfermedades en su organismo, y al segundo fue un caso similar, pero le dimos 82 años de edad, podemos ver qué al primero le dio el 27% de riesgo de muerte por contraer el COVID-19 y al segundo por su edad un 29% de riesgo siendo el más alto de la lista .

## Ejecución #2

### Entrenamiento del modelo de red neuronal de 25 épocas

```
Epoch 1/25
254794/254794 [=====] - 252s 991us/step - loss: 0.0416 - accuracy: 0.9485
Epoch 2/25
254794/254794 [=====] - 244s 958us/step - loss: 0.0412 - accuracy: 0.9486
Epoch 3/25
254794/254794 [=====] - 245s 962us/step - loss: 0.0412 - accuracy: 0.9486
Epoch 4/25
254794/254794 [=====] - 243s 956us/step - loss: 0.0411 - accuracy: 0.9485
Epoch 5/25
254794/254794 [=====] - 224s 880us/step - loss: 0.0412 - accuracy: 0.9485
Epoch 6/25
254794/254794 [=====] - 222s 871us/step - loss: 0.0411 - accuracy: 0.9484
Epoch 7/25
254794/254794 [=====] - 233s 915us/step - loss: 0.0410 - accuracy: 0.9485
Epoch 8/25
254794/254794 [=====] - 243s 955us/step - loss: 0.0408 - accuracy: 0.9485
Epoch 9/25
254794/254794 [=====] - 239s 936us/step - loss: 0.0408 - accuracy: 0.9485
Epoch 10/25
254794/254794 [=====] - 230s 902us/step - loss: 0.0407 - accuracy: 0.9486
Epoch 11/25
254794/254794 [=====] - 227s 892us/step - loss: 0.0407 - accuracy: 0.9486
Epoch 12/25
254794/254794 [=====] - 231s 905us/step - loss: 0.0406 - accuracy: 0.9486
Epoch 13/25
254794/254794 [=====] - 215s 845us/step - loss: 0.0408 - accuracy: 0.9484
```

```
Epoch 14/25
254794/254794 [=====] - 219s 859us/step - loss: 0.0408 - accuracy: 0.9485
Epoch 15/25
254794/254794 [=====] - 217s 850us/step - loss: 0.0408 - accuracy: 0.9485
Epoch 16/25
254794/254794 [=====] - 217s 850us/step - loss: 0.0407 - accuracy: 0.9485
Epoch 17/25
254794/254794 [=====] - 220s 862us/step - loss: 0.0407 - accuracy: 0.9485
Epoch 18/25
254794/254794 [=====] - 215s 844us/step - loss: 0.0406 - accuracy: 0.9485
Epoch 19/25
254794/254794 [=====] - 215s 842us/step - loss: 0.0405 - accuracy: 0.9485
Epoch 20/25
254794/254794 [=====] - 227s 890us/step - loss: 0.0405 - accuracy: 0.9485
Epoch 21/25
254794/254794 [=====] - 225s 881us/step - loss: 0.0405 - accuracy: 0.9485
Epoch 22/25
254794/254794 [=====] - 226s 886us/step - loss: 0.0404 - accuracy: 0.9485
Epoch 23/25
254794/254794 [=====] - 225s 882us/step - loss: 0.0403 - accuracy: 0.9485
Epoch 24/25
254794/254794 [=====] - 225s 884us/step - loss: 0.0403 - accuracy: 0.9485
Epoch 25/25
254794/254794 [=====] - 224s 881us/step - loss: 0.0402 - accuracy: 0.9486
```



	ORIGEN	EDAD	SECTOR	ENTIDAD_UM	SEXO	ENTIDAD_NAC	ENTIDAD_RES	MUNICIPIO_RES	TIPO_PACIENTE	INTUBADO	...	DIABETES	EPOC
0	2	60	3	2	1	2	2	2	2	2	1 ...	1	1
1	2	82	2	12	2	12	12	1	2	2	2 ...	2	1
2	2	70	2	14	1	14	14	120	1	97	...	1	1
3	2	52	4	27	1	27	27	4	1	97	...	1	1
4	2	45	4	27	1	27	27	4	1	97	...	1	2
5	2	25	4	21	1	21	21	114	1	97	...	2	1

ASMA	INMUSUPR	HIPERTENSION	OTRA_COM	CARDIOVASCULAR	OBESIDAD	RENAL_CRONICA	TABAQUISMO
1	1	2	1	1	1	1	1
2	2	1	2	1	2	1	1
1	1	1	2	2	2	2	2
1	1	1	2	2	1	2	2
2	2	2	2	2	2	2	2
1	2	1	1	1	1	2	2

### Datos predichos

```

[[0.2987704 ]
 [0.26242033]
 [0.01751943]
 [0.00445614]
 [0.00309843]
 [0.00305216]]

```

### **Comentarios:**

Podemos apreciar que, pese a que el modelo no se entrenó lo suficiente con 25 épocas, si llego a darnos resultados acercados a los que esperábamos y mejoro con respecto al anterior modelo, lo comprobamos con los primeros dos pacientes al primero de ellos le dimos 60 años y le adjuntamos que padece varios tipos de enfermedades en su organismo, y al segundo fue un caso similar, pero le dimos 82 años de edad, podemos ver qué al primero le dio el 29% de riesgo de muerte por contraer el COVID-19 y al segundo por su edad un 26% de riesgo siendo el más alto de la lista .

### **Ejecución #3**

Entrenamiento del modelo de la red neuronal de 50 épocas

```
Epoch 1/50
254794/254794 [=====] - 547s 2ms/step - loss: 0.0423 - accuracy: 0.9485
Epoch 2/50
254794/254794 [=====] - 534s 2ms/step - loss: 0.0410 - accuracy: 0.9485
Epoch 3/50
254794/254794 [=====] - 347s 1ms/step - loss: 0.0408 - accuracy: 0.9485
Epoch 4/50
254794/254794 [=====] - 257s 1ms/step - loss: 0.0407 - accuracy: 0.9485
Epoch 5/50
254794/254794 [=====] - 257s 1ms/step - loss: 0.0407 - accuracy: 0.9485
Epoch 6/50
254794/254794 [=====] - 251s 985us/step - loss: 0.0406 - accuracy: 0.9485
Epoch 7/50
254794/254794 [=====] - 256s 1ms/step - loss: 0.0406 - accuracy: 0.9485
Epoch 8/50
254794/254794 [=====] - 266s 1ms/step - loss: 0.0406 - accuracy: 0.9485
Epoch 9/50
254794/254794 [=====] - 256s 1ms/step - loss: 0.0406 - accuracy: 0.9485
Epoch 10/50
254794/254794 [=====] - 285s 1ms/step - loss: 0.0405 - accuracy: 0.9485
Epoch 11/50
254794/254794 [=====] - 250s 982us/step - loss: 0.0404 - accuracy: 0.9485
Epoch 12/50
```

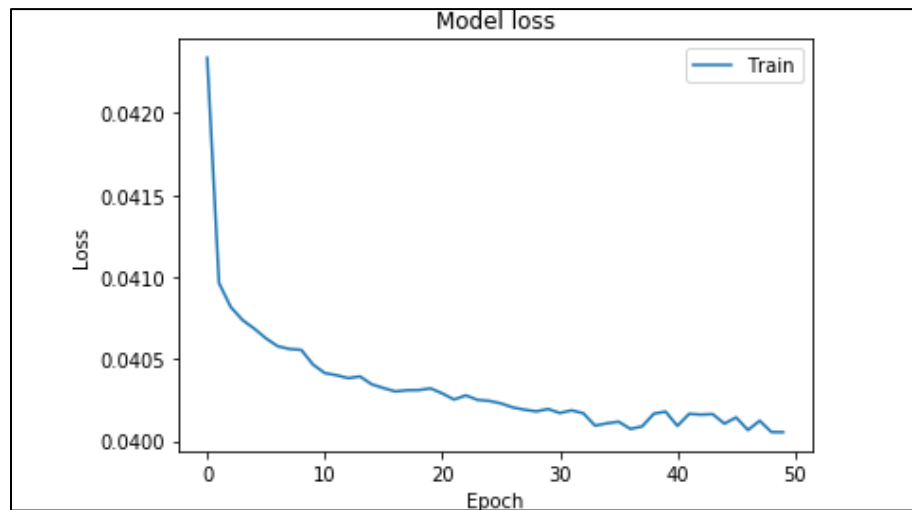
```
Epoch 13/50
254794/254794 [=====] - 249s 976us/step - loss: 0.0404 - accuracy: 0.9485
Epoch 14/50
254794/254794 [=====] - 249s 977us/step - loss: 0.0404 - accuracy: 0.9485
Epoch 15/50
254794/254794 [=====] - 252s 991us/step - loss: 0.0403 - accuracy: 0.9485
Epoch 16/50
254794/254794 [=====] - 251s 983us/step - loss: 0.0403 - accuracy: 0.9485
Epoch 17/50
254794/254794 [=====] - 250s 981us/step - loss: 0.0403 - accuracy: 0.9485
Epoch 18/50
254794/254794 [=====] - 250s 982us/step - loss: 0.0403 - accuracy: 0.9485
Epoch 19/50
254794/254794 [=====] - 252s 989us/step - loss: 0.0403 - accuracy: 0.9485
Epoch 20/50
254794/254794 [=====] - 250s 981us/step - loss: 0.0403 - accuracy: 0.9485
Epoch 21/50
254794/254794 [=====] - 254s 995us/step - loss: 0.0403 - accuracy: 0.9485
Epoch 22/50
254794/254794 [=====] - 251s 985us/step - loss: 0.0403 - accuracy: 0.9485
Epoch 23/50
254794/254794 [=====] - 253s 993us/step - loss: 0.0403 - accuracy: 0.9485
Epoch 24/50
254794/254794 [=====] - 253s 994us/step - loss: 0.0403 - accuracy: 0.9485
Epoch 25/50
254794/254794 [=====] - 251s 986us/step - loss: 0.0402 - accuracy: 0.9485
```

```
Epoch 26/50
254794/254794 [=====] - 251s 984us/step - loss: 0.0402 - accuracy: 0.9485
Epoch 27/50
254794/254794 [=====] - 251s 985us/step - loss: 0.0402 - accuracy: 0.9485
Epoch 28/50
254794/254794 [=====] - 251s 986us/step - loss: 0.0402 - accuracy: 0.9485
Epoch 29/50
254794/254794 [=====] - 255s 999us/step - loss: 0.0402 - accuracy: 0.9485
Epoch 30/50
254794/254794 [=====] - 252s 987us/step - loss: 0.0402 - accuracy: 0.9485
Epoch 31/50
254794/254794 [=====] - 251s 987us/step - loss: 0.0402 - accuracy: 0.9485
Epoch 32/50
254794/254794 [=====] - 252s 987us/step - loss: 0.0402 - accuracy: 0.9485
Epoch 33/50
254794/254794 [=====] - 258s 1ms/step - loss: 0.0402 - accuracy: 0.9485
Epoch 34/50
254794/254794 [=====] - 254s 996us/step - loss: 0.0401 - accuracy: 0.9485
Epoch 35/50
254794/254794 [=====] - 252s 988us/step - loss: 0.0401 - accuracy: 0.9485
Epoch 36/50
254794/254794 [=====] - 251s 984us/step - loss: 0.0401 - accuracy: 0.9485
Epoch 37/50
254794/254794 [=====] - 250s 980us/step - loss: 0.0401 - accuracy: 0.9485
Epoch 38/50
254794/254794 [=====] - 252s 990us/step - loss: 0.0401 - accuracy: 0.9485
Epoch 39/50
254794/254794 [=====] - 250s 980us/step - loss: 0.0402 - accuracy: 0.9485
Epoch 40/50
254794/254794 [=====] - 245s 963us/step - loss: 0.0402 - accuracy: 0.9485
```

```

Epoch 41/50
254794/254794 [=====] - 245s 962us/step - loss: 0.0401 - accuracy: 0.9485
Epoch 42/50
254794/254794 [=====] - 478s 2ms/step - loss: 0.0402 - accuracy: 0.9485
Epoch 43/50
254794/254794 [=====] - 560s 2ms/step - loss: 0.0402 - accuracy: 0.9485
Epoch 44/50
254794/254794 [=====] - 539s 2ms/step - loss: 0.0402 - accuracy: 0.9485
Epoch 45/50
254794/254794 [=====] - 546s 2ms/step - loss: 0.0401 - accuracy: 0.9485
Epoch 46/50
254794/254794 [=====] - 563s 2ms/step - loss: 0.0401 - accuracy: 0.9485
Epoch 47/50
254794/254794 [=====] - 567s 2ms/step - loss: 0.0401 - accuracy: 0.9485
Epoch 48/50
254794/254794 [=====] - 555s 2ms/step - loss: 0.0401 - accuracy: 0.9485
Epoch 49/50
254794/254794 [=====] - 555s 2ms/step - loss: 0.0401 - accuracy: 0.9485
Epoch 50/50
254794/254794 [=====] - 343s 1ms/step - loss: 0.0401 - accuracy: 0.9485

```



	ORIGEN	EDAD	SECTOR	ENTIDAD_UM	SEXO	ENTIDAD_NAC	ENTIDAD_RES	MUNICIPIO_RES	TIPO_PACIENTE	INTUBADO	...	DIABETES	EPOC
0	2	60	3	2	1	2	2	2	2	2	1 ...	1	1
1	2	82	2	12	2	12	12	1	2	2	2 ...	2	1
2	2	70	2	14	1	14	14	120	1	97	...	1	1
3	2	52	4	27	1	27	27	4	1	97	...	1	1
4	2	45	4	27	1	27	27	4	1	97	...	1	2
5	2	25	4	21	1	21	21	114	1	97	...	2	1

ASMA	INMUSUPR	HIPERTENSION	OTRA_COM	CARDIOVASCULAR	OBESIDAD	RENAL_CRONICA	TABAQUISMO
1	1	2	1	1	1	1	1
2	2	1	2	1	2	1	1
1	1	1	2	2	2	2	2
1	1	1	2	2	1	2	2
2	2	2	2	2	2	2	2
1	2	1	1	1	1	2	2



### **Datos predichos**

```
[[0.36975297]
 [0.35064846]
 [0.0156737 ]
 [0.0156737 ]
 [0.00896008]
 [0.00407114]]
```

### **Comentarios:**

Podemos apreciar que, pese a que el modelo no se entrenó lo suficiente con 50 épocas, pero mejoro con respecto a los predecesores entrenamientos si llego a darnos resultados acercados a los que esperábamos, lo comprobamos con los primeros dos pacientes al primero de ellos le dimos 60 años y le adjuntamos que padece varios tipos de enfermedades en su organismo, y al segundo fue un caso similar, pero le dimos 82 años de edad, podemos ver qué al primero le dio el 36% de riesgo de muerte por contraer el COVID-19 y al segundo por su edad un 35% de riesgo siendo el más alto de la lista.

### **Ejecución #4**

Entrenamiento de la red neuronal de 150 épocas

```
Epoch 1/150
254794/254794 [=====] - 308s 1ms/step - loss: 0.0418 - accuracy: 0.9485
Epoch 2/150
254794/254794 [=====] - 317s 1ms/step - loss: 0.0414 - accuracy: 0.9485
Epoch 3/150
254794/254794 [=====] - 330s 1ms/step - loss: 0.0412 - accuracy: 0.9485
Epoch 4/150
254794/254794 [=====] - 342s 1ms/step - loss: 0.0411 - accuracy: 0.9485
Epoch 5/150
254794/254794 [=====] - 325s 1ms/step - loss: 0.0410 - accuracy: 0.9485
Epoch 6/150
254794/254794 [=====] - 302s 1ms/step - loss: 0.0410 - accuracy: 0.9486
Epoch 7/150
254794/254794 [=====] - 286s 1ms/step - loss: 0.0408 - accuracy: 0.9485
Epoch 8/150
254794/254794 [=====] - 258s 1ms/step - loss: 0.0408 - accuracy: 0.9485
Epoch 9/150
254794/254794 [=====] - 222s 872us/step - loss: 0.0408 - accuracy: 0.9485
```

```
Epoch 10/150
254794/254794 [=====] - 213s 837us/step - loss: 0.0407 - accuracy: 0.9485
Epoch 11/150
254794/254794 [=====] - 214s 838us/step - loss: 0.0407 - accuracy: 0.9485
Epoch 12/150
254794/254794 [=====] - 218s 855us/step - loss: 0.0406 - accuracy: 0.9485
Epoch 13/150
254794/254794 [=====] - 222s 871us/step - loss: 0.0405 - accuracy: 0.9485
Epoch 14/150
254794/254794 [=====] - 212s 834us/step - loss: 0.0407 - accuracy: 0.9484
Epoch 15/150
254794/254794 [=====] - 222s 872us/step - loss: 0.0405 - accuracy: 0.9485
Epoch 16/150
254794/254794 [=====] - 213s 837us/step - loss: 0.0405 - accuracy: 0.9486
Epoch 17/150
254794/254794 [=====] - 209s 821us/step - loss: 0.0405 - accuracy: 0.9486
Epoch 18/150
254794/254794 [=====] - 222s 871us/step - loss: 0.0405 - accuracy: 0.9485
```

```
Epoch 19/150
254794/254794 [=====] - 218s 856us/step - loss: 0.0403 - accuracy: 0.9485
Epoch 20/150
254794/254794 [=====] - 217s 850us/step - loss: 0.0404 - accuracy: 0.9486
Epoch 21/150
254794/254794 [=====] - 217s 852us/step - loss: 0.0403 - accuracy: 0.9486
Epoch 22/150
254794/254794 [=====] - 222s 870us/step - loss: 0.0403 - accuracy: 0.9485
Epoch 23/150
254794/254794 [=====] - 223s 875us/step - loss: 0.0403 - accuracy: 0.9486
Epoch 24/150
254794/254794 [=====] - 220s 862us/step - loss: 0.0403 - accuracy: 0.9485
Epoch 25/150
254794/254794 [=====] - 232s 910us/step - loss: 0.0402 - accuracy: 0.9485
Epoch 26/150
254794/254794 [=====] - 234s 919us/step - loss: 0.0402 - accuracy: 0.9486
Epoch 27/150
254794/254794 [=====] - 220s 865us/step - loss: 0.0401 - accuracy: 0.9486
```

```
Epoch 28/150
254794/254794 [=====] - 233s 916us/step - loss: 0.0401 - accuracy: 0.9485
Epoch 29/150
254794/254794 [=====] - 237s 929us/step - loss: 0.0400 - accuracy: 0.9485
Epoch 30/150
254794/254794 [=====] - 227s 890us/step - loss: 0.0401 - accuracy: 0.9486
Epoch 31/150
254794/254794 [=====] - 227s 893us/step - loss: 0.0401 - accuracy: 0.9486
Epoch 32/150
254794/254794 [=====] - 238s 935us/step - loss: 0.0400 - accuracy: 0.9485
Epoch 33/150
254794/254794 [=====] - 232s 911us/step - loss: 0.0399 - accuracy: 0.9486
Epoch 34/150
254794/254794 [=====] - 221s 866us/step - loss: 0.0400 - accuracy: 0.9485
Epoch 35/150
254794/254794 [=====] - 219s 860us/step - loss: 0.0400 - accuracy: 0.9485
```

```
Epoch 36/150
254794/254794 [=====] - 219s 858us/step - loss: 0.0401 - accuracy: 0.9486
Epoch 37/150
254794/254794 [=====] - 228s 893us/step - loss: 0.0400 - accuracy: 0.9486
Epoch 38/150
254794/254794 [=====] - 223s 875us/step - loss: 0.0399 - accuracy: 0.9487
Epoch 39/150
254794/254794 [=====] - 223s 874us/step - loss: 0.0400 - accuracy: 0.9487
Epoch 40/150
254794/254794 [=====] - 250s 980us/step - loss: 0.0398 - accuracy: 0.9486
Epoch 41/150
254794/254794 [=====] - 302s 1ms/step - loss: 0.0397 - accuracy: 0.9486
Epoch 42/150
254794/254794 [=====] - 307s 1ms/step - loss: 0.0397 - accuracy: 0.9486
Epoch 43/150
254794/254794 [=====] - 306s 1ms/step - loss: 0.0397 - accuracy: 0.9486
Epoch 44/150
254794/254794 [=====] - 306s 1ms/step - loss: 0.0396 - accuracy: 0.9486
```

```
Epoch 45/150
254794/254794 [=====] - 305s 1ms/step - loss: 0.0396 - accuracy: 0.9487
Epoch 46/150
254794/254794 [=====] - 309s 1ms/step - loss: 0.0395 - accuracy: 0.9488
Epoch 47/150
254794/254794 [=====] - 305s 1ms/step - loss: 0.0395 - accuracy: 0.9488
Epoch 48/150
254794/254794 [=====] - 306s 1ms/step - loss: 0.0395 - accuracy: 0.9486
Epoch 49/150
254794/254794 [=====] - 306s 1ms/step - loss: 0.0395 - accuracy: 0.9487
Epoch 50/150
254794/254794 [=====] - 308s 1ms/step - loss: 0.0395 - accuracy: 0.9488
Epoch 51/150
254794/254794 [=====] - 306s 1ms/step - loss: 0.0395 - accuracy: 0.9487
Epoch 52/150
254794/254794 [=====] - 306s 1ms/step - loss: 0.0394 - accuracy: 0.9488
Epoch 53/150
254794/254794 [=====] - 291s 1ms/step - loss: 0.0394 - accuracy: 0.9488
```

```
Epoch 54/150
254794/254794 [=====] - 294s 1ms/step - loss: 0.0394 - accuracy: 0.9487
Epoch 55/150
254794/254794 [=====] - 293s 1ms/step - loss: 0.0393 - accuracy: 0.9492
Epoch 56/150
254794/254794 [=====] - 292s 1ms/step - loss: 0.0392 - accuracy: 0.9489
Epoch 57/150
254794/254794 [=====] - 292s 1ms/step - loss: 0.0393 - accuracy: 0.9489
Epoch 58/150
254794/254794 [=====] - 295s 1ms/step - loss: 0.0393 - accuracy: 0.9491
Epoch 59/150
254794/254794 [=====] - 294s 1ms/step - loss: 0.0393 - accuracy: 0.9490
Epoch 60/150
254794/254794 [=====] - 293s 1ms/step - loss: 0.0395 - accuracy: 0.9486
Epoch 61/150
254794/254794 [=====] - 221s 867us/step - loss: 0.0392 - accuracy: 0.9490
Epoch 62/150
254794/254794 [=====] - 224s 878us/step - loss: 0.0391 - accuracy: 0.9491
```

Epoch 63/150  
254794/254794 [=====] - 221s 866us/step - loss: 0.0391 - accuracy: 0.9493  
Epoch 64/150  
254794/254794 [=====] - 221s 866us/step - loss: 0.0393 - accuracy: 0.9491  
Epoch 65/150  
254794/254794 [=====] - 221s 869us/step - loss: 0.0391 - accuracy: 0.9490  
Epoch 66/150  
254794/254794 [=====] - 220s 864us/step - loss: 0.0390 - accuracy: 0.9494  
Epoch 67/150  
254794/254794 [=====] - 221s 866us/step - loss: 0.0391 - accuracy: 0.9491  
Epoch 68/150  
254794/254794 [=====] - 223s 875us/step - loss: 0.0389 - accuracy: 0.9492  
Epoch 69/150  
254794/254794 [=====] - 221s 866us/step - loss: 0.0389 - accuracy: 0.9493  
Epoch 70/150  
254794/254794 [=====] - 222s 872us/step - loss: 0.0390 - accuracy: 0.9492  
Epoch 71/150  
254794/254794 [=====] - 220s 864us/step - loss: 0.0390 - accuracy: 0.9494

Epoch 72/150  
254794/254794 [=====] - 220s 863us/step - loss: 0.0388 - accuracy: 0.9494  
Epoch 73/150  
254794/254794 [=====] - 224s 878us/step - loss: 0.0390 - accuracy: 0.9494  
Epoch 74/150  
254794/254794 [=====] - 212s 834us/step - loss: 0.0389 - accuracy: 0.9494  
Epoch 75/150  
254794/254794 [=====] - 211s 827us/step - loss: 0.0389 - accuracy: 0.9492  
Epoch 76/150  
254794/254794 [=====] - 211s 828us/step - loss: 0.0388 - accuracy: 0.9496  
Epoch 77/150  
254794/254794 [=====] - 212s 833us/step - loss: 0.0388 - accuracy: 0.9495  
Epoch 78/150  
254794/254794 [=====] - 212s 831us/step - loss: 0.0387 - accuracy: 0.9497  
Epoch 79/150  
254794/254794 [=====] - 213s 835us/step - loss: 0.0387 - accuracy: 0.9499  
Epoch 80/150  
254794/254794 [=====] - 214s 839us/step - loss: 0.0388 - accuracy: 0.9497

Epoch 81/150  
254794/254794 [=====] - 217s 851us/step - loss: 0.0388 - accuracy: 0.9495  
Epoch 82/150  
254794/254794 [=====] - 212s 831us/step - loss: 0.0387 - accuracy: 0.9497  
Epoch 83/150  
254794/254794 [=====] - 212s 833us/step - loss: 0.0387 - accuracy: 0.9495  
Epoch 84/150  
254794/254794 [=====] - 213s 834us/step - loss: 0.0385 - accuracy: 0.9500  
Epoch 85/150  
254794/254794 [=====] - 213s 837us/step - loss: 0.0385 - accuracy: 0.9499  
Epoch 86/150  
254794/254794 [=====] - 213s 835us/step - loss: 0.0387 - accuracy: 0.9499  
Epoch 87/150  
254794/254794 [=====] - 213s 836us/step - loss: 0.0385 - accuracy: 0.9498  
Epoch 88/150  
254794/254794 [=====] - 212s 832us/step - loss: 0.0385 - accuracy: 0.9499  
Epoch 89/150  
254794/254794 [=====] - 214s 838us/step - loss: 0.0385 - accuracy: 0.9500

Epoch 90/150  
254794/254794 [=====] - 213s 837us/step - loss: 0.0386 - accuracy: 0.9499  
Epoch 91/150  
254794/254794 [=====] - 214s 838us/step - loss: 0.0386 - accuracy: 0.9498  
Epoch 92/150  
254794/254794 [=====] - 213s 837us/step - loss: 0.0384 - accuracy: 0.9502  
Epoch 93/150  
254794/254794 [=====] - 213s 835us/step - loss: 0.0385 - accuracy: 0.9500  
Epoch 94/150  
254794/254794 [=====] - 211s 829us/step - loss: 0.0385 - accuracy: 0.9499  
Epoch 95/150  
254794/254794 [=====] - 213s 835us/step - loss: 0.0383 - accuracy: 0.9502  
Epoch 96/150  
254794/254794 [=====] - 214s 841us/step - loss: 0.0384 - accuracy: 0.9503  
Epoch 97/150  
254794/254794 [=====] - 213s 835us/step - loss: 0.0384 - accuracy: 0.9501  
Epoch 98/150  
254794/254794 [=====] - 213s 836us/step - loss: 0.0384 - accuracy: 0.9502

Epoch 99/150  
254794/254794 [=====] - 213s 837us/step - loss: 0.0384 - accuracy: 0.9501  
Epoch 100/150  
254794/254794 [=====] - 218s 855us/step - loss: 0.0383 - accuracy: 0.9502  
Epoch 101/150  
254794/254794 [=====] - 223s 877us/step - loss: 0.0383 - accuracy: 0.9502  
Epoch 102/150  
254794/254794 [=====] - 220s 863us/step - loss: 0.0383 - accuracy: 0.9504  
Epoch 103/150  
254794/254794 [=====] - 219s 860us/step - loss: 0.0383 - accuracy: 0.9503  
Epoch 104/150  
254794/254794 [=====] - 221s 866us/step - loss: 0.0382 - accuracy: 0.9502  
Epoch 105/150  
254794/254794 [=====] - 218s 856us/step - loss: 0.0384 - accuracy: 0.9500  
Epoch 106/150  
254794/254794 [=====] - 219s 861us/step - loss: 0.0382 - accuracy: 0.9505

Epoch 107/150  
254794/254794 [=====] - 222s 872us/step - loss: 0.0383 - accuracy: 0.9504  
Epoch 108/150  
254794/254794 [=====] - 219s 860us/step - loss: 0.0384 - accuracy: 0.9501  
Epoch 109/150  
254794/254794 [=====] - 218s 857us/step - loss: 0.0383 - accuracy: 0.9504  
Epoch 110/150  
254794/254794 [=====] - 220s 863us/step - loss: 0.0383 - accuracy: 0.9502  
Epoch 111/150  
254794/254794 [=====] - 220s 862us/step - loss: 0.0382 - accuracy: 0.9503  
Epoch 112/150  
254794/254794 [=====] - 224s 878us/step - loss: 0.0383 - accuracy: 0.9503  
Epoch 113/150  
254794/254794 [=====] - 300s 1ms/step - loss: 0.0384 - accuracy: 0.9504  
Epoch 114/150  
254794/254794 [=====] - 301s 1ms/step - loss: 0.0383 - accuracy: 0.9505  
Epoch 115/150  
254794/254794 [=====] - 300s 1ms/step - loss: 0.0382 - accuracy: 0.9503

```
Epoch 116/150
254794/254794 [=====] - 302s 1ms/step - loss: 0.0383 - accuracy: 0.9503
Epoch 117/150
254794/254794 [=====] - 299s 1ms/step - loss: 0.0384 - accuracy: 0.9505
Epoch 118/150
254794/254794 [=====] - 300s 1ms/step - loss: 0.0382 - accuracy: 0.9504
Epoch 119/150
254794/254794 [=====] - 299s 1ms/step - loss: 0.0385 - accuracy: 0.9501
Epoch 120/150
254794/254794 [=====] - 301s 1ms/step - loss: 0.0383 - accuracy: 0.9504
Epoch 121/150
254794/254794 [=====] - 299s 1ms/step - loss: 0.0383 - accuracy: 0.9506
Epoch 122/150
254794/254794 [=====] - 299s 1ms/step - loss: 0.0384 - accuracy: 0.9500
Epoch 123/150
254794/254794 [=====] - 299s 1ms/step - loss: 0.0383 - accuracy: 0.9502
Epoch 124/150
254794/254794 [=====] - 301s 1ms/step - loss: 0.0383 - accuracy: 0.9505
```

```
Epoch 125/150
254794/254794 [=====] - 299s 1ms/step - loss: 0.0384 - accuracy: 0.9501
Epoch 126/150
254794/254794 [=====] - 299s 1ms/step - loss: 0.0383 - accuracy: 0.9502
Epoch 127/150
254794/254794 [=====] - 299s 1ms/step - loss: 0.0382 - accuracy: 0.9505
Epoch 128/150
254794/254794 [=====] - 301s 1ms/step - loss: 0.0384 - accuracy: 0.9503
Epoch 129/150
254794/254794 [=====] - 298s 1ms/step - loss: 0.0385 - accuracy: 0.9501
Epoch 130/150
254794/254794 [=====] - 309s 1ms/step - loss: 0.0384 - accuracy: 0.9504
Epoch 131/150
254794/254794 [=====] - 308s 1ms/step - loss: 0.0384 - accuracy: 0.9503
Epoch 132/150
254794/254794 [=====] - 308s 1ms/step - loss: 0.0384 - accuracy: 0.9503
Epoch 133/150
254794/254794 [=====] - 307s 1ms/step - loss: 0.0383 - accuracy: 0.9505
```

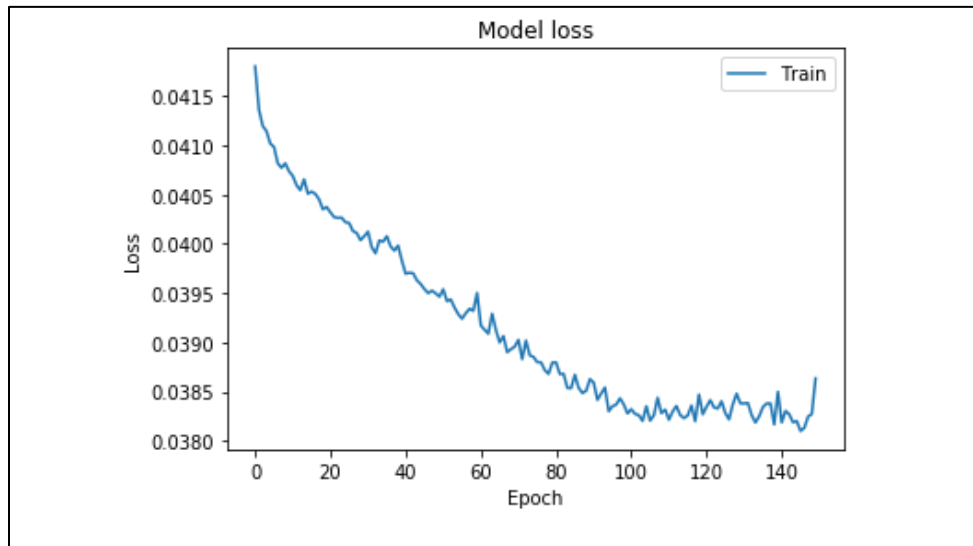
```
Epoch 134/150
254794/254794 [=====] - 306s 1ms/step - loss: 0.0382 - accuracy: 0.9506
Epoch 135/150
254794/254794 [=====] - 236s 926us/step - loss: 0.0382 - accuracy: 0.9504
Epoch 136/150
254794/254794 [=====] - 219s 859us/step - loss: 0.0383 - accuracy: 0.9504
Epoch 137/150
254794/254794 [=====] - 221s 866us/step - loss: 0.0384 - accuracy: 0.9503
Epoch 138/150
254794/254794 [=====] - 227s 891us/step - loss: 0.0384 - accuracy: 0.9503
Epoch 139/150
254794/254794 [=====] - 219s 861us/step - loss: 0.0382 - accuracy: 0.9505
Epoch 140/150
254794/254794 [=====] - 220s 862us/step - loss: 0.0385 - accuracy: 0.9501
Epoch 141/150
254794/254794 [=====] - 218s 856us/step - loss: 0.0382 - accuracy: 0.9505
Epoch 142/150
254794/254794 [=====] - 221s 868us/step - loss: 0.0383 - accuracy: 0.9503
```



```

Epoch 143/150
254794/254794 [=====] - 220s 862us/step - loss: 0.0383 - accuracy: 0.9502
Epoch 144/150
254794/254794 [=====] - 218s 856us/step - loss: 0.0382 - accuracy: 0.9504
Epoch 145/150
254794/254794 [=====] - 219s 859us/step - loss: 0.0382 - accuracy: 0.9506
Epoch 146/150
254794/254794 [=====] - 221s 868us/step - loss: 0.0381 - accuracy: 0.9508
Epoch 147/150
254794/254794 [=====] - 212s 831us/step - loss: 0.0381 - accuracy: 0.9507
Epoch 148/150
254794/254794 [=====] - 218s 854us/step - loss: 0.0382 - accuracy: 0.9506
Epoch 149/150
254794/254794 [=====] - 215s 845us/step - loss: 0.0383 - accuracy: 0.9504
Epoch 150/150
254794/254794 [=====] - 218s 858us/step - loss: 0.0386 - accuracy: 0.9500

```



A partir de este modelo ya entrenado se optó por realizarse varias pruebas con distintos tipos de sujetos que padezcan COVID-19 pero a la fecha, no estén muertos para poder hacer un mejor análisis de resultados.

### **Sujetos a predecir su riesgo de muerte prueba #1**

	ORIGEN	EDAD	SECTOR	ENTIDAD_UM	SEXO	ENTIDAD_NAC	ENTIDAD_RES	MUNICIPIO_RES	TIPO_PACIENTE	INTUBADO	...	DIABETES	EPOC
0	2	60	3	2	1	2	2	2	2	1	...	1	1
1	2	82	2	12	2	12	12	1	2	2	...	2	1
2	2	70	2	14	1	14	14	120	1	97	...	1	1
3	2	52	4	27	1	27	27	4	1	97	...	1	1
4	2	45	4	27	1	27	27	4	1	97	...	1	2
5	2	25	4	21	1	21	21	114	1	97	...	2	1

ASMA	INMUSUPR	HIPERTENSION	OTRA_COM	CARDIOVASCULAR	OBESIDAD	RENAL_CRONICA	TABAQUISMO
1	1	2	1	1	1	1	1
2	2	1	2	1	2	1	1
1	1	1	2	2	2	2	2
1	1	1	2	2	1	2	2
2	2	2	2	2	2	2	2
1	2	1	1	1	1	2	2

### **Datos predichos**

```
[[0.4745202 ]
 [0.4931217 ]
 [0.01501571]
 [0.00475939]
 [0.00441132]
 [0.00440106]]
```

### **Comentarios:**

Podemos apreciar que el modelo mejoro considerablemente y no dio resultados acercados a los que esperábamos, lo comprobamos con los primeros dos pacientes al primero de ellos le dimos 60 años y le adjuntamos que padece varios tipos de enfermedades en su organismo, y al segundo fue un caso similar, pero le dimos 82 años de edad, podemos ver qué al primero le dio el 47% de riesgo de muerte por contraer el COVID-19 y al segundo por su edad un 49% de riesgo siendo el más alto de la lista.



## Sujetos a predecir su riesgo de muerte prueba #2

	ORIGEN	EDAD	SECTOR	ENTIDAD_UM	SEXO	ENTIDAD_NAC	ENTIDAD_RES	MUNICIPIO_RES	TIPO_PACIENTE	INTUBADO	...	DIABETES	EPOC
0	2	75	3	2	1	2	2	2	2	2	...	2	2
1	2	82	2	12	2	12	12	1	1	1	...	1	1
2	2	75	2	14	1	14	14	120	1	97	...	1	1
3	2	52	4	27	1	27	27	4	1	97	...	2	2
4	2	45	4	27	1	27	27	4	1	97	...	1	2
5	2	18	4	21	1	21	21	114	1	1	...	1	1

ASMA	INMUSUPR	HIPERTENSION	OTRA_COM	CARDIOVASCULAR	OBESIDAD	RENAL_CRONICA	TABAQUISMO
2	2	2	2	2	2	2	1
1	1	1	1	1	1	1	1
1	1	1	2	2	2	2	2
2	2	1	1	1	2	2	2
2	2	2	2	2	2	2	2
1	1	1	1	1	1	1	1

## Datos predichos

```
[[0.26466122]
 [0.4034593 ]
 [0.0156737 ]
 [0.0156737 ]
 [0.00896008]
 [0.07834356]]
```

## Comentarios

Para esta segunda prueba del modelo entrenado, lo que realizamos fue lo siguiente en la primera fila añadimos un sujeto de 75 años que solo padece de tabaquismo su porcentaje fue de 26%, también agregamos a un sujeto de 82 años padece todas las enfermedades y su porcentaje fue de 40% de muerte, para el siguiente agregamos una persona de 75 años que padece solo de asma, diabetes e hipertensión su porcentaje fue de 0.15% muy por debajo, saltaremos a los últimos dos una persona que solo tiene diabetes y 45 años su porcentaje de riesgo fue 0.008% muy por debajo y para el último y más interesante una persona de 18 años que cuenta con todas las enfermedades fue de 0.078% de riesgo esto nos dice que mientras más joven sea el individuo menor es el riesgo.

### Sujetos a predecir su riesgo de muerte prueba #3

	ORIGEN	EDAD	SECTOR	ENTIDAD_UM	SEXO	ENTIDAD_NAC	ENTIDAD_RES	MUNICIPIO_RES	TIPO_PACIENTE	INTUBADO	...	DIABETES	EPOC
0	2	75	3	2	1	2	2	2	2	2	...	2	2
1	2	10	2	12	1	12	12	1	1	1	...	1	1
2	2	72	2	14	1	14	14	120	1	97	...	1	1
3	2	67	4	27	2	27	27	4	1	97	...	2	2
4	2	78	4	27	2	27	27	4	1	97	...	1	2
5	2	61	4	21	1	21	21	114	1	1	...	1	1

ASMA	INMUSUPR	HIPERTENSION	OTRA_COM	CARDIOVASCULAR	OBESIDAD	RENAL_CRONICA	TABAQUISMO
2	2	2	2	2	2	2	1
1	1	1	1	1	1	1	1
1	1	1	2	2	2	2	2
2	2	1	1	1	2	2	2
2	2	2	2	2	2	2	2
1	1	1	1	1	1	1	1

### Datos predichos

```
[[0.26466122]  
[0.03015539]  
[0.0156737 ]  
[0.0156737 ]  
[0.0156737 ]  
[0.30984578]]
```

### Comentarios

Para esta tercera prueba del modelo entrenado, lo que realizamos fue lo siguiente en la primera fila se conservó igual, pero en la segunda fila de 10 años de edad que cuenta con todas las enfermedades existentes y nos sorprendió mucho el resultado la probabilidad de riesgo de muerte fue de 0.030% muy por debajo lo que significa es que la edad es un valor de mucha vital relevancia para correr riesgo de muerte por esta enfermedad como así decirlo al niño no le hizo ni cosquillas el virus, eso si aún puede morir por las otras enfermedades que padece, pero no por el COVID-19, también en la penúltima fila se añadió una persona de 78 años el cual no padece ninguna enfermedad su riesgo fue de 0.015%, lo que significa que pese a la edad, si no tiene alguna otra enfermedad, el riesgo también es bajo.

### Sujetos a predecir su riesgo de muerte prueba #4

	ORIGEN	EDAD	SECTOR	ENTIDAD_UM	SEXO	ENTIDAD_NAC	ENTIDAD_RES	MUNICIPIO_RES	TIPO_PACIENTE	INTUBADO	...	DIABETES	EPOC
0	2	82	3	2	1	2	2	2	2	2	2	1	1
1	2	15	2	12	1	12	12	1	1	1	...	1	2
2	2	32	2	14	1	14	14	120	1	97	...	1	1
3	2	56	4	27	2	27	27	4	1	97	...	2	2
4	2	78	4	27	2	27	27	4	1	97	...	1	2
5	2	10	4	21	1	21	21	114	1	1	...	1	1

ASMA	INMUSUPR	HIPERTENSION	OTRA_COM	CARDIOVASCULAR	OBESIDAD	RENAL_CRONICA	TABAQUISMO
1	1	2	2	2	1	1	1
2	2	2	2	2	2	2	2
1	1	1	2	2	2	2	2
2	2	1	1	1	2	2	2
2	2	2	2	2	2	2	2
1	1	1	1	1	1	1	1

### Datos predichos

```
[[0.27662495]  
[0.03456692]  
[0.00407114]  
[0.0156737 ]  
[0.0156737 ]  
[0.06905888]]
```

### Comentarios

Para esta cuarta prueba del modelo entrenado, lo que realizamos fue lo siguiente en la primera fila se agregó un sujeto de 82 años que padece unas cuantas enfermedades como diabetes, asma y obesidad su porcentaje de riesgo de muerte fue de 27% lo que significa que, si es probable que muera por contraer dicho virus, también se agregó en la cuarta fila una persona que padece de hipertensión y otra complicación de 56 años y su riesgo de muerte fue algo bajo con 0.015%

### Sujetos a predecir su riesgo de muerte prueba #5

	ORIGEN	EDAD	SECTOR	ENTIDAD_UM	SEXO	ENTIDAD_NAC	ENTIDAD_RES	MUNICIPIO_RES	TIPO_PACIENTE	INTUBADO	...	DIABETES	EPOC	ASMA
0	2	82	3	2	1	2	2	2	2	2	...	1	1	
1	2	72	2	12	1	12	12	1	1	1	...	1	1	
2	2	92	2	14	1	14	14	120	1	97	...	1	1	
3	2	60	4	27	2	27	27	4	1	97	...	2	2	
4	2	78	4	27	2	27	27	4	1	97	...	1	2	
5	2	53	4	21	1	21	21	114	1	1	...	1	1	

ASMA	INMUSUPR	HIPERTENSION	OTRA_COM	CARDIOVASCULAR	OBESIDAD	RENAL_CRONICA	TABAQUISMO
1	1	2	2	2	1	1	1
2	2	2	2	2	2	2	2
1	1	1	2	2	2	2	2
1	2	1	1	1	2	2	2
1	2	2	2	2	2	2	2
1	1	1	1	1	1	1	1

### Datos predichos

```
[[0.27662495]
 [0.33673918]
 [0.0156737 ]
 [0.0156737 ]
 [0.0156737 ]
 [0.30939665]]
```

### Comentarios

Para esta cuarta prueba del modelo entrenado, lo que realizamos fue lo siguiente en la segunda fila agregamos una persona que tiene 72 años y padece de diabetes salió con una probabilidad de riesgo alta de 33% lo que significa que aun con una sola enfermedad si eres muy adulto bastara con que tengas una enfermedad para poder correr el riesgo de morir por contraer este virus.

## **CONCLUSION**

Como conclusión podemos decir que, efectivamente, vivimos en una sociedad la cual día con día no está segura a fuera con el COVID-19 en las calles, notamos en la elaboración de este proyecto que los datos sobre casos de COVID-19 fueron de vital importancia para diagnosticar el comportamiento de los individuos que padezcan alguna enfermedad o que tengan una edad avanzada pueden correr alto riesgo de morir debido a contraer este virus, también comprobamos que lo que se dice de esta enfermedad sobre que los niños o personas de una edad muy joven son los menos afectados por contraer este virus es verdad, ya que en general suelen salir con un porcentaje mucho más bajo, es por esto que debemos guardar la sana distancia para evitar la propagación de este virus ya que aunque a nosotros no nos afecte, a nuestros seres queridos de una edad adulta o que padezca de alguna enfermedad si podría afectarlos, y la mejor manera en lo que se desarrollan y se aprueban las vacunas, es quedarse en casa.

También podemos destacar el uso de las redes neuronales en el mundo de la inteligencia artificial para realizar predicciones con datos reales y aunque nos hubiera gustado que los porcentajes de riesgo de muerte fueran un poco más reales y menos inexactos, para que esto suceda, sabemos que necesitamos de un red y un modelo muy bien calibrado y entrenado para que las predicciones sean lo más acercadas a la realidad, para esto se requiere de mucho entrenamiento , ya que pudimos notar que con una horas de entrenamiento y con miles de registros no basta, pero con un número de neuronas adecuados y un numero de épocas alto se podrían tener mucho mejores resultados, el entrenamiento de una red neuronal con muchos datos puede llevar días para que este muy bien entrenada, pero una vez teniendo desarrollado el modelo adecuado se puede almacenar en disco y se le podrán hacer las pruebas que se requieran y esto lo supimos gracias a la persistencia de modelos que posee la librería Keras que nos permitió realizar las diversas pruebas al modelo y sacar nuestro análisis a los datos de una manera muy sencilla.

## **FUENTES**

- [https://ml4a.github.io/ml4a/es/neural\\_networks/](https://ml4a.github.io/ml4a/es/neural_networks/)
- <https://puentesdigitales.com/2018/02/14/todo-lo-que-necesitas-saber-sobre-tensorflow-la-plataforma-para-inteligencia-artificial-de-google/>
- <https://www.aarp.org/espanol/salud/enfermedades-y-tratamientos/info-2020/coronavirus-sintomas-evitar-el-contagio.html>
- <https://coronavirus.gob.mx/datos/>
- <https://coronavirus.gob.mx/>
- <https://es.wikipedia.org/wiki/Keras>
- [https://es.wikipedia.org/wiki/Red\\_neuronal\\_artificial](https://es.wikipedia.org/wiki/Red_neuronal_artificial)
- [https://www.ibm.com/support/knowledgecenter/es/SS3RA7\\_sub/modeler\\_mainhelp\\_client\\_ddita/components/neuralnet/neuralnet\\_model.html](https://www.ibm.com/support/knowledgecenter/es/SS3RA7_sub/modeler_mainhelp_client_ddita/components/neuralnet/neuralnet_model.html)
- <https://enmilocalfunciona.io/deep-learning-basico-con-keras-parte-1/>