

UNIVERSIDAD NACIONAL TECNOLÓGICA DE LIMA SUR
FACULTAD DE INGENIERÍA Y GESTIÓN
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



**“MODELO DE RED NEURONAL ARTIFICIAL PARA EL
PRONOSTICO DEL CRECIMIENTO DE CONTAGIOS POR COVID19
EN EL DISTRITO DE SAN JUAN DE MIRAFLORES”**

TRABAJO DE SUFICIENCIA PROFESIONAL

Para optar el Título Profesional de
INGENIERO DE SISTEMAS

PRESENTADO POR EL BACHILLER

ALEJANDRO GAMARRA, LUIS ENRIQUE

**Villa El Salvador
2020**

Dedicatoria

A mis padres, hermana y sobrinas ya que ellos son los que me apoyan y ayudan a ser un gran profesional y una gran persona.

Agradecimiento

A mis profesores que me ayudaron a forjarme como profesional y a mis familiares que me dieron su apoyo y ayuda incondicional.

Índice

INTRODUCCIÓN	11
OBJETIVOS	12
CAPÍTULO I: MARCO TEÓRICO	13
1.1. Bases Teóricas	13
1.1.1 Antecedentes.....	13
1.1.2 Pronóstico	15
1.1.3 Redes Neuronales Artificiales	16
1.1.4 Metodología de Kaastra y Boyd.....	27
1.2. Definición de términos básicos	36
CAPÍTULO II: METODOLOGÍA DE DESARROLLO DEL TRABAJO PROFESIONAL	38
2.1. Delimitación temporal y espacial del trabajo	38
2.1.1 Temporal	38
2.1.2 Espacial.....	38
2.2. Determinación y análisis del problema	38
2.3. Modelo de solución propuesto	39
2.3.1 Paso 1: Selección de la Variable	39
2.3.2 Paso 2: Recolección de Datos	40
2.3.3 Paso 3: Pre - procesamiento de datos	40
2.3.4 Paso 4: Entrenamiento, Prueba y Validación de datos.	40
2.3.5 Paso 5: Paradigma de Redes Neuronales	41
2.3.6 Paso 6: Criterios de Evaluación.	42
2.3.7 Paso 7: Entrenamiento de la Red Neuronal	42
2.3.8 Paso 8: Implementación.	43
2.4. Resultados	56
CONCLUSIONES	57
RECOMENDACIONES	58

Bibliografía	59
ANEXOS	63
Anexo 1: Cantidad de Camas de Hospitalización en los distritos de Lima Metropolitana	63
Anexo 2: Cantidad de Muertos en el Distrito de San Juan de Miraflores	63
Anexo 3 Diagrama de Drew Conway	64
Anexo 4: Overfitting y Underfitting	64
Anexo 5: Procesamiento de Datos	65
Anexo 6: Formula de Des estandarización.	69

Lista de Figuras

Figura 1- Comparación de una Neurona Biológica y una Artificial	17
Figura 2 - Modelo de McCullonch y Pitts	17
Figura 3 – Perceptrón	18
Figura 4 - Regla Delta	19
Figura 5 - Perceptrón Multicapa	20
Figura 6 - Función Lineal	21
Figura 7 - Función RELU	21
Figura 8 - Función Tangente Hiperbólica	22
Figura 9 - Función Sigmoide	23
Figura 10 - Normalización Estadística	29
Figura 11 - Normalización Max - Min	30
Figura 12 - Normalización Sigmoide	30
Figura 13 - Error Cuadrático Medio	34
Figura 14 - Error Absoluto Medio	34
Figura 15 - Error Porcentual Absoluto Medio	34
Figura 16 - Línea de Tiempo	41
Figura 17 - Google Colab	43
Figura 18 - Pyhton, Numpy, Keras	44
Figura 19 - Importación de Librerías y Carga de Datos	44
Figura 20 - Particionamiento de Datos	45
Figura 21 - Estandarización de Datos	45
Figura 22 - Datos de Entrada y Salida	46
Figura 23 - Variables de la Red Neuronal	47
Figura 24 - Estructura de la Red Neuronal Artificial	48
Figura 25 - Creación de Capas	48
Figura 26 - Configuración de la Red Neuronal	49
Figura 27 - Entrenamiento de la Red	49
Figura 28 - Resultado de Entrenamiento	50
Figura 29 - Grafico de Error Vs Épocas	51
Figura 30 - Grafico de los Datos de Entrenamiento	52
Figura 31 - Datos de Prueba	53
Figura 32 - Resultado con los Datos de Prueba	53

Figura 33 - Grafico de los Datos de Prueba	54
Figura 34 - Resultados con los Datos de Validación	54
Figura 35 - Grafico de Datos de Validación	55
Figura 36 – Pronóstico	55

Lista de Tablas

Tabla 1 - Metodología de Kaastra y Boyd	28
Tabla 2 - Infectados por COVID-19	38
Tabla 3 - Personal de Salud Infectados por COVID-19 al 20/03/2020	39
Tabla 4 - Datos de Entrenamiento	51
Tabla 5 - Datos de Prueba	53
Tabla 6 - Datos de Validación	55

RESUMEN

En la Actualidad el problema que aqueja a la Humanidad es el Virus Covid-19 o también conocido como SARS COV 2 el cual lleva cobrando la vida de millones de personas. En Perú la cifra de personas infectadas por esta enfermedad a sobre pasado los 500 mil contagiados y bordea una cantidad de 26 mil muertos. (Instituto Nacional de Salud y Centro Nacional de Epidemiología, Prevención y Control de Enfermedades, 2020)

El distrito de San Juan de Miraflores fue establecido mediante la Ley N° 15382 el 12 de enero de 1965 siendo su capital Ciudad de Dios. (Municipal Distrital de San Juan de Miraflores, 2020)

Se identificó según las muestras tomadas en el Mercado “CIUDAD DE DIOS” por el Equipo de Respuestas Rápidas de la dirección de Redes Integrales de Salud (DIRIS), el 54.5% dieron positivo para Covid-19. (Ministerio de Salud, 2020)

La falta de información proyectada de contagiados por Covid-19 es una necesidad álgida pues esta información ayudara a la toma de decisiones como tener preparado camas de hospitalización para personas infectadas por Covid-19, tener previsto cantidad de equipos de protección personal, etc.

También se encontró que según el informe del Ministerio de Salud la oferta de camas UCI para personas con COVID-19 es de 0 camas disponibles para el Distrito de San Juan de Miraflores (Ver Anexo 1), se tiene alrededor de 663 habitantes por cama hospitalaria. (Oficina General de Estadística e Informática, 2005)

Existe una preocupación en el distrito de San Juan de Miraflores con respecto al índice de decesos que publica el MINSA, con el promedio de decesos de 107 personas por mes y un grado de letalidad del 5% (Ver Anexo 2).

La metodología que se utilizara se basada en el trabajo realizado por Kaastra y Boyd en el año 1996, esta metodología consta de 8 pasos, explica de una manera detallada como poder diseñar tu red neuronal artificiales de pronóstico de datos, desde la recolección de información y selección de variable hasta la implementación de la red neuronal artificiales.

Obteniendo como Resultado una Red Neuronal Artificial con 3 capas, una capa de entrada con dos neuronas, una capa oculta compuesta por 4 neuronas y una capa de salida con una 1 neurona. El error obtenido en la fase de entrenamiento fue del 1.59%, en la fase de Prueba fue del 0.46% y en la fase de Validación fue del 2.5%; obteniendo finalmente el pronóstico de la siguiente semana correspondiente al periodo del 30 de agosto al 12 de setiembre.

INTRODUCCIÓN

A lo largo de los tiempos la humanidad ha sufrido diversos tipos de pandemias tales como la peste negra o peste bubónica, la gripe española, la gripe asiática, la gripe H1N1 o gripe porcina la cual ha cobrado la vida de millones de personas en todo el mundo.

Desde finales del año 2019, el 31 de diciembre un virus llamado Covid-19 o SARS COV 2 fue notificado en Wuhan (China) el cual viene cobrando la vida de millones de personas en el mundo. (Organización Mundial de la Salud, 2020)

En el Perú se registró la aparición de este virus en el 06 de marzo, el gobierno tomó medidas para poder controlar la rápida propagación de esta enfermedad. (MINSA, 2020)

A pesar de haberse tomado medidas estrictas por el gobierno, la falta de infraestructura, médicos intensivistas y la falta equipos de protección personal (EPP) de los hospitales complicaba aún más la atención de las personas contagiadas debido a la gran cantidad de personas infectadas por este virus. (Dr. Valverde, 2020) (Rivas, 2020)

La distribución de estos implica saber la situación actual y pronosticar el crecimiento del virus para los distintos departamentos y distritos, para así poder distribuir de manera eficiente los productos necesarios para la atención de los pacientes.

Este trabajo busca desarrollar una red neuronal artificial para pronosticar el crecimiento de contagios por Covid-19 en el distrito de San Juan de Miraflores.

Está compuesto por 2 capítulos; en el primer capítulo se establece las bases teóricas y las definiciones de los términos utilizado y el último capítulo se desarrolla la metodología de Kaastra y Boyd para la elaboración de la red neuronal artificial y los resultados obtenidos.

OBJETIVOS

a) Objetivo General

- Pronosticar con la red neuronal artificial el crecimiento de contagios por Covid-19 en el Distrito de San Juan de Miraflores.

b) Objetivo Específicos

- Diseñar una red neuronal artificial para el pronóstico del crecimiento de contagios por covid-19 en el Distrito de San Juan de Miraflores.
- Entrenar una red neuronal artificial para el pronóstico del crecimiento de contagios por covid-19 en el Distrito de San Juan de Miraflores.

CAPÍTULO I: MARCO TEÓRICO

1.1. Bases Teóricas

1.1.1 Antecedentes

Antecedentes Internacionales

En el trabajo realizado por Lilia Alejandra Flores Castillo y Martín Carlos Ramales Osorio, titulado “Redes Neuronales Artificiales: efecto asimetría y curtosis en la evaluación de portafolios de inversión” en el año 2017.

El problema que encontró Flores y Ramales es que el nivel de incertidumbre hace cada vez más difícil la diversificación del riesgo por lo que se requiere estrategias que apoyen en la toma de decisiones. La metodología usada para la resolución de este problema fue la propuesta por Kaastra y Boyd en 1996, logran obtener como resultado una red neuronal artificial que pronostique los rendimientos de los activos financieros con una mayor probabilidad de obtener rendimientos positivos. (Flores Castillo & Ramales Osorio, 2017)

En el trabajo realizado por Víctor Daniel Gil Vera, titulado “Forecasting Electricity Demand for Small Colombian Populations” en el año 2015.

El problema que encontró Gil fue que el pronóstico que se realizaba con otras técnicas con respecto al consumo eléctrico, se mejoraría con la ayuda de las redes neuronales. La metodología usada para la resolución de este problema fue la propuesta por Kaastra y Boyd en 1996, obteniendo como resultado un mejor pronóstico del consumo eléctrico con un porcentaje de error del 2% logrando superar a otras metodologías. (Gil Vera, 2015)

Antecedentes Nacionales

El trabajo realizado por Veneranda Calla Charrez titulado “Modelo de pronóstico para número de egresos hospitalarios usando redes

neuronales – Hospital Regional Manuel Núñez Butrón – Puno, 2009 – 2018” en el 2019.

El problema que encontró Calla fue que el Hospital Regional Manuel Núñez Butrón no contaba con una documentación que les permita tomar referencias sobre pronósticos de los egresos hospitalarios. La metodología usada para la resolución de este problema fue la propuesta por Kaastra y Boyd en 1996 obteniendo como resultado una red neuronal multicapa la cual tenía un error de pronóstico de los egresos hospitalarios del 2% en la fase de entrenamiento y 4% de error en la fase de validación. (Calla Charrez, 2019)

El trabajo realizado por Manuel Alexander Malaver Elera, titulado “Aplicación de Redes Neuronales para determinar el pronóstico de las ventas en la empresa catering & Buffet MyS ubicada en la ciudad de Piura” en el 2015.

El problema que encontró Malaver fue que la empresa Catering & Buffer Mys tenía un porcentaje de error muy elevado lo cual implicaba proyecciones erróneas. La metodología usada para la resolución de este problema fue la propuesta por Kaastra y Boyd en 1996 obteniendo como resultado un menor porcentaje de error del 1% en el pronóstico de ventas, de igual manera se comprobó que las redes multicapas son las mas adecuadas para realizar pronósticos. (Malaver Elera, 2015)

Antecedentes Locales

El trabajo realizado por Percy Humberto Cueva Rios titulado “Evaluación de índices de confiabilidad en redes de Electrocentro S.A. para mejorar la calidad de servicio utilizando redes neuronales” en el 2019.

El problema que encontró Cueva fue que el servicio eléctrico que ofrecía la empresa Electrocentro S.A tenía interrupciones lo cual generaba malestar en los usuarios de las zonas rurales. La

metodología usada para la resolución de este problema fue la propuesta por Kaastra y Boyd en 1996 obteniendo como resultado una red neuronal multicapa la cual ayudo a la disminución de la frecuencia y el tiempo de las interrupciones para los alimentadores A4701 y A4702 de la S.E Ninatambo, considerando la reubicación e inclusión de equipos de protección en el recorrido de las líneas. (Cueva Rios, 2019)

1.1.2 Pronóstico

El pronóstico es la estimación de una o varias variables con información actual o del pasado para dar a conocer un evento futuro. (Villarreal, 2016)

El pronóstico es la ciencia que permite conocer o predecir el comportamiento futuro a través de datos históricos. Para poder predecir hechos futuros se debe tener información confiable para poder modelar un pronóstico lo más acertado posible, para esto se deberá encontrar los patrones que identifiquen nuestros datos recolectados. (Rojas Díaz, 2010)

CLASIFICACIÓN DE MÉTODOS DE PRONOSTICO

Los pronósticos se clasifican según el tipo de método o técnica que se utilice, Arturo Farrera lo clasifica en dos. (Farrera Gutiérrez, 2013)

MÉTODOS CUALITATIVOS

Para este tipo de método se requiere la experiencia y el juicio de expertos que establezcan de manera subjetiva el pronóstico. (Farrera Gutiérrez, 2013)

Aquí presentamos algunos métodos cualitativos.

- Método Delphi
- Pronostico Visionario
- Analogías Históricas
- Consenso de un Panel

MÉTODOS CUANTITATIVOS

Para este tipo de método se requiere de información histórica o actual la cual sea confiable. (Farrera Gutiérrez, 2013)

Aquí presentamos algunos métodos cuantitativos.

- Regresión Lineal Simple
- Promedio Móvil Simple
- Promedio Móvil Ponderado
- Suavizado Exponencial Simple
- Promedio Móvil Lineal

1.1.3 Redes Neuronales Artificiales

Las Redes Neuronales artificiales son técnicas inmersas en el campo del machine learning que han tenido mucha fama durante la última década.

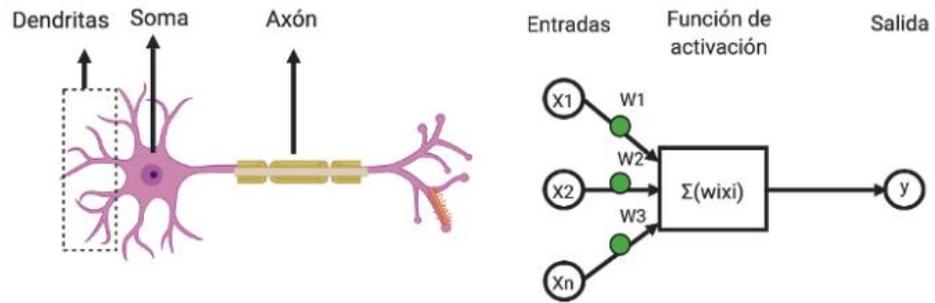
Las Redes Neuronales Artificiales están inspiradas en las neuronas biológicas del ser humano estas tratan de emular el comportamiento del sistema neuronal biológico. (Matich, Facultad Regional del Rosario - Universidad Tecnológica Nacional, 2001)

En la Figura 1 Se refleja la comparación que tiene cada componente de una neurona biológica con la artificial.

Para la neurona biología las dendritas son las que se encargan de recoger la información ya sea del exterior o de una neurona previa a la actual la cual enviara a través de impulsos electro-químicos, el soma se encargara de procesar toda esa información capturada por las dendritas y el axón llevara el resultado de la neurona hacia otras células. La analogía para las neuronales artificiales donde las dendritas vendrían a hacer los datos que le entregamos a la neurona artificial para ser procesados, el soma vendría a hacer el proceso matemático de multiplicar cada dato de entrada por su respectivo peso luego estos datos serán procesados por la función de activación

y el axón vendría a hacer el resultado que nos dará la neurona artificial.

Figura 1- Comparación de una Neurona Biológica y una Artificial

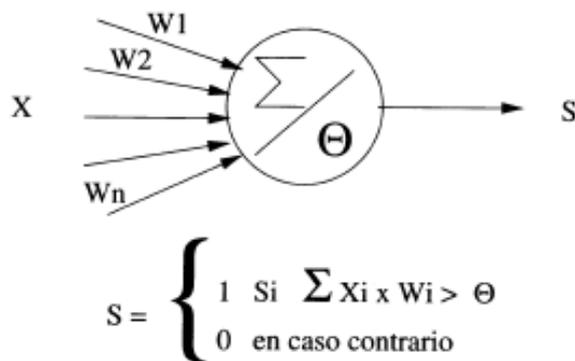


Fuente: (García, 2020)

El primer modelo de red neuronal artificial fue propuesto por los investigadores Warren McCulloch y Walter Pitts en el artículo “A Logical Calculus of the Ideas Immanent in Nervous Activity” que fue publicado en el año de 1943, este tipo de red neuronal se le llamado la célula de McCulloch-Pitts. (McCulloch & Pitts, 1988)

Plantearon el funcionamiento de una neurona biológica de manera simplicidad con dos estados apagado y encendido. A partir de este modelo se define las redes neuronales artificiales

Figura 2 - Modelo de McCulloch y Pitts



Fuente: (Isasi Viñuela & Galván León, 2004)

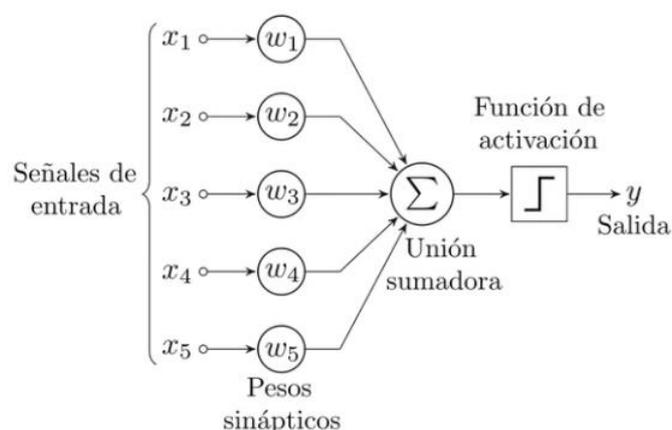
Plantearon el funcionamiento como se aprecia en la figura 2, donde X viene a ser todas las entradas que recibe la neurona procedente de la salida de otras células, W_1, W_2 hasta W_n vienen hacer los pesos correspondientes a cada una de las conexiones de las entradas, dentro de la célula se realizara la sumatoria de la multiplicación de cada una de las entradas con su correspondiente peso, el umbral está representado por Θ , para el procesamiento de la información se realizara de la siguiente manera si la sumatoria es mayor al umbral la célula nos dará como resultado 1 y en caso contrario nos dará el resultado de 0.

Tipo de Redes Neuronales artificiales

Perceptrón.

En el año de 1958 Frank Rosenblatt inspirado en la célula de McCullonch-Pitts creo el perceptrón. (Rosenblatt, 1958) La arquitectura de esta red es también llamada mono capa ya que solo cuenta con una capa la cual es la de salida como se aprecia en la figura 3, este tipo de red utiliza los valores de salida pasados por la función de activación para el aprendizaje.

Figura 3 – Perceptrón



Fuente: (Isasi Viñuela & Galván León, 2004)

En el Año de 1960, los investigadores Bernard Widrow y Marcian Hoff diseñaron lo que denominaron como ADALINE (Adaptive Linear

Neuron), ADALINE a diferencia del perceptrón toma en cuenta el grado de corrección de la salida proporcionada por la red neuronal con respecto a lo deseado. Esta es llamada la regla Delta que la calcula con la diferencia de la salida esperada y la salida obtenida de la red neuronal al pasarles los datos de entrada. (Widrow & Hoff, 1960)

Como se aprecia en la figura 4, “d” representa a la salida deseada con respecto a la entrada “p” esta se restará con la salida que da como resultado la red neuronal que está representada por “y”.

Figura 4 - Regla Delta

$$|d^p - y^p|$$

Fuente: Propia

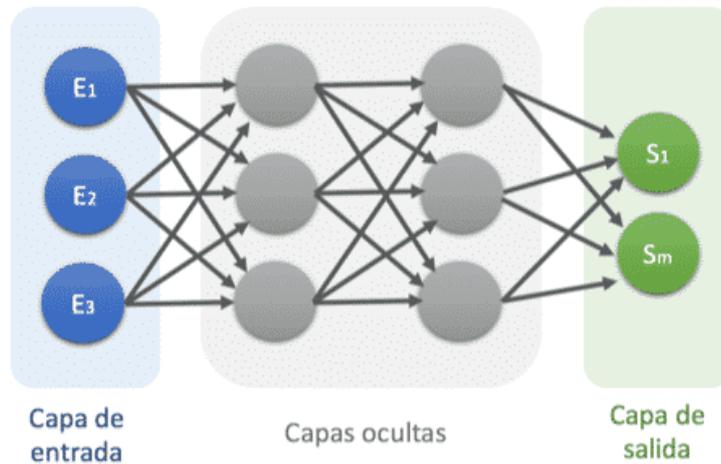
Perceptrón Multicapa

En 1969 Marvin Minsky y Seymour Papert en su libro PERCEPTRONS en el cual muestran las dificultades que tienen los perceptrones para separabilidad no lineal, ellos propusieron “redes Gamba” que no son nada más que redes con capas ocultas. (Minsky & Papert, 1988)

El perceptrón multicapa o también llamado red multicapa fue la solución a estos problemas de separabilidad no lineal.

En la figura 5 se aprecia la estructura de las redes multicapa, conformada por la capa de entrada la cual son los datos propiamente dichos, las capas ocultas que procesan la información y la capa de salida que nos dará el resultado luego de haber procesado la información.

Figura 5 - Perceptrón Multicapa



Fuente: (Calvo, 2018)

Sin embargo, para este tipo de arquitecturas la regla de aprendizaje que fue propuesto en el perceptrón no se podía aplicar para las redes multicapa, esto sirvió de base para estudios posteriores.

En el año de 1986 D. Rumelhart, G. Hinton y R. Williams presentan la manera de poder retropropagar el error obtenido en la capa de salida hacia las capas ocultas, consiste en ajustar los pesos de las conexiones para poder minimizar el error obtenido de la diferencia entre la salida deseada y la salida obtenida por la red. Este algoritmo se llamó Backpropagation, esto no es más que la generalización de la regla delta que se tenía en el perceptrón. (Rumelhart, Hinton, & Williams, 1986)

Funciones de Activación de una Red Neuronal Artificial.

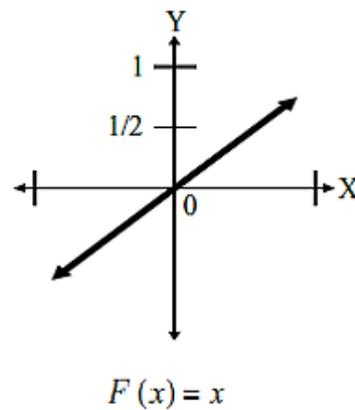
La Función Activación de la red neuronal artificial tiene como objetivo limitar el rango de salida de la red neuronal artificial. Esto quiere decir que los datos de entrada que han sido multiplicados por sus respectivos pesos son pasados por una función ya sea lineal o no lineal. Las Capas intermedias o ocultas y la capa final de nuestra red neuronal artificial tiene una función de activación. (Matich, Facultad Regional Rosario - Universidad Tecnológica Nacional, 2001)

Aquí se presentan algunas funciones de Activación:

- **Función Lineal**

La función Lineal o también llamada función de Identidad. Este tipo de función de activación se utilizó en arquitecturas pequeñas como el perceptrón mono capa, ya que, si es utilizado en redes neuronales con arquitecturas más complejas, las cuales en sus capas tiene este tipo de función de activación equivaldría a una sola neurona. (Ver Figura 6)

Figura 6 - Función Lineal

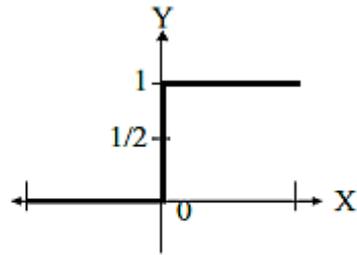


Fuente: Propia

- **Función RELU**

Función RELU o también llamada Rectificante, esta función lo que realiza con los valores de entrada es si el valor es negativo la función lo anula y en caso fuese un valor positivo deja el valor tal y como se ingresó. (Ver Figura 7)

Figura 7 - Función RELU



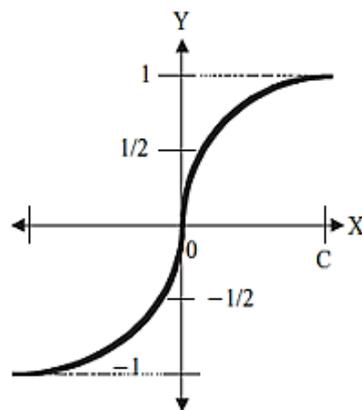
$$F(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases}$$

Fuente: Propia

- Función Tangente Hiperbólica

La Función de Activación Tangente Hiperbólica se encuentra acotada entre los valores de -1 a 1 donde los valores más bajos esta función lo aproxima a -1 y en caso el valor sea más alto lo aproxima a 1. Este tipo de función es muy similar a la función Sigmoide. (Ver Figura 8)

Figura 8 - Función Tangente Hiperbólica



$$F(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}}$$

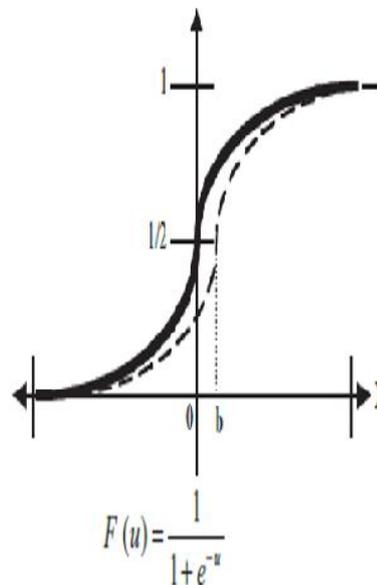
Fuente: Propia

- Función Sigmoide

La función Sigmoide o también llamada función logística, la salida de esta función es continua y se encuentra entre un

rango de 0 a 1. Si se evalúa datos muy negativos esta función la iguala a 0 y en su defecto sean valores altos la aproxima a 1. Este tipo de función tiene un buen desempeño en la última capa de la red neuronal artificial. (Ver Figura 9)

Figura 9 - Función Sigmoide



Fuente: Propia

Tipos de Aprendizaje de una Red Neuronal Artificial.

Las Redes Neuronales artificiales tratan de imitar el comportamiento que tiene el cerebro humano.

Pedro Isasi y Inés Galván en su libro *Redes neuronales Artificiales* un enfoque práctico define como aprendizaje para las redes neuronales, el mecanismo por el cual se procesa la información recopilada antigua para poder transformarla en conocimiento. Las redes neuronales son capaces de abstraer características esenciales de los datos, para esto es importante que los datos sean lo más representativo y significativo para nuestro problema a resolver. (Isasi Viñuela & Galván León, 2004)

El aprendizaje consiste en encontrar los pesos de todas las conexiones de la red neuronal artificial que hagan que nuestra red pueda resolver el problema lo más eficientemente posible, para esto

se le introduce los datos de ejemplo de nuestro set de datos de aprendizaje para que nuestra red paulatinamente modifique estos pesos hasta obtener un resultado óptimo. (Isasi Viñuela & Galván León, 2004)

Se distinguen tres tipos de aprendizaje según (Isasi Viñuela & Galván León, 2004)

- **Aprendizaje Supervisado.**

Para este tipo de aprendizaje se tiene dos pares de datos, los datos de entrada propiamente dichos o Input Data y los datos de salida que tiene el resultado u Output Data. (Isasi Viñuela & Galván León, 2004)

Este tipo de aprendizaje busca descubrir la relación que tienen los datos de entrada con respecto a los datos de salida es decir el aprendizaje surge de mostrarle el resultado que deseamos obtener, que vienen a ser nuestros datos de salida, para un determinado dato de entrada. (Isasi Viñuela & Galván León, 2004)

Esto quiere decir que cada vez que un dato de entrada es introducido se procesa iterativamente para obtener un resultado, esto influirá en la modificación de los pesos ya que a mayor sea la diferencia entre el resultado obtenido y el esperado mayor será el cambio del peso de la red neuronal y entre menor sea la diferencia menor será el cambio de los pesos.

Para esto hay un agente externo el cual se encargará de comprobar si el resultado producido por la red neuronal artificial es la adecuada comparándola con el resultado esperado. (Isasi Viñuela & Galván León, 2004)

- **Aprendizaje No Supervisado.**

El aprendizaje no supervisado solo tiene un tipo de dato que es el dato de entrada o Input data. Para este tipo de aprendizaje no se existe datos de salida u output data y tampoco un agente externo que valide o determine si la red neuronal artificial está obteniendo buenos resultados. (Isasi Viñuela & Galván León, 2004)

Para este tipo de aprendizaje la red neuronal artificial modificara los pesos de sus conexiones de acuerdo a las características, rasgos, regularidades o redundancias que encuentre en los datos de entrada. (Isasi Viñuela & Galván León, 2004)

- **Aprendizaje Reforzado.**

El aprendizaje reforzado se encuentra entre el aprendizaje supervisado y no supervisado. Las similitudes que tiene con el aprendizaje supervisado es que existe una forma de supervisión, y las similitudes que tiene con el aprendizaje no supervisado es que no se entregue de forma específica el resultado que se desea obtener para una entrada en particular. (Isasi Viñuela & Galván León, 2004)

Esto quiere decir que la red neuronal que tenga este tipo de aprendizaje reforzado recibe una retroalimentación del entorno una vez que haya seleccionado una determinada salida a una entrada en particular. (Isasi Viñuela & Galván León, 2004)

También la denominan aprendizaje por premio-castigo, ya que el agente de acuerdo a la salida que proporcione la red neuronal se le premiara o penalizara sin proporcionar detalles de lo que se espera.

Aplicación de las Redes Neuronales artificiales.

- **Manufacturas:** (Gómez, Sánchez, Ocampo, & Restrepo, 2012) utilizaron las redes neuronales en la clasificación de arcillas con el propósito de disminuir la cantidad de productos rechazados para una fábrica del sector construcción obteniendo resultados del 97.9% de precisión.
- **Medicina:** (Mateus, González, & Branch, 2014) utilizaron Entorno inteligente (EVI) el cual media la vista de los pacientes y esta le pasaba la información a la red neuronal la cual clasificaba a los pacientes con astigmatismo, miopía e hipertrofia.
- **Finanzas:** (Villada, Muñoz, & García-Quintero, 2016) aplicaron las redes neuronales para pronosticar el precio del oro en el mercado de Londres logrando obtener un error del 5% en sus pronósticos. (Alarcón Jaque, 2014) utilizó las redes neuronales para la predicción del signo de las variaciones semanales de los multi-fondos pertenecientes a AFP Cuprum S.A.
- **Seguridad:** (Pérez rivera, Britto Montoya, & Isaza Echeverry, 2005) Aplicaron las redes neuronales para detectar intrusos en redes y sistemas de información logran obtener resultados de 98.09% de precisión en la detección de intrusos.
- **Psicología:** (Fumero & Navarrete, 2016) Aplicación de redes neuronales con el objetivo de reducir la complejidad de sistemas psicopatológicos tales como la somatizaciones, ansiedad, depresión y disfunción social logran obtener resultados con un error promedio del 5%.
- **Traducción:** (Linn, 2018) aplicaron las redes neuronales artificiales en la traducción de artículos de noticia del chino al

inglés igualando el desempeño (calidad y precisión) que tiene un humano.

- **Voz:** (Ramos, Rojas, & Saby, 2016) aplicación de redes neuronales que permitan a personas con trastornos de audición (TA) la posibilidad de emplearla como una herramienta de apoyo en el habla obteniendo un 81% de precisión.
- **Conducción:** (Waymo, 2018) aplicaron las redes neuronales para la conducción de vehículos de carga autónomos para la logística de productos en el proyecto Waymo.

1.1.4 Metodología de Kaastra y Boyd

Para Kaastra y Boyd las redes neuronales artificiales son grandes aproximadores de funciones, altamente flexibles y pueden tolerar el ruido mejor que otros métodos; pero la elaboración de estas redes neuronales en lo largo del tiempo ha requerido mucho ensayo y error debido a la gran cantidad de parámetros que debe tenerse en cuenta para la elaboración. Y las diversas dificultades para poder obtener una solución estable como el peligro del sobre ajuste o overfitting, el tiempo de entrenamiento excesivo. Si bien muchas organizaciones han implementado esta tecnología, muy pocas la han implementado con éxito, debido a la cantidad de recursos y horas que han tenido que experimentar y ajustar las redes neuronales para obtener un resultado exitoso. (Kaastra & Boyd, 1996)

Por lo cual Kaastra y Boyd proponen 8 pasos para poder diseñar un modelo de red neuronal en el cual explica los parámetros que se deben tener en cuenta para la elaboración de una red neuronal, los errores más comunes que tienen los investigadores. Si bien no se tiene reglas estrictas para la elaboración de una red neuronal, existen reglas generales y consejos que ayudan a los investigadores a obtener un diseño de red neuronal que resuelva el problema con éxito.

Tabla 1 - Metodología de Kaastra y Boyd

Eight steps in designing a neural network forecasting model

Step 1: Variable selection

Step 2: Data collection

Step 3: Data preprocessing

Step 4: Training, testing, and validation sets

Step 5: Neural network paradigms

number of hidden layers

number of hidden neurons

number of output neurons

transfer functions

Step 6: Evaluation criteria

Step 7: Neural network training

number of training iterations

learning rate and momentum

Step 8: Implementation

Fuente: (Kaastra & Boyd, 1996)

En Tabla 1 se aprecia los 8 pasos propuestos por Kaastra y Boyd

1. Selección de Variable

La selección de la variable es de gran importancia en el éxito del diseño de la red neuronal artificial, la preocupación se centra en los datos brutos a partir de los cuales se elegirá la variable a pronosticar. (Kaastra & Boyd, 1996)

La frecuencia de estos datos depende del objetivo del investigador, esto quiere decir si el horizonte del investigador es a largo plazo utilizara datos anuales o mensuales y si en su defecto es corto plazo utilizara datos diarios o semanales. (Kaastra & Boyd, 1996)

2. Recolección de Data

El investigador debe tener muy en cuenta el costo y la disponibilidad que tendrán la recolección de los datos con respecto a la variable elegida. Ya que si los datos para poder elaborar una red neuronal son muy costosos recolectarlos o no

se tiene disponibilidad de ellos debido a que la información sea muy sensible para la empresa será muy difícil para el investigador proseguir con su investigación. Estos datos también deben ser obtenidos por un proveedor confiable ya que si la información sea poco confiable traería muchos problemas para que la red neuronal de un resultado exitoso en la resolución del problema. (Kaastra & Boyd, 1996)

3. Preprocesamiento de Datos

El preprocesamiento de los datos se refiere a la transformación de los datos brutos obtenidos de las variables de entrada y salida para así poder minimizar el ruido y que nuestra red neuronal pueda detectar tendencias y patrones relevantes. (Kaastra & Boyd, 1996)

Las variables de entrada y salida muy pocos investigadores la introducen sin procesarlas, normalmente los investigadores realizan escalamientos de sus datos para obtener mejores resultados. (Kaastra & Boyd, 1996)

A continuación, se presentan algunas de las normalizaciones usadas.

- **Normalización Estadística**

Figura 10 - Normalización Estadística

$$z = \frac{x - \bar{a}}{\sigma}$$

Fuente: Propia

En la Figura 10; x es el dato a normalizar, \bar{a} es la media del conjunto de datos, σ es la desviación típica de los datos y z es el resultado ya estandarizado.

- **Normalización Máximos y Mínimos (Max-Min)**

Figura 11 - Normalización Max - Min

$$r = \frac{Y - Min}{Max - Min}$$

Fuente: Propia

Escala los valores entre un rango de 0 a 1.

Como se aprecia en la Figura 11, Y es el dato a estandarizar, Min es el mínimo valor de todo el set de datos, Max el máximo valor de todo el set de datos y r es el resultado estandarizado.

- **Normalización Sigmoide**

Figura 12 - Normalización Sigmoide

$$z = \frac{1}{1 + e^{-\frac{(x - \bar{a})}{\sigma}}}$$

Fuente: Propia

En la figura 12 se aprecia la formula donde x es el valor a estandarizar, \bar{a} es la media del conjunto de datos, σ es la desviación típica de los datos y z es el resultado ya estandarizado.

4. Entrenamiento, Prueba y Validación de datos

La práctica más usual por parte de los investigadores es dividir los datos en 3 grupos, datos de entrenamiento, datos de prueba y datos de validación. (Kaastra & Boyd, 1996)

- **Datos de Entrenamiento**

Los datos de entrenamiento es el grupo de datos más grande que sirve a la red neuronal para poder entrenarse y aprender los patrones presentes en los datos. (Kaastra & Boyd, 1996)

- **Datos de Prueba**

El conjunto de Prueba varía entre el 10% y el 30% de nuestro conjunto de entrenamiento, este conjunto de datos nos ayudara a poder evaluar que tanto puede generalizar de nuestra red neuronal. (Kaastra & Boyd, 1996)

- **Datos de Validación**

El conjunto de Validación nos servirá para poder verificar el rendimiento de nuestra red neuronal previamente entrenada, este conjunto de datos está compuesta por los últimos datos o más recientes del total de datos. (Kaastra & Boyd, 1996)

5. Paradigma de Redes Neuronales

Existen diversas maneras de poder construir una red neuronal por lo cual la neurodinámica y la arquitectura definen el paradigma de las redes neuronales. (Kaastra & Boyd, 1996)

La neurodinámica define las propiedades de la neurona como la función de activación que tendrá y las neuronas de entrada. (Kaastra & Boyd, 1996)

La arquitectura define la estructura que tendrá la red neuronal artificial como la cantidad de capas ocultas que tendrá y la cantidad de conexiones. (Kaastra & Boyd, 1996)

- **Número de Neurona de Entrada**

La cantidad de neuronas de entrada es algo sencillo debido a que en pasos anteriores se definió la cantidad de variables, cada variable representara una neurona de entrada. (Kaastra & Boyd, 1996)

- **Número de Capas Ocultas**

Las capas ocultas proporcionan a la red neuronal artificial la capacidad de poder generalizar, los investigadores han obtenido grandes resultados con una y dos capas ocultas, si en caso se desea aumentar la cantidad de capas ocultas se tiene que considerar el aumento del tiempo que utilizara nuestro procesador para poder realizar los cálculos. (Kaastra & Boyd, 1996)

Kaastra y Boyd recomiendan iniciar con una o dos capas ocultas.

- **Número de Neuronas Ocultas**

Si bien no se tiene una fórmula que nos de la cantidad de neuronas en la capa oculta los investigadores recurren mucho a la experimentación. (Kaastra & Boyd, 1996)

Sin embargo, se presentan algunos trabajos realizados por investigadores para cierto tipo de redes neuronales. (Kaastra & Boyd, 1996)

- Bailey y Thompsom sugieren que la cantidad de neurona en una red de 1 capa oculta debe ser el

75% del número de neuronas de entrada. (Bailey & Thompson, 1990)

- Katz sugiere que la cantidad de neuronas ocultas se encuentra entre el triple y la mitad del número de neuronas. (Katz, 1992)

- **Número de Neuronas de Salida**

La cantidad de neuronas de salida es más sencilla de hallar ya que depende del resultado que se desea obtener el investigador, esto quiere decir si se desea obtener un pronóstico el cual indique un conjunto de meses se necesitara la misma cantidad de neuronas de salida en su defecto si solo se desea pronosticar el mes siguiente solo hace falta una sola neurona. (Kaastra & Boyd, 1996)

- **Función de Transferencia**

Función de transferencia, también llamada función de activación o umbral no es más que una función matemática que determina el procesamiento de los datos de una neurona. (Kaastra & Boyd, 1996)

Existen varias funciones de activación tales como la función lineal, función tangente hiperbólica, función sigmoide, función escalonada, etc.

6. Criterios de Evaluación

El criterio de evaluación busca es medir el grado en el cual nuestra red neurona artificial a aprendido a generalizar, para esto los investigadores utilizan funciones de error las cuales nos mostrar cuál es el porcentaje de error que se obtenido con respecto a nuestros datos reales. (Kaastra & Boyd, 1996)

A continuación, se muestran algunos de las funciones de error:

- **Error Cuadrático medio**

Figura 13 - Error Cuadrático Medio

$$\frac{1}{n} \sum_{i=1}^n \left(\bar{y}_i - y_i \right)^2$$

Fuente: Propia

Calcula la media diferencial absoluta entre el dato real que está representado por \bar{y}_i y el dato obtenido por la red neuronal que está representado por y_i . (Ver figura 13)

- **Error Absoluto medio**

Figura 14 - Error Absoluto Medio

$$\frac{\sum_{i=1}^n |y_i - x_i|}{n}$$

Fuente: Propia

El cual calcula la media diferencial absoluta entre el dato real representado por y_i y el dato obtenido por la red neuronal que está representado por x_i . (Ver Figura 14)

- **Error Porcentual Absoluto medio**

Figura 15 - Error Porcentual Absoluto Medio

$$\frac{1}{n} \sum_{t=1}^n \left| \frac{A_t - F_t}{A_t} \right|$$

Fuente: Propia

Esta función calcula el error porcentual absoluto medio entre el dato de real que está representado por A_t y el dato obtenido por la red neuronal artificial que está representado por F_t . (Ver Figura 15)

7. Entrenamiento de la red neuronal

El entrenamiento de una red neuronal implica presentarle a la red neuronal artificial iterativamente ejemplos, el objetivo de esto es encontrar los pesos ideales que determinen el mínimo de la función de error. (Kaastra & Boyd, 1996)

- **Numero de Iteraciones de entrenamiento.**

El número de iteraciones representa la cantidad de veces que se le presentara a la red neuronal artificial los ejemplos con su respectivo resultado correcto. (Kaastra & Boyd, 1996)

Hay que tener en cuenta la cantidad de iteraciones debido a que a mayor cantidad de iteraciones nuestro computador va requerir más tiempo en realizar los cálculos. (Kaastra & Boyd, 1996)

- **Tasa de aprendizaje y Momentum.**

La tasa de aprendizaje o también conocida como la razón de aprendizaje no es más que tan rápido o tan lento converge la red neuronal artificial. (Kaastra & Boyd, 1996)

El Momentum o impulso permite suavizar las oscilaciones en la trayectoria hacia la convergencia. El momentum se encuentra acotado entre valores mayores a 0 pero menores que 1.

8. Implementación

Para la implementación se recopilará todos los datos antes mencionados para así poder diseñar la red neuronal artificial. (Kaastra & Boyd, 1996)

Se debe tener en cuenta el software que se utilizara para la elaboración ya sea el lenguaje de programación, el entorno de desarrollo y el hardware.

1.2. Definición de términos básicos

- a) **Inteligencia artificial:** Margaret Boden en su libro Artificial Intelligence and Natural Man, la define como el uso de programas de computación y de técnicas de programación que proyectan luz sobre principios de la inteligencia en general y la inteligencia humana en particular (Boden, 1984)

Para Barr y Feigenbaum en su libro The Handbook of Artificial Intelligence la definen como parte de la Informática que trata de imitar o emular la inteligencia humana diseñando sistemas de computadores inteligentes. (Barr & Feigenbaum, 1981)

- b) **Maching learning:** Machine Learning o aprendizaje automático, Según Tom Mitchell en su libro Maching Learning lo define como el estudio de algoritmos de computación que mejoran automáticamente su rendimiento y desempeño en tareas específicas gracias a la experiencia. (Mitchell, 1997)

Drew Conway mediante un diagrama de Venn (Ver anexo 3) define al machining learning como la unión de la matemática y la estadística con las habilidades informáticas. (Conway, 2010)

Andres Gonzales la define como una disciplina científica del ámbito de la Inteligencia Artificial que crea sistemas que aprenden automáticamente, esto quiere decir identificar patrones complejos en millones de datos. (González, 2019)

c) **Redes Neuronales Artificiales:** J. Hilera y V. Martínez en su libro *Redes Neuronales Artificiales: fundamentos, modelos y aplicaciones*; define a las redes neuronales como redes procesadoras de información cuya interacción depende del comportamiento del conjunto del sistema. (Hilera González & Martínez Hernando, 1994)

M. del Brío y S. Molina en su libro *Redes Neuronales y Sistemas Borrosos* la definieron como Sistemas que imitan el comportamiento del cerebro humano, mediante el aprendizaje y la extracción de conocimiento genérico a partir de datos. (Del Brío & Molina, 2001)

d) **Overfitting:** El overfitting o sobreajuste, este problema es muy común en las redes neuronales ya que lo que sucede es que la red neuronal aprende los casos particulares que se le mostro en la fase de entrenamiento y será incapaz de poder reconocer nuevos datos. Esto quiere decir que se le da mucha flexibilidad al modelo haciendo que se sobre ajuste a los datos de entrenamiento y que con datos nunca antes vistos generalice peor (Ver anexo 4). (Bagnato, 2017)

Es muy peligroso ya que conduciría fácilmente a predicciones erradas.

e) **Underfitting:** El underfitting o subajuste, Se define como la insuficiente complejidad del modelo para poder adaptarse a los datos de entrenamiento. Esto quiere decir red neuronal artificial no es lo suficientemente flexible para adaptarse a los datos, haciendo que nuestro modelo no generalice bien (Ver anexo 4). (Bagnato, 2017)

CAPÍTULO II: METODOLOGÍA DE DESARROLLO DEL TRABAJO PROFESIONAL

2.1. Delimitación temporal y espacial del trabajo

2.1.1 Temporal

El presente proyecto abarca una duración de 3 meses; empezando en Setiembre del 2020, y culminando en noviembre del 2020.

2.1.2 Espacial

Se realizará en el distrito de San Juan de Miraflores, en la ciudad de Lima, Perú.

2.2. Determinación y análisis del problema

El desarrollo del trabajo de Suficiencia profesional se llevó a cabo en el distrito de San Juan de Miraflores en el contexto de una pandemia Mundial llamada COVID-19 la cual aqueja y preocupa al distrito de San Juan de Miraflores.

A pesar de los esfuerzos realizados por las autoridades para poder controlar la cantidad de infectados esta siguió creciendo obteniendo un incremento del 234% del periodo correspondiente del 05/04/2020 - 11/04/2020 con respecto al periodo del 29/03/2020 – 04/04/2020.(Tabla 2)

Tabla 2 - Infectados por COVID-19

Periodo	Cantidad de Infectados
29/03/2020 - 04/04/2020	61
05/04/2020 - 11/04/2020	204
12/04/2020 - 18/04/2020	229
19/04/2020 - 25/04/2020	335
26/04/2020 - 02/05/2020	419
03/05/2020 - 09/05/2020	512
10/05/2020 - 16/05/2020	533
17/05/2020 - 23/05/2020	735

Fuente: (Ministerio de Salud - MINSA, 2020)

Debido al gran incremento de contagios los implemento para los médicos que atienden a los infectados y el personal asistencial es muy escasa. (Dr. Valverde, 2020)

Con lo cual el propio personal de salud se ve contagiado debido a la falta de implementos de protección de personal como se muestra en la tabla 3.

Tabla 3 - Personal de Salud Infectados por COVID-19 al 20/03/2020

Profesión/ocupación	EsSalud	Minsa	PNP	Privado	Otro	Total	%
Médico residente	4	3	1			8	33,33
Médico cirujano	2	3			2	7	29,17
Enfermera	1		1		1	3	12,50
Médico especialista		1	1			2	8,33
Obstetra	1					1	4,17
Odontóloga			1			1	4,17
Psicóloga					1	1	4,17
Técnica de enfermería	1					1	4,17
Total general	9	7	2	2	4	24	100,00

Fuente: (De La Cruz Vargas, 2020)

El distrito de San Juan de Miraflores al igual que los demás distritos no cuentan con pronóstico que les permita visualizar el crecimiento de contagios por Covid-19.

Esto dificulta la distribución los productos necesarios para la atención de los pacientes infectados por este virus.

2.3. Modelo de solución propuesto

La metodología que se realizó para este trabajo de suficiencia profesional fue basada en la investigación realizada por Kaastra y Boyd en el año de 1996.

2.3.1 Paso 1: Selección de la Variable

Para este proyecto la variable a predecir es la cantidad de personas infectadas por el Covid-19 de tipo cuantitativa ya que representa la

cantidad de personas infectadas por este virus semanalmente en el distrito de san juan de Miraflores.

2.3.2 Paso 2: Recolección de Datos

Los datos obtenidos para realizar este trabajo fueron proporcionados por el MINSA (Ministerio de Salud - MINSA, 2020), estas son mediciones de la cantidad de personas Infectadas por Covid-19 en los distintos departamentos del Perú de los cuales se extrajo del departamento de Lima el distrito de San Juan de Miraflores, este periodo está comprendido desde el 29 de marzo hasta el 12 de Setiembre del Presente Año

2.3.3 Paso 3: Pre - procesamiento de datos

Para tener un óptimo desempeño de nuestra red neuronal artificial se optó primero por agrupar los datos por semanas obteniendo 24 semanas (Ver Anexo 5) y luego se estandarizo los datos con la estandarización de máximos y mínimos. Para poder volver los datos a su estado original se aplicará la fórmula de des estandarización. (Ver Anexo 6)

2.3.4 Paso 4: Entrenamiento, Prueba y Validación de datos.

En este paso se realiza la división de todos los datos obtenidos en tres conjuntos.

Conjunto de Entrenamiento.

El conjunto de datos de entrenamiento representa el alrededor del 75% del total de Registros obtenidos esto corresponde a la cantidad de 18 semanas.

Conjunto de Prueba

El conjunto de Prueba está representado alrededor del 27% del total de Registros de entrenamiento esto corresponde a la cantidad de 5 semanas

Conjunto de Validación.

El conjunto de Validación está representado alrededor del 25% del total de Registros utilizados para el entrenamiento esto corresponde a la cantidad de 6 los cuales son las últimas semanas.

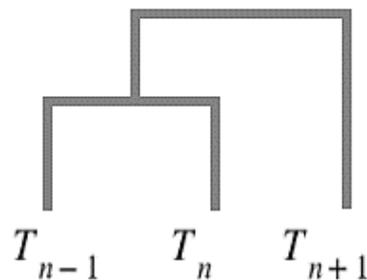
2.3.5 Paso 5: Paradigma de Redes Neuronales

Este paso se define la arquitectura de la red neuronal artificial que se empleará para nuestro estudio, se definirá el número de neuronas para la capa de entrada, el número de capas ocultas, el número de Neuronas ocultas y el número de Neuronas de Salida y la función de activación que se utilizará.

Numero de Neuronas de Entrada

El número de neuronas de la capa de entrada para este estudio son 2 neuronas.

Figura 16 - Línea de Tiempo



Fuente: Propia

En la Figura 16 se puede apreciar que con los datos T_{n-1} y T_n que representa a las 2 semanas posteriores podremos pronosticar cómo será el resultado en una semana a futuro que está representada por T_{n+1} .

Numero de Capas Ocultas

Para este estudio fue suficiente con una capa oculta ya que con solo una capa nuestra red neuronal artificial fue capaz de generalizar nuestros datos.

Numero de Neuronas de la capa Oculta

Para este estudio fue necesario 4 neuronas en la capa oculta ya que con esto la red neuronal artificial tuvo una buena capacidad de pronóstico.

Numero de Neuronas de Salida

Se tomó solo una neurona en la capa de salida ya que con solo una neurona en la capa de salida es suficiente para el pronóstico que se requiere para este estudio.

Función de Transferencia

La función de activación con el que se obtuvo una buena predicción fue la función de Activación sigmoide tanto en la capa oculta como en la capa de salida.

2.3.6 Paso 6: Criterios de Evaluación.

Para poder evaluar la eficiencia de nuestra red neuronal artificial para realizar un pronóstico, se toma como criterio de evaluación el error cuadrático medio.

2.3.7 Paso 7: Entrenamiento de la Red Neuronal

En este paso lo que definiremos es la cantidad de iteraciones, la razón de aprendizaje y el momentum que necesitara la red neuronal artificial con el objetivo de encontrar el conjunto de pesos que minimicen la función de error.

Numero de Iteraciones

La cantidad de Iteraciones que mejor se acoplo y que mejores resultados dio para el entrenamiento de la red neuronal Artificial fue de 200 iteraciones.

Razón de Aprendizaje y Momentum

La razón de aprendizaje se definió con una cantidad de 0.6 y el momentum se definió con una cantidad de 0.01 ya que con esto se obtuvo buenos resultados.

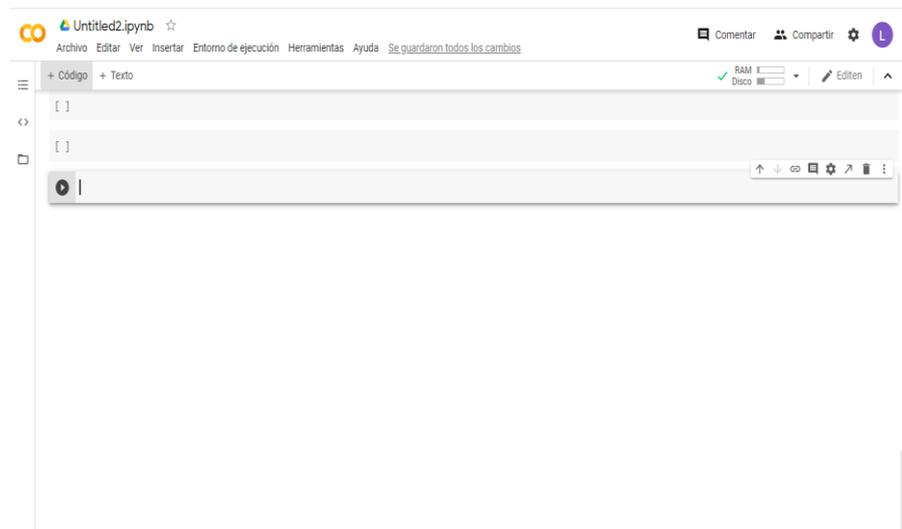
2.3.8 Paso 8: Implementación.

Para el desarrollo de esta red neuronal se utilizaron las siguientes herramientas

- Se utiliza Google Colab
- Python
- Librerías Numpy, Keras

Google Colab es una plataforma en la nube como se aprecia en la figura 17, te permite ejecutar código Python en tu navegador favorito, puedes acceder a la GPU gratuita que te ofrece Google.

Figura 17 - Google Colab



Fuente: Propia

Python es el lenguaje que se utilizó para poder desarrollar esta red neuronal con ayuda de la librería numpy que permitió ampliar las funciones matemáticas, y la librería Keras que cuenta con diversos tipos de algoritmos desde clasificación, regresión, etc. Como se aprecia en la figura 18.

Figura 18 - Python, Numpy, Keras



Fuente: Propia

Lo primero que realizaremos es la importación de las librerías numpy y Keras, luego se cargara todo el Set de datos que se obtuvo de la página del MINSA como una lista de [1 X 24].

Figura 19 - Importación de Librerías y Carga de Datos

```
[5] import numpy as np
import tensorflow.keras as kr

#Set de Datos
Datos_B = [61,204,229,335,419,512,533,735,
650,423,516,550,336,279,293,456,
592,607,681,752,916,641,596,574]
```

Fuente: Propia

En la Figura 19 se aprecia la importación de las librerías las cuales le pondremos un alias a **numpy** como **np** y a **keras** como **kr**.

Con los datos obtenidos por el Minsa diariamente procederemos a acumularlos por semanas obteniendo 24 semanas. (Ver Anexo 5)

A partir de ello procederemos a particionar nuestro set de datos en datos de entrenamiento, datos de Test y datos de validación como se explicó líneas arriba se particionará de la siguiente manera. (Figura 20)

- Datos Entrenamiento: 18
- Datos Test: 5 registros
- Datos Validación: 6 registros

Figura 20 - Particionamiento de Datos

```
Set_Datos_Entrenamiento=[61,204,229,335,419,512,533,735,650,
                          423,516,550,336,279,293,456,592,607]

Set_Datos_Test= [279,293,456,592,607]

Set_Datos_Validacion = [681,752,916,641,596,574]
```

Fuente: Propia

Una vez ya tengamos particionado nuestros datos realizaremos la estandarización de nuestros datos.

Figura 21 - Estandarización de Datos

```
d_maxi = max(Datos_B)
d_min = min(Datos_B)

def transformacion_Datos (datos):
    transf_datos = []
    for i in datos:
        transf_datos.append((i-d_min)/(d_maxi-d_min))
    return transf_datos

Datos_Entrena_Proc = transformacion_Datos(Set_Datos_Entrenamiento)
Datos_Test_Proc = transformacion_Datos(Set_Datos_Test)
Datos_Valida_Proc = transformacion_Datos(Set_Datos_Validacion)
```

Fuente: Propia

Como se aprecia en la Figura 21, primero obtendremos el Máximo y Mínimo valor de todo el set de datos para esto nos ayudaremos con la función **max** y **min**, almacenaremos estos valores en **d_maxi** el valor máximo y **d_min** el valor mínimo; luego crearemos una función llamada **transformacion_Datos** esta nos ayudara a realizar la conversión solicitando como parámetro los datos a estandarizar, dentro de esta función se creara una lista llamado **transf_datos** el cual almacenara nuestros datos ya procesados, se creara un bucle **for** el cual ira iterando de acuerdo a la cantidad de datos que se enviaron a la función como parámetro, dentro de este bucle nos apoyaremos con el método **append** el cual agregara al final de nuestra lista el cual será retornado por la función **transformacion_Datos**.

Los datos ya estandarizados se almacenarán en las siguientes variables.

Datos_Entrena_Proc = Datos estandarizados de Entrenamiento.

Datos_Test_Proc = Datos estandarizados de Test.

Datos_Valida_Proc = Datos estandarizados de Validación.

Figura 22 - Datos de Entrada y Salida

```
def Ordenamiento_datos(datos):
    E=[]
    for i in range(0,len(datos)-2):
        E.append([datos[i],datos[i+1],datos[i+2]])
    return E

Datos_entrenamiento= np.array(Ordenamiento_datos(Datos_Entrena_Proc))
X = Datos_entrenamiento[:,0:2]
Y = Datos_entrenamiento[:,2]
```

Fuente: Propia

En la Figura 22 se realizó la separación de nuestros datos de entrada o input y nuestros datos de salida u output de todo el set de Entrenamiento, como se había mencionado los parámetros de

entrada serian dos ya que con dos semanas posteriores podremos predecir lo que sucederá la semana siguiente. Para esto realizamos la función **Ordenamiento_datos** que solicitara como parámetro de entrada los datos que va procesar, dentro de esta función generaremos una lista llamada **“E”** en la cual agregaremos los datos con ayuda de un bucle **for** el cual recorrerá todos los datos hasta el antepenúltimo datos ya que estos se agregaran de 3 en 3, donde los 2 primeros datos indicaran el input y el último dato será el output.

Utilizaremos la librería numpy la cual nos ayudara a crear un array con los datos procesados, los cuales serán almacenados en la variable **“Datos_”**, estos datos almacenados se particionarán en la variable **X** que tendrá los datos de entra (Input) y la variable **Y** los datos de salida (output).

Luego de esto crearemos unas variables las cuales almacenaran la razón de aprendizaje, la estructura de nuestra red neuronal y la cantidad de épocas.

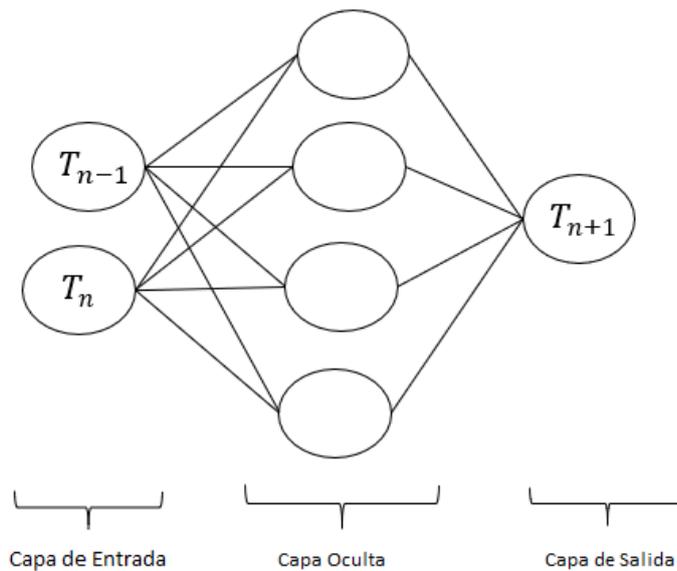
Figura 23 - Variables de la Red Neuronal

```
lr= 0.6  
red=[2,4,1]  
epocas = 200
```

Fuente: Propia

Como se puede apreciar en la figura 23, la razón de aprendizaje se almacena en la variable **“lr”** la cual tiene un valor de 0.6; la estructura de nuestra red se almacena en la variable **“red”** la cual está comprendida por 2 neuronas de entrada, 4 neuronas en la capa oculta y 1 neurona en la capa de salida (Ver Figura 24) y por último la cantidad de épocas se almacenará en la variable **“épocas”** que tiene un valor de 200.

Figura 24 - Estructura de la Red Neuronal Artificial



Fuente: Propia

Ahora realizaremos la creación de las capas de nuestra red neuronal, para esto utilizaremos la librería keras la cual utilizando su modelo secuencial crearemos capas para nuestra red neuronal artificial.

Figura 25 - Creación de Capas

```
model=kr.Sequential()

#capa1
model.add(kr.Input(shape=red[0]))
#capa2
model.add(kr.layers.Dense(red[1],activation='sigmoid'))
#capa3
model.add(kr.layers.Dense(red[2],activation='sigmoid'))
```

Fuente: Propia

Como se aprecia en la Figura 25 creamos una variable llamada “**model**” en la cual almacenaremos las capas de la red neuronal artificial.

Crearemos tres capas, la primera capa corresponderá a la capa de entrada que está compuesta por las 2 neuronas, la segunda capa corresponde a la capa oculta de la red neuronal artificial la cual está compuesta por 4 neuronas, la tercera capa corresponde a la capa de salida con solo una neurona, las capas ocultas y la capa de salida tendrán una función de activación para nuestro caso la función de activación será la función sigmoide.

Figura 26 - Configuración de la Red Neuronal

```
model.compile(optimizer=kr.optimizers.SGD(
    learning_rate=lr,
    momentum=0.01),
    metrics="mean_squared_error")
```

Fuente: Propia

Una vez se haya realizado la creación de capas realizaremos la configuración de la red neuronal artificial para eso **keras** nos pedirá los siguientes parámetros; **metrics** indica que función se utilizara para juzgar el rendimiento de la red neuronal artificial, para nuestro estudio se utilizará el error cuadrático medio que equivale a “**mean_squared_error**” el otro indicador es el optimizador que está representado por el parámetro **optimizer** que no es más que el algoritmo con el cual nos ayudara a minimizar la función de error utilizaremos el algoritmo del descenso del gradiente que equivale a decir “**SGD**” pasando como argumentos la razón de aprendizaje y el momentum que ya se había definido.(Ver Figura 26)

Ahora pasaremos a realizar el entrenamiento de nuestra red neuronal para esto utilizaremos el método **fit()** el cual nos solicitara tres parámetros.

Figura 27 - Entrenamiento de la Red

```
model.fit(X,Y,epochs=epocas)
```

Fuente: Propia

En la Figura 27 se aprecia que el método **fit()** nos solicita 3 parámetros el primer parámetro corresponde a los datos de entrada de entrenamiento en nuestro caso nuestro dato de entrada para entrenar a la red neuronal artificial están almacenados en “**X**”, el segundo parámetro son los datos de salida los cuales están almacenados en “**Y**” y por el ultimo el parámetro de épocas la cual se definió en la variable “**epocas**” que corresponde a 200.

Ahora luego de ejecutar nuestro código veremos los resultados del entrenamiento de la red neuronal artificial

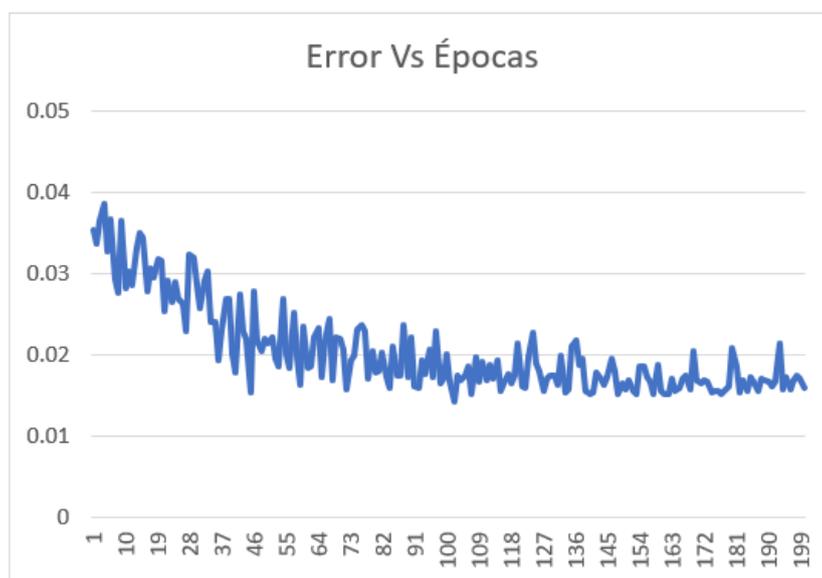
Figura 28 - Resultado de Entrenamiento

```
Epoch 186/200
8/8 [=====] - 0s 3ms/step - mse: 0.0167
Epoch 187/200
8/8 [=====] - 0s 3ms/step - mse: 0.0167
Epoch 188/200
8/8 [=====] - 0s 3ms/step - mse: 0.0171
Epoch 189/200
8/8 [=====] - 0s 3ms/step - mse: 0.0168
Epoch 190/200
8/8 [=====] - 0s 3ms/step - mse: 0.0167
Epoch 191/200
8/8 [=====] - 0s 3ms/step - mse: 0.0186
Epoch 192/200
8/8 [=====] - 0s 3ms/step - mse: 0.0167
Epoch 193/200
8/8 [=====] - 0s 3ms/step - mse: 0.0214
Epoch 194/200
8/8 [=====] - 0s 3ms/step - mse: 0.0158
Epoch 195/200
8/8 [=====] - 0s 3ms/step - mse: 0.0172
Epoch 196/200
8/8 [=====] - 0s 3ms/step - mse: 0.0158
Epoch 197/200
8/8 [=====] - 0s 4ms/step - mse: 0.0168
Epoch 198/200
8/8 [=====] - 0s 3ms/step - mse: 0.0167
Epoch 199/200
8/8 [=====] - 0s 3ms/step - mse: 0.0170
Epoch 200/200
8/8 [=====] - 0s 3ms/step - mse: 0.0159
<tensorflow.python.keras.callbacks.History at 0x7f414d108e48>
```

Fuente: Propia

Como se aprecia en la Figura 28 nuestro error va disminuyendo con el pasar del tiempo logrando un error del 1.59% de error en 200 épocas. El aprendizaje termino debido a que se alcanzó le máximo de épocas.

Figura 29 - Grafico de Error Vs Épocas



Fuente: Propia

En la Figura 29 se aprecia cómo se va reduciendo el error con respecto a las épocas.

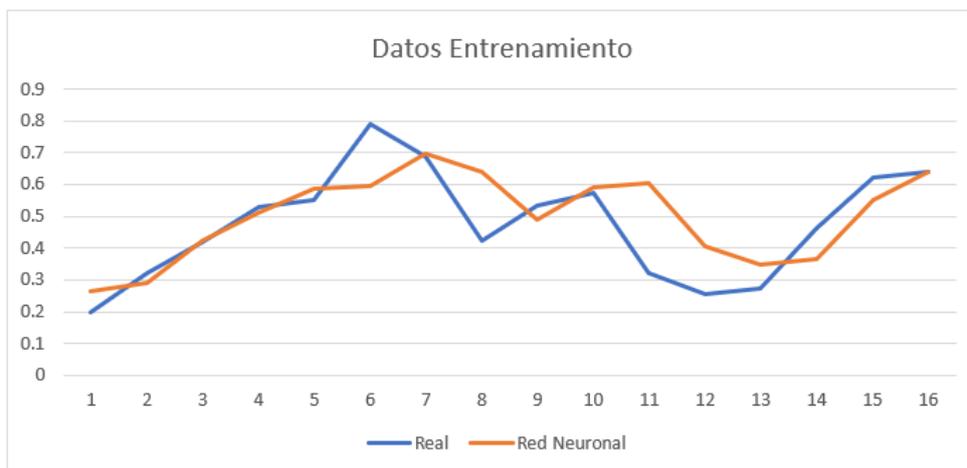
Tabla 4 - Datos de Entrenamiento

Infected Real	Infected Transformed	Neural Network Forecast	Error
229	0.19649123	0.2659732	-0.06948197
335	0.32046784	0.2909992	0.02946864
419	0.41871345	0.42270875	-0.0039953
512	0.52748538	0.51113147	0.01635391
533	0.55204678	0.5875455	-0.03549872
735	0.78830409	0.59444	0.19386409
650	0.68888889	0.6984553	-0.00956641
423	0.42339181	0.64169526	-0.21830345
516	0.53216374	0.49144495	0.04071879
550	0.57192982	0.5902425	-0.01831268

336	0.32163743	0.6058451	-0.28420767
279	0.25497076	0.4059399	-0.15096914
293	0.27134503	0.3475227	-0.07617767
456	0.4619883	0.3675238	0.0944645
592	0.62105263	0.5510943	0.06995833
607	0.63859649	0.63824093	0.00035556

Fuente: Propia

Figura 30 - Grafico de los Datos de Entrenamiento



Fuente: Propia

Como se aprecia en la Tabla 4 la cantidad de infectados se va asemejando a la cantidad que pronostica la red neuronal, el cual es graficado como se aprecia en la Figura 30.

Validación con los datos de Prueba

Separamos nuestros datos de prueba en dos, los datos de entrada los almacenaremos en la variable “**X1**” y los datos de salida que se almacenaran en la variable “**X2**” ya estandarizados para validar la capacidad de generalizar de nuestra red neuronal artificial como se aprecia en la Figura 31.

Figura 31 - Datos de Prueba

```
Datos_test= np.array(Ordenamiento_datos(Datos_test_Proc))
X1= Datos_test[:,0:2]
Y1= Datos_test[:,2]
```

Fuente: Propia

Ahora para evaluar a nuestra red neuronal artificial utilizaremos el siguiente método “**evaluate**” el cual nos pedirá dos parámetros de entrada los cuales son los datos de entrada que están en la variable “**X1**” y los datos de salida que están en la variable “**X2**” de nuestro set de datos de prueba este nos ayudara a ver el error que tiene nuestra red neuronal artificial como se aprecia en la Figura 32.

Figura 32 - Resultado con los Datos de Prueba

```
model.evaluate(X1,Y1)
1/1 [=====] - 0s 2ms/step - mse: 0.0046
[0.004605945665389299]
```

Fuente: Propia

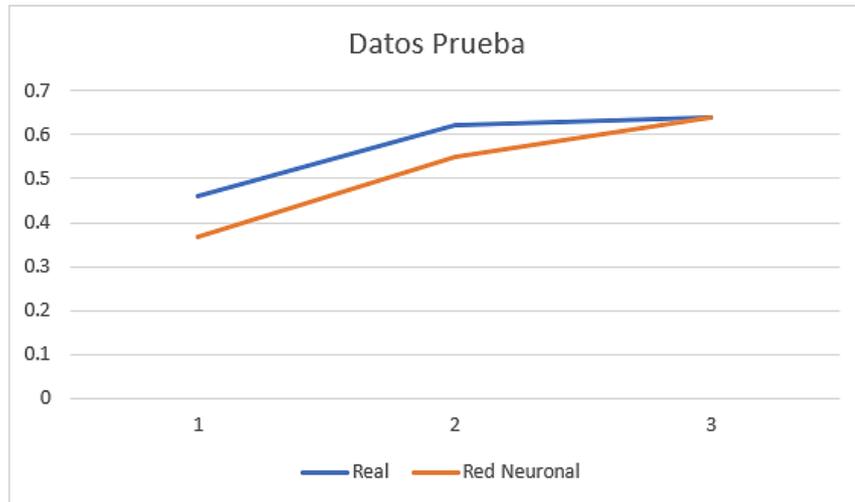
Como se visualiza en la Figura 32 el error que tenemos para pronosticar con los datos de prueba es del 0.46%.

Tabla 5 - Datos de Prueba

Infected Real	Infected Transformados	Prediction Red Neuronal	Error
456	0.4619883	0.3675238	0.0944645
592	0.62105263	0.5510943	0.06995833
607	0.63859649	0.63824093	0.00035556

Fuente: Propia

Figura 33 - Grafico de los Datos de Prueba



Fuente: Propia

Como se aprecia en la Tabla 5 la cantidad de infectados se va asemejando a la cantidad que pronostica la red neuronal, el cual es graficado como se aprecia en la Figura 33

Validación con los datos de Validación

Ahora validaremos el error que tiene nuestra red neuronal artificial con el set de datos de Validación.

Figura 34 - Resultados con los Datos de Validación

```
Datos_validacion= np.array(Ordenamiento_datos(Datos_Valida_Proc))
X2= Datos_validacion[:,0:2]
Y2= Datos_validacion[:,2]

model.evaluate(X2,Y2)

1/1 [=====] - 0s 1ms/step - mse: 0.0249
[0.024875285103917122]
```

Fuente: Propia

Como se aprecia en la Figura 34 el error de tenemos con los datos de validación es del 2.5%.

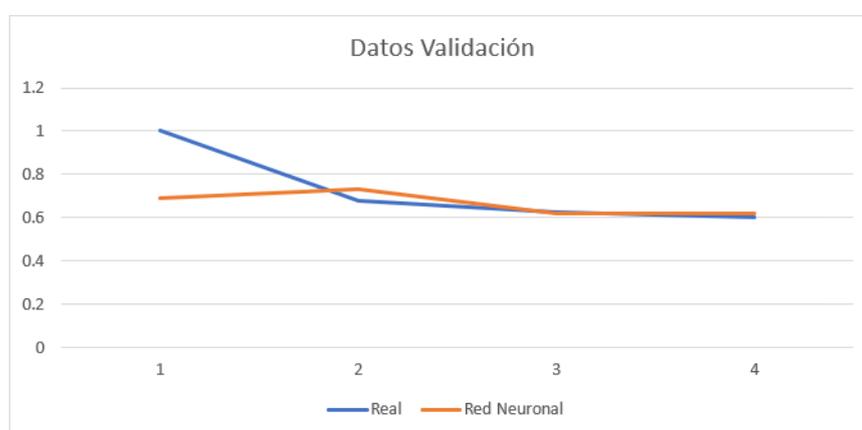
Tabla 6 - Datos de Validación

Infected Real	Infected Transformed	Neural Network Forecast	Error
916	1	0.69028866	0.30971134
641	0.67836257	0.7334943	-0.05513173
596	0.62573099	0.6195651	0.00616589
574	0.6	0.62241644	-0.02241644

Fuente: Propia

Como se aprecia en la Tabla 6 la cantidad de infectados se va asemejando a la cantidad que pronostica la red neuronal, el cual es graficado como se aprecia en la Figura 35.

Figura 35 - Grafico de Datos de Validación



Fuente: Propia

Pronóstico

Figura 36 – Pronóstico

```
X3=transformacion_Datos([596,574])
model.predict(X3)
array([[0.6136645]], dtype=float32)
```

Fuente: Propia

Para poder realizar un pronóstico utilizaremos el método llamado “predict” en el cual enviaremos como parámetros las dos últimas semanas de nuestro set de datos, estas dos semanas están

comprendidas desde el día 30 agosto al 12 de setiembre. (Ver Figura 36)

El cual nos da como resultado 0.6136645 el cual realizaremos la des estandarización (Ver Anexo 6) obteniendo como resultado 585 infectados para la semana correspondiente desde el 13 de Setiembre al 20 de Setiembre.

2.4. Resultados

Los resultados obtenidos satisfacen a los objetivos planteados.

- Se diseño la arquitectura de la red neuronal la cual ayudara a pronosticar la cantidad de contagios por Covid-19; compuesta por dos neuronas en la capa de entrada la cual corresponde a las dos semanas posteriores, una capa oculta compuesta por cuatro neuronas cada una de ellas con una función de activación sigmoide y una capa de salida compuesta por una neurona con una función de activación sigmoide.
- Se entreno a la red neuronal artificial obteniendo como resultado un porcentaje de error del 1.59%.
- Se realizo el pronóstico correspondiente a la siguiente semana correspondiente desde el 13 de Setiembre al 20 de Setiembre obteniendo como resultado la cantidad de 585 infectados por Covid-19.

CONCLUSIONES

- Se diseño la red neuronal artificial siguiente la metodología propuesta por Kaastra y Boyd, con la cual se obtuvo buenos resultado para el desarrollo de este proyecto; además se estandarizaron los datos para poder obtener mejores resultados en cuanto al aprendizaje la red neuronal artificial.
- Se realizo el pronóstico de la cantidad de infectados por Covid-19 para la siguiente semana correspondiente desde el 13 de Setiembre al 20 de Setiembre. Esto ayudara para la toma de decisiones de las autoridades.

RECOMENDACIONES

- Se recomienda a las autoridades de San Juan de Miraflores el uso de este proyecto para poder tomar decisiones ante esta pandemia llamada Covid-19.
- Se recomienda a las autoridades de Perú el uso de redes neuronal artificiales ya que es una gran herramienta de pronóstico.
- Para el desarrollo de una red neuronal tener en cuenta librerías gratuitas como keras, Tensorflow y Sklearn las cuales ayudan mucho en la creación de una red neuronal.

Bibliografía

- Dr. Valverde, J. (09 de Abril de 2020). Sin respiro: Unidades de Cuidados Intensivos necesitan 700 médicos. (F. Torres, Entrevistador)
- Alarcón Jaque, M. (2014). *MODELOS DE REDES NEURONALES APLICADO EN LA PREDICCIÓN DEL SIGNO DE LOS FONDOS DE AFP CUPRUM*. Santiago - Chile: Universidad de Chile.
- Bagnato, J. I. (12 de Diciembre de 2017). *Aprende Machine Learning*. Obtenido de Aprende Machine Learning: <https://www.aprendemachinelarning.com/que-es-overfitting-y-underfitting-y-como-solucionarlo/>
- Bailey, D. L., & Thompson, D. M. (1990). *Developing neural-network applications*. Al Expert.
- Barr, A., & Feigenbaum, E. A. (1981). *The Handbook of artificial Intelligence*. Los Altos (California): Elsevier Inc.
- Boden, M. A. (1984). *Inteligencia artificial y hombre natural*. Madrid: Madrid Tecnos D.L.
- Calvo, D. (8 de Diciembre de 2018). *Diego Calvo*. Obtenido de Diego Calvo: <https://www.diegocalvo.es/perceptron-multicapa/>
- Calla Charrez, V. (2019). *Modelo de pronóstico para número de egresos hospitalarios usando redes neuronales – Hospital Regional Manuel Núñez Butrón – Puno, 2009 – 2018*. Puno.
- Carrasquilla L, J. d. (1993). Contagio e infección. *Revista de la Facultad de Medicina - Universidad Nacional de Colombia (Sede Bogotá)*, 827-830.
- Centro Nacional de Epidemiología, Prevención y Control de Enfermedades. (29 de Agosto de 2020). *Ministerio de Salud*. Obtenido de Ministerio de Salud: <https://www.dge.gob.pe/portal/docs/tools/coronavirus/coronavirus290820.pdf>
- Centro para el Control y la Prevención de Enfermedades. (16 de julio de 2020). *Centro para el Control y la Prevención de Enfermedades*. Obtenido de Centro para el Control y la Prevención de Enfermedades: <https://espanol.cdc.gov/coronavirus/2019-ncov/prevent-getting-sick/how-covid-spreads.html#:~:text=C%C3%B3mo%20se%20propaga%20el%20COVID%2D19&text=Se%20cree%20que%20el%20COVID,s%C3%ADntomas%20pueden%20propagar%20el%20virus>.
- Conway, D. (2010). *Drew Conway Data Consulting*. Obtenido de Drew Conway Data Consulting: https://s3.amazonaws.com/aws.drewconway.com/viz/venn_diagram/data_science.html
- Cueva Rios, P. (2019). *Evaluación de índices de confiabilidad en redes de Electrocentro S.A. para mejorar la calidad de servicio utilizando redes neuronales*. Lima.
- De La Cruz Vargas, J. A. (2020). *Protegiendo al personal de la salud en la pandemia Covid-19*. Lima: Instituto de Investigación en Ciencias Biomédicas, Universidad Ricardo Palma, Lima -Perú. Obtenido de http://www.scielo.org.pe/scielo.php?pid=S2308-05312020000200173&script=sci_arttext
- Del Brío, B. M., & Molina, A. S. (2001). *Redes neuronales y sistemas borrosos*. Madrid: Ra-Ma.

- Farrera Gutiérrez, A. (2013). *Manual de Pronósticos para la toma de decisiones*. Monterrey - Mexico: Editorial Digital Tecnológico de Monterrey.
- Flores Castillo, L. A., & Ramales Osorio, M. C. (2017). *Redes neuronales artificiales: efecto asimetría y curtosis en la evaluación de portafolios de inversión*. México.
- Flores, D., Gómez, C., Cervantes, D., Abaroa, A., Castro, C., & Castalleda Martínez, R. (2017). Prediciendo la Actividad Cardíaca de la Almeja *Tivela stultorum* con Digoxina Utilizando Redes Neuronales Artificiales. *Revista Mexicana de Ingeniería BioMedica*, 208-216.
- Fumero, A., & Navarrete, G. (2016). Personalidad y Malestar Psicológico: Aplicación de un Modelo de Redes. *Revista Iberoamericana de Diagnóstico y Evaluación - e Avaliação Psicológica*, 28-38.
- García, U. (24 de Enero de 2020). *Medium*. Obtenido de Medium: <https://medium.com/futurelabmx/introducci%C3%B3n-a-las-redes-neuronales-pt-ia73f87933f8e>
- Gil Vera, V. D. (2015). *Forecasting Electricity Demand for Small Colombian Populations*. Colombia.
- Gómez, J., Sánchez, J., Ocampo, A., & Restrepo, J. W. (2012). APLICACIÓN DE REDES NEURONALES EN LA CLASIFICACIÓN DE ARCILLAS. *Revista EIA*, 183-191.
- González, A. (12 de 10 de 2019). *Cleverdata*. Obtenido de Cleverdata: <https://cleverdata.io/ques-machine-learning-big-data>
- Hilera González, J. R., & Martínez Hernando, V. J. (1994). *Redes neuronales artificiales: fundamentos, modelos y aplicaciones*. España: RA-MA S.A. Editorial y Publicaciones.
- Instituto Nacional de Salud y Centro Nacional de Epidemiología, Prevención y Control de Enfermedades. (21 de 08 de 2020). *Sala Situacional Covid-19 Perú*. Obtenido de Sala Situacional Covid-19 Perú: https://covid19.minsa.gob.pe/sala_situacional.asp
- Isasi Viñuela, P., & Galván León, I. M. (2004). *Redes Neuronales Artificiales un enfoque practico*. Madrid - España: Person Education S.A.
- Kaastra, I., & Boyd, M. (1996). Designing a neural network for forecasting financial. *Neurocomputing*, 217.
- Katz, J. O. (1992). *Developing Neural Network Forecasters for Trading*. Technical Analysis of Stocks and Commodities.
- Licona, A. E. (2001). Definición de una red neuronal para clasificación. *Revista Mexicana de Ingeniería Biomedica*, 5.
- Linn, A. (14 de 03 de 2018). *Microsoft*. Obtenido de Microsoft: <https://blogs.microsoft.com/ai/chinese-to-english-translator-milestone/>
- Malaver Elera, M. A. (2015). *APLICACIÓN DE REDES NEURONALES PARA DETERMINAR EL PRONÓSTICO DE LAS VENTAS EN LA EMPRESA CATERING & BUFFETS MyS UBICADA EN LA CIUDAD DE PIURA*. Piura: universidad Cesar Vallejo.
- Mateus, S. P., González, N., & Branch, J. W. (2014). *Aplicación de Redes Neuronales Artificiales en Entornos*. Medellín - Colombia: Universidad Nacional de Colombia.

- Matich, D. J. (02 de 03 de 2001). *Facultad Regional del Rosario - Universidad Tecnológica Nacional*. Obtenido de Facultad Regional del Rosario - Universidad Tecnológica Nacional: https://www.frro.utn.edu.ar/repositorio/catedras/quimica/5_ano/orientadora1/monografias/matich-redesneuronales.pdf
- Matich, D. J. (12 de 03 de 2001). *Facultad Regional Rosario - Universidad Tecnológica Nacional*. Obtenido de Facultad Regional Rosario - Universidad Tecnológica Nacional: https://www.frro.utn.edu.ar/repositorio/catedras/quimica/5_ano/orientadora1/monografias/matich-redesneuronales.pdf
- MAYO CLINIC. (02 de Setiembre de 2020). *MAYO CLINIC*. Obtenido de MAYO CLINIC: <https://www.mayoclinic.org/es-es/diseases-conditions/coronavirus/symptoms-causes/syc-20479963>
- McCulloch, W. S., & Pitts, W. (1988). A logical calculus of the ideas immanent in nervous activity. *Neurocomputing*, 15-27.
- Ministerio de Salud - MINSa. (27 de 05 de 2020). *Datos Abiertos Minsa y Gestión del Conocimiento en Covid-19*. Obtenido de Datos Abiertos Minsa y Gestión del Conocimiento en Covid-19: <https://www.datosabiertos.gob.pe/dataset/casos-positivos-por-covid-19-ministerio-de-salud-minsa/resource/690e57a6-a465-47d8-86fd>
- Ministerio de Salud. (12 de 05 de 2020). *Diris Lima Sur*. Obtenido de Diris Lima Sur: <https://www.dirislimasur.gob.pe/2020/05/12/mas-del-50-de-comerciantes-del-mercado-ciudad-de-dios-resultaron-positivo-al-covid-19/>
- MINSa. (08 de Mayo de 2020). *Ministerio de Salud*. Obtenido de Ministerio de Salud: <https://www.dge.gob.pe/portal/docs/tools/coronavirus/analisiscoronavirus080520.pdf>
- MINSa. (8 de Agosto de 2020). *Ministerio de Salud*. Obtenido de Ministerio de Salud: [http://portal.susalud.gob.pe/wp-content/uploads/archivo/registro-camas/2020/agosto/08/20200808_turnotarde_Reporte%20Ejecutivo%20-%20Comando%20COVID-19%20-%20F500.2%20\(Camas\)-%20ZC%20LM%20y%20Callao%20\(4\).pdf](http://portal.susalud.gob.pe/wp-content/uploads/archivo/registro-camas/2020/agosto/08/20200808_turnotarde_Reporte%20Ejecutivo%20-%20Comando%20COVID-19%20-%20F500.2%20(Camas)-%20ZC%20LM%20y%20Callao%20(4).pdf)
- MINSa. (03 de Julio de 2020). *Plataforma Nacional de Datos Abiertos*. Obtenido de Plataforma Nacional de Datos Abiertos: <https://www.datosabiertos.gob.pe/dataset/casos-positivos-por-covid-19-ministerio-de-salud-minsa>
- Minsky, M. L., & Papert, S. (1988). *Perceptrons*. Cambridge: Massachusetts Institute of Technology.
- Mitchell, T. (1997). *Machine Learning*. New York, EE. UU: McGraw-Hill.
- MORENO, J. J. (2002). *Redes Neuronales Artificiales aplicadas al Análisis de Datos*. PALMA DE MALLORCA.
- Municipal Distrital de San Juan de Miraflores. (22 de 08 de 2020). *Sitio Oficial de la Municipal Distrital de San Juan de Miraflores*. Obtenido de Sitio Oficial de la Municipal Distrital de San Juan de Miraflores: <https://munisjm.gob.pe/historia/>
- Oficina General de Estadística e Informática. (15 de 12 de 2005). *MINSa*. Obtenido de MINSa: <http://www.minsa.gob.pe/estadisticas/estadisticas/boletines/infraestructurasanitaria.pdf>

- OMS. (10 de Setiembre de 2020). *Organización Mundial de la Salud*. Obtenido de Organización Mundial de la Salud: <https://www.who.int/es/emergencias/diseases/novel-coronavirus-2019/advice-for-public/q-a-coronaviruses>
- Organización Mundial de la Salud. (10 de Setiembre de 2020). *Organización Mundial de la Salud*. Obtenido de Organización Mundial de la Salud: <https://www.who.int/es/emergencias/diseases/novel-coronavirus-2019>
- Pérez rivera, C. A., Britto Montoya, J. A., & Isaza Echeverry, G. A. (2005). *APLICACIÓN DE REDES NEURONALES PARA LA DETECCIÓN DE INTRUSOS EN*. Pereira - Colombia: Universidad Tecnológica de Pereira.
- Ramos, O. L., Rojas, D. A., & Saby, J. E. (2016). Reconocimiento de Patrones Vocálicos mediante la implementación de una red Neuronal Artificial Utilizando Sistemas Embebidos. *Inf. tecnol*, 133-142.
- Rivas, A. (30 de Abril de 2020). *Colegio Medicos del Peru*. Obtenido de Colegio Medicos del Peru: <https://www.cmp.org.pe/cmp-envia-carta-a-fiscal-de-la-nacion-para-que-se-exija-cumplimiento-de-entrega-de-epp-a-medicos-y-personal-de-salud/>
- Rojas Díaz, A. (25 de 10 de 2010). *unavdocs*. Obtenido de unavdocs: <https://unavdocs.files.wordpress.com/2010/10/pronosticodecostos.pdf>
- Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*. *Psychological Review*, 386–408.
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature* 323, 533–536.
- Tablada, C. J., & Torres, G. A. (2009). Redes Neuronales Artificiales. *Revista de Educacion Matemática*, 30.
- Villada, F., Muñoz, N., & García-Quintero, E. (2016). *Redes Neuronales Artificiales aplicadas a la Predicción del*. Medellin - Colombia: Universidad de Antioquia.
- Villarreal, F. (20 de 09 de 2016). *Departamento de Matemática - Universidad Nacional del Sur*. Obtenido de Departamento de Matemática - Universidad Nacional del Sur: https://www.matematica.uns.edu.ar/uma2016/material/Introduccion_a_los_Modelos_de_Pronosticos.pdf
- Waymo. (9 de 3 de 2018). *Waymo*. Obtenido de Waymo: <https://medium.com/waymo/same-driver-different-vehicle-bringing-waymo-self-driving-technology-to-trucks-e55824b55b8f>
- Widrow, B., & Hoff, M. E. (1960). Adaptive Switching Circuits. *IRE WESCON Convention Record*, 96-104.

ANEXOS

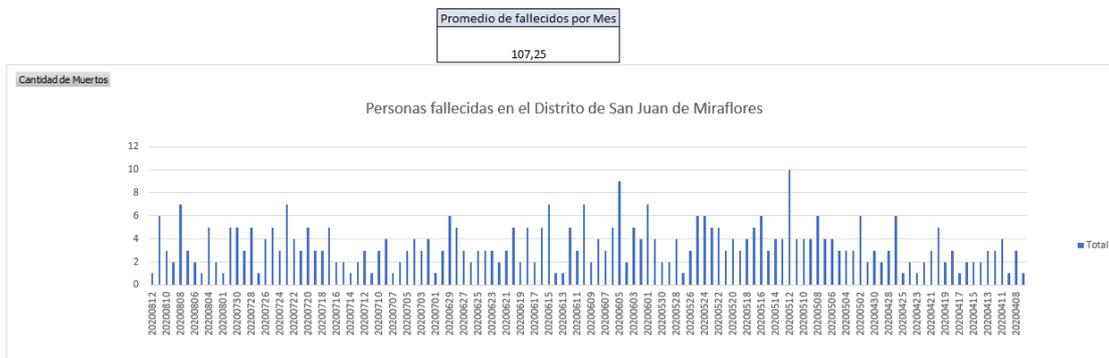
Anexo 1: Cantidad de Camas de Hospitalización en los distritos de Lima Metropolitana

OFERTA DE CAMAS DE HOSPITALIZACIÓN ZONA COVID-19 A NIVEL DISTRITOS DE LIMA METROPOLITANA

Districtos	IPRESS	En uso	Disponible	Total
	73	818	0	818
CARMEN DE LA LEGUA-REYNOSO	1	0	0	0
MAGDALENA DEL MAR	1	0	0	0
PUENTE PIEDRA	1	0	0	0
SANTA ANITA	2	0	0	0
VENTANILLA	2	0	0	0
CHORRILLOS	1	4	0	4
LA MOLINA	1	4	0	4
LURIGANCHO	3	4	0	4
COMAS	2	8	0	8
INDEPENDENCIA	1	8	0	8
SAN JUAN DE LURIGANCHO	3	10	0	10
VILLA MARIA DEL TRIUNFO	1	14	0	14
EL AGUSTINO	1	17	0	17
PUEBLO LIBRE	4	19	0	19
MIRAFLORES	4	21	0	21
SAN BORJA	5	22	0	22
SANTIAGO DE SURCO	2	22	0	22
SAN MIGUEL	4	25	0	25
SAN JUAN DE MIRAFLORES	2	27	0	27
SAN ISIDRO	5	33	0	33
CALLAO	2	41	0	41
SAN MARTIN DE PORRES	3	43	0	43
VILLA EL SALVADOR	3	50	0	50
LA VICTORIA	2	52	0	52
BELLAVISTA	3	53	0	53
LIMA	7	88	0	88
ATE	4	100	0	100
JESUS MARIA	3	153	0	153

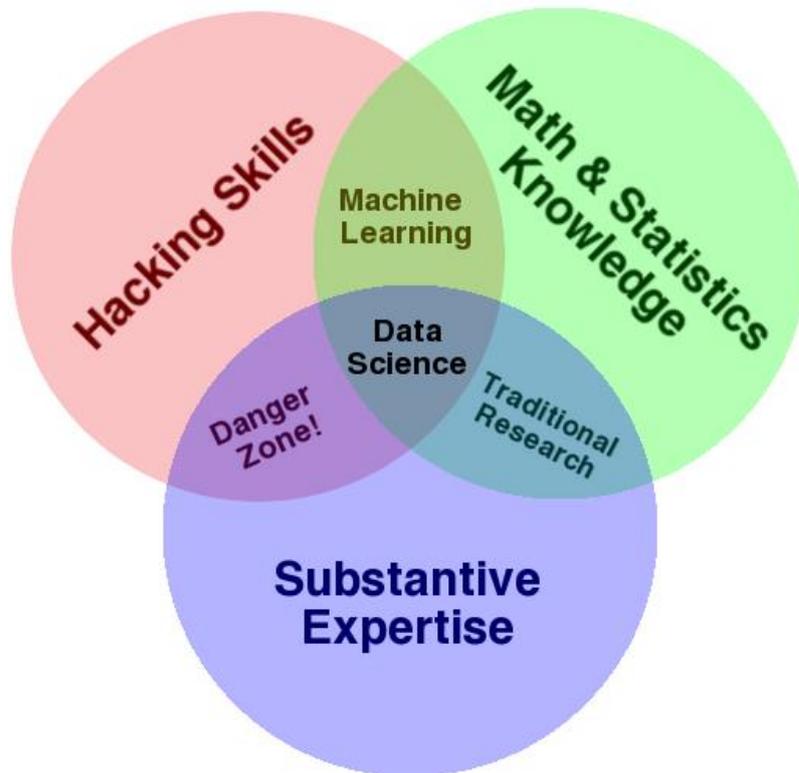
Fuente: (MINSa, 2020)

Anexo 2: Cantidad de Muertos en el Distrito de San Juan de Miraflores



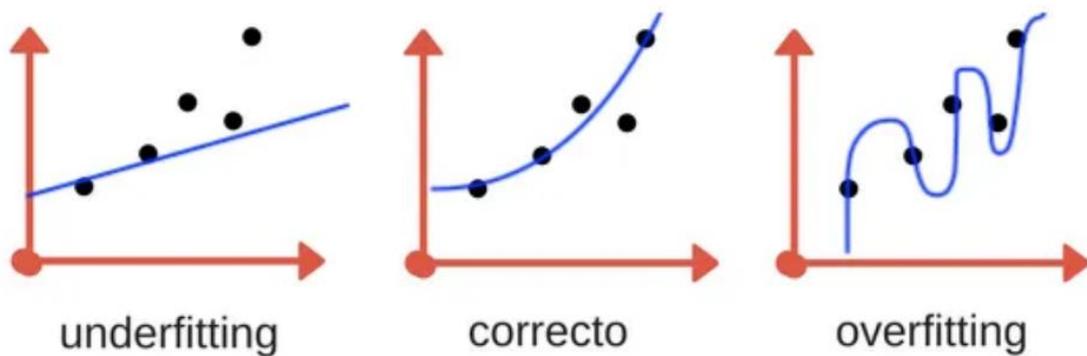
Fuente: (MINSa, 2020)

Anexo 3 Diagrama de Drew Conway



Fuente: (Conway, 2010)

Anexo 4: Overfitting y Underfitting



Fuente: (Bagnato, 2017)

Anexo 5: Procesamiento de Datos

Fecha	Infectados	Acumulado por Semana
20200329	5	61
20200330	2	
20200331	11	
20200401	2	
20200402	6	
20200403	12	
20200404	23	
20200405	9	204
20200406	29	
20200407	21	
20200408	37	
20200409	32	
20200410	36	
20200411	40	
20200412	27	229
20200413	45	
20200414	28	
20200415	41	
20200416	37	
20200417	19	
20200418	32	
20200419	36	335
20200420	34	
20200421	47	
20200422	52	
20200423	57	
20200424	48	
20200425	61	
20200426	33	419
20200427	53	
20200428	74	
20200429	73	
20200430	49	
20200501	36	
20200502	101	
20200503	57	512
20200504	106	
20200505	87	
20200506	97	
20200507	69	

20200508	57	533
20200509	39	
20200510	22	
20200511	122	
20200512	80	
20200513	90	
20200514	56	
20200515	81	
20200516	82	
20200517	52	735
20200518	80	
20200519	109	
20200520	124	
20200521	154	
20200522	136	
20200523	80	
20200524	27	650
20200525	100	
20200526	111	
20200527	129	
20200528	128	
20200529	91	
20200530	64	
20200531	43	423
20200601	80	
20200602	61	
20200603	21	
20200604	45	
20200605	89	
20200606	84	
20200607	41	516
20200608	64	
20200609	105	
20200610	141	
20200611	60	
20200612	56	
20200613	49	550
20200614	35	
20200615	157	
20200616	155	
20200617	70	
20200618	58	
20200619	34	
20200620	41	

20200621	37	336
20200622	43	
20200623	40	
20200624	53	
20200625	65	
20200626	60	
20200627	38	
20200628	21	279
20200629	32	
20200630	35	
20200701	39	
20200702	38	
20200703	54	
20200704	60	
20200705	20	293
20200706	45	
20200707	53	
20200708	42	
20200709	53	
20200710	45	
20200711	35	
20200712	32	456
20200713	55	
20200714	47	
20200715	70	
20200716	85	
20200717	100	
20200718	67	
20200719	42	592
20200720	81	
20200721	104	
20200722	106	
20200723	96	
20200724	91	
20200725	72	
20200726	44	607
20200727	96	
20200728	51	
20200729	93	
20200730	108	
20200731	97	
20200801	118	
20200802	53	681
20200803	112	

20200804	141	
20200805	88	
20200806	106	
20200807	75	
20200808	106	
20200809	61	752
20200810	99	
20200811	134	
20200812	105	
20200813	90	
20200814	130	
20200815	133	
20200816	130	916
20200817	128	
20200818	128	
20200819	132	
20200820	125	
20200821	177	
20200822	96	
20200823	44	641
20200824	138	
20200825	139	
20200826	89	
20200827	73	
20200828	94	
20200829	64	
20200830	39	596
20200831	83	
20200901	91	
20200902	61	
20200903	70	
20200904	156	
20200905	96	
20200906	61	574
20200907	86	
20200908	67	
20200909	99	
20200910	36	
20200911	88	
20200912	137	

Fuente: Propia

Anexo 6: Formula de Des estandarización.

$$Y = r(\text{Max} - \text{Min}) + \text{Min}$$

Fuente: Propia