



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL
POSGRADOS

PROGRAMA DE MAESTRÍA EN MATEMÁTICA APLICADA
MODALIDAD DE TITULACIÓN PROYECTO DE DESARROLLO
Trabajo de titulación previo a la obtención del grado académico de
Magíster en Matemática Aplicada

**“SIMULACIÓN DE PRONÓSTICOS DE VENTAS EN LA
EMPRESA IMPACTEX MEDIANTE REDES NEURONALES”**

Autor: Ing. Mayra Cristina Llunitasig Galarza

Director: Ing. Fabián Rodrigo Salazar Escobar, Ph.D

Ambato – Ecuador

2021

APROBACIÓN DEL TRABAJO DE TITULACIÓN

A la Unidad de Titulación/Unidad Académica de Titulación de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial. El Tribunal receptor de la Defensa del Trabajo de Titulación presidido por Ing. Elsa Pilar Urrutia Urrutia Mg., e integrado por los señores: Ing. Víctor Santiago Manzano Villafuerte Mg. y Dr. Freddy Geovanny Benalcázar Palacios Mg, designados por la Unidad de Titulación de la Universidad Técnica de Ambato, para receptor el Trabajo de Titulación con el tema: “Simulación de pronósticos de ventas en la empresa IMPACTEX mediante redes neuronales.”, elaborado y presentado por la Ing. Mayra Cristina Llumitasig Galarza, para optar por el Grado Académico de Magister en Matemática Aplicada; una vez escuchada la defensa oral del Trabajo de Titulación el Tribunal aprueba y remite el trabajo para uso y custodia en las bibliotecas de la Universidad Técnica de Ambato.



Firmado electrónicamente por:
**ELSA PILAR
URRUTIA**

Ing. Elsa Pilar Urrutia Urrutia Mg.

Presidente y Miembro del Tribunal de Defensa



Firmado electrónicamente por:
**VÍCTOR SANTIAGO
MANZANO
VILLAFUERTE**

Ing. Víctor Santiago Manzano Villafuerte Mg.

Miembro del Tribunal de Defensa



Firmado electrónicamente por:
**FREDDY GEOVANNY
BENALCAZAR
PALACIOS**

Dr. Freddy Geovanny Benalcázar Palacios Mg.

Miembro del Tribunal de Defensa

AUTORÍA DEL TRABAJO DE TITULACIÓN

La responsabilidad de las opiniones, comentarios y críticas emitidas en el trabajo de titulación presentado con el tema: “Simulación de pronóstico de ventas en la empresa IMPACTEX mediante redes neuronales”, le corresponde exclusivamente a la: Ing. Mayra Cristina Llunitasig Galarza, autor bajo la dirección del: Ing. Fabián Rodrigo Salazar Escobar, Ph.D, director del trabajo de investigación; y el patrimonio intelectual pertenece a la Universidad Técnica de Ambato.



Firmado electrónicamente por:
**MAYRA CRISTINA
LLUNITASIG
GALARZA**

Ing. Mayra Cristina Llunitasig Galarza.



Firmado electrónicamente por:
**FABIAN RODRIGO
SALAZAR ESCOBAR**

Ing. Fabián Rodrigo Salazar Escobar, Ph.D.

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que el trabajo de titulación, sirva como un documento disponible para su lectura, consulta y procesos de investigación, según las normas de la Institución.

Cedo los derechos de mi trabajo de titulación, con fines de difusión pública, además apruebo la reproducción de éste, dentro de las regulaciones de la Universidad Técnica de Ambato.



Firmado electrónicamente por:
**MAYRA CRISTINA
LLUMITASIG
GALARZA**

Ing. Mayra Cristina Llumitasig Galarza

DEDICATORIA

El presente proyecto de investigación está dedicado a mis padres Hugo y María, por su apoyo incondicional, amor y confianza brindados siempre.

Mayra Cristina LLumitasig Galarza

AGRADECIMIENTO

A mis padres por el apoyo, confianza y cariño que siempre me brindan en el transcurso de mi etapa estudiantil.

A la empresa IMPACTEX por la apertura brindada durante todo el desarrollo del presente trabajo.

A mi tutor por los conocimientos transmitidos durante el desarrollo del proyecto de investigación.

Mayra Cristina LLumitasig Galarza

Índice

APROBACIÓN DEL TRABAJO DE TITULACIÓN	I
AUTORÍA DEL TRABAJO DE TITULACIÓN	II
DERECHOS DE AUTOR	III
DEDICATORIA	IV
AGRADECIMIENTO	V
Índice de figuras	IX
Índice de Tablas	X
RESUMEN EJECUTIVO	XI
ABSTRACT	XII
1. CAPÍTULO I	
PROBLEMA DE INVESTIGACIÓN	1
1.1. Introducción	1
1.2. Justificación	2
1.3. Objetivos	3
1.3.1. Objetivo General	3
1.3.2. Objetivos Específicos	3
2. CAPÍTULO II	
ANTECEDENTES INVESTIGATIVOS	4
2.1. Estado del Arte	4
2.1.1. Sistema de Pronóstico de la Demanda de Productos Farmacéuticos Basado en Redes Neuronales	4
2.1.2. Pronóstico de demanda por medio de redes neuronales artificiales	5
2.1.3. Modelo de elaboración de pronóstico de ventas mediante el uso de redes neuronales artificiales y SVR	6
2.1.4. Modelo para el pronóstico de la demanda de agua potable de EMAPA-I aplicando Redes Neuronales Artificiales.	6

2.1.5. Pronóstico de ventas de las empresas del sector alimentos: una aplicación de redes neuronales	7
2.1.6. Predicción demanda eléctrica española. Implementación de redes neuronales recurrentes en Python	8
2.1.7. Modelo basado en Redes Neuronales Recurrentes LSTM para la predicción de la siguiente actividad en procesos de negocios	8
2.1.8. Obtención de un modelo neuronal para la estimación de la concentración de etanol en la destilería Héctor Molina	9
2.1.9. Modelado y Simulación de un Sistema de Detección de Intrusos Utilizando Redes Neuronales Recurrentes	9
2.1.10. Modelos de redes neuronales recurrentes en clasificación de patentes	10
2.1.11. Modelos predictivos basados en Redes Neuronales en tiempo discreto	10
2.1.12. Comparación de predicción basada en redes neuronales contra métodos estadísticos en el pronóstico de ventas	10
2.1.13. Predicción de proteínas con redes neuronales: Redes Feedforward vs Redes Recurrentes	11
2.2. Fundamentación teórica.	11
2.2.1. Redes Neuronales Artificiales	11
2.2.2. Modelo Neuronal	13
2.2.3. Arquitecturas Neuronales	15
2.2.4. Clasificación según el número de capas	15
2.2.5. Clasificación según el tipo de conexiones	16
2.2.6. Clasificación según el tipo de aprendizaje	18
2.2.7. Clasificación según su aplicación	20
2.2.8. Funciones de activación	20
2.3. Principales librerías de Python utilizadas en Machine Learning	22
2.4. Pronósticos	23
2.4.1. Pronósticos de ventas y su Clasificación.	23
2.5. Empresa Impactex y sus productos:	24
3. CAPÍTULO III	
MARCO METODOLÓGICO	30
3.1. Ubicación.	30
3.2. Equipos y materiales.	30

3.3. Tipo de investigación.	30
3.3.1. Investigación bibliográfica.	30
3.3.2. Investigación aplicada.	30
3.4. Idea a defender.	31
3.5. Población o muestra.	31
3.6. Recolección de información.	31
3.7. Procesamiento de la información y análisis estadístico.	31
3.8. Resultados alcanzados.	31
4. CAPÍTULO IV	
RESULTADOS Y DISCUSIÓN	33
4.1. Análisis de resultados.	33
4.1.1. Desarrollo de la propuesta.	33
4.1.2. Recolección y tratamiento de datos históricos.	34
4.1.3. Implementación de la red neuronal artificial.	42
4.1.4. Selección de parámetros de la red neuronal	44
4.2. Resultados	51
5. CAPÍTULO V	
CONCLUSIONES Y RECOMENDACIONES	55
5.1. CONCLUSIONES	55
5.2. RECOMENDACIONES	56
GLOSARIO	57
6. BIBLIOGRAFÍA	58
7. ANEXOS	63
7.1. Código de entrenamiento de la Red Neuronal Artificial	63
7.2. Código de implementación para el pronóstico	67
7.3. Interface creada para elegir el número de años del pronóstico	83
7.4. Tabla de pronóstico para 5 años.	84

Índice de figuras

1. Estructura de una Neurona Biológica	12
2. Estructura de una Neurona Artificial.	13
3. Estructura de una Neurona Artificial	14
4. Estructura de una Red Neuronal Monocapa	15
5. Estructura de una Red Neuronal Multicapa	16
6. Estructura de una Red Neuronal Recurrente	17
7. Algunas funciones de activación	20
8. Función de activación tangente hiperbólica y sigmoide	21
9. Línea de productos Mao Slim	25
10. Línea de productos Mao Jr	25
11. Línea de productos Mao Underaction	26
12. Línea de productos Mao Kids	26
13. Esquema de la metodología utilizada para el pronóstico de ventas	33
14. Datos filtrados de la empresa IMPACTEX	35
15. Ejemplo de la estructura de matrices de datos para entrenamiento y testeo.	40
16. Estructura de una Red Neuronal Recurrente.	43
17. Estructura interna de una Red Neuronal Recurrente.	44
18. Gráfica de MSE con diferentes parámetros	45
19. Resultados de los parámetros seleccionados	48
20. Estructura de una Red Neuronal Recurrente.	48
21. Respuesta del error para los productos BH7010, C585, 1112.6	49
22. Curva de pronóstico para tres años del producto BH1060	52
23. Interface para elegir el número de años para el pronóstico.	83
24. Pronóstico para 5 años del producto BH1060	86
25. Pronóstico para 5 años del producto BH7010	87
26. Pronóstico para 5 años del producto C585	87

Índice de tablas

1. Descripción de códigos de productos.	27
2. Ejemplo de descripción de códigos.	27
3. Tabla códigos de todos los productos ofertados.	28
4. Resumen de clasificación ABC	37
5. Histórico de ventas de los años 2007 y 2008 de la empresa IMPACTEX	38
7. Error de entrenamiento y de testeo para el producto BH1060	46
8. Error de entrenamiento y de testeo para el producto BH7010	50
9. Error de entrenamiento y de testeo para el producto C585	50
10. Error de entrenamiento y de testeo para el producto 1112.6	51
11. Error para el pronóstico por cada producto	52
12. Pronóstico para tres años de los productos BH1060,BH7010, C55, 1112.6	53
13. Pronóstico para cinco años de los productos BH1060,BH7010, C55, 1112.6	84

**UNIVERSIDAD TÉCNICA DE AMBATO FACULTAD DE
INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL
MAESTRÍA EN MATEMÁTICA APLICADA.**

TEMA:

"Simulación de pronósticos de ventas en la empresa IMPACTEX mediante redes neuronales."

AUTOR: Ing. Mayra Cristina Llumitasig Galarza.

DIRECTOR: Ing. Fabián Rodrigo Salazar Escobar, Ph.D.

LÍNEA DE INVESTIGACIÓN:

Tecnología de la Información y Sistemas de Control.

FECHA: 07 de julio de 2021.

RESUMEN EJECUTIVO

En el presente trabajo de investigación se realizó la simulación del pronóstico de venta en la empresa IMPACTEX mediante Redes Neuronales Artificiales. Para lo que se utilizó el software libre Python con la ayuda de las librerías Tensorflow y Keras, los datos utilizados fueron datos históricos obtenidos de la empresa IMPACTEX de los años 2008-2019, de la cual se obtuvo un registro de ventas por código de producto (127 productos en total), para realizar el pronóstico se realizó un análisis ABC que permitió determinar los productos con mayor demanda obteniendo como resultado el producto BH1060 seguido por el producto BH7010 ubicados en el grupo A, mientras que en el grupo B se ubicó el producto C585 y el grupo C lo conformó el producto 1112.6. La estructura para cada red neuronal se obtuvo luego de variar los parámetros de epoch, número de capas de la red neuronal y número de neuronas, coincidiendo en tres el número de capas ocultas para los cuatro productos utilizados, el número de neuronas por capas y epoch varía dependiendo la respuesta de error obtenida en la simulación. Los parámetros seleccionados generaron un error de 2,60 % para el producto tipo A (BH1060), para el producto tipo A (BH1070) 3,64 % , para el producto tipo B (C585) 3,02 % y un error del 3,27 % para el producto C (1112.6).

Palabras clave: Redes neuronales artificiales, pronóstico de ventas, históricos de ventas, redes LSTM, redes recurrentes, Función de activación, hiperparámetros, Redes neuronales Python, machine learning, inteligencia artificial.

**TECHNICAL UNIVERSITY OF AMBATO FACULTY OF
SYSTEMS, ELECTRONIC AND INDUSTRIAL ENGINEERING
MASTER OF APPLIED MATHEMATICS**

THEME:

“Sales forecast simulation in the IMPACTEX company using neural networks.”

AUTHOR: Ing. Mayra Cristina Llumitasig Galarza.

DIRECTED BY: Ing. Fabián Rodrigo Salazar Escobar, Ph.D.

LINE OF RESEARCH:

Information Technology and Control Systems.

DATE: July 07, 2021

EXECUTIVE SUMMARY

In the present research work, the simulation of the sales forecast was carried out in the IMPACTEX company using Artificial Neural Networks. For which the free Python software was used with the help of the Tensorflow and Keras libraries, the data used was historical data obtained from the IMPACTEX company for the years 2008-2019, from which a sales record was obtained by product code. (127 products in total), to carry out the forecast an ABC analysis was carried out that allowed determining the products with the highest demand, obtaining as a result the product BH1060 followed by the product BH7010 located in group A, while in group B the product C585 and group C was made up of product 1112.6. The structure for each neural network was obtained after varying the epoch parameters, the number of layers of the neural network and the number of neurons, the number of hidden layers coinciding in three for the four products used, the number of neurons per layer and epoch. varies depending on the error response obtained in the simulation. The selected parameters generated an error of 2.60 % for product type A (BH1060), for product type A (BH1070) 3.64 %, for product type B (C585) 3.02 % and a 3.27 % error for product C (1112.6).

Keywords: Artificial neural networks, sales forecast, sales history, LSTM networks, recurring networks, Trigger function, hyperparameters, Python neural networks, machine learning, artificial intellig.

1. CAPÍTULO I

PROBLEMA DE INVESTIGACIÓN

1.1. Introducción.

El este trabajo se presenta una metodología que sigue el comportamiento de ventas en la empresa IMPACTEX partiendo de datos históricos mediante el uso de Redes Neuronales Artificiales.

Debido a que el mundo en el que operan las organizaciones cambia constantemente, siempre ha sido necesario los pronósticos. Sin embargo, recientemente se ha incrementado la dependencia de los métodos que implica técnicas sofisticadas de manipulación de datos. De la noche a la mañana han surgido nuevas tecnologías y disciplinas, la competencia en muchas áreas se ha vuelto más reñida, el internet se ha vuelto una fuente importante de datos e información para la toma de decisiones. Las organizaciones que no reaccionen ante las condiciones cambiantes y no puedan proveer un futuro con cierto grado de precisión, están condenadas a extinguirse [Hanke and Wichern, 2006].

Los métodos estadísticos convencionales usan en general determinados parámetros o datos asociados a períodos de tiempo cuya relación con la previsión de demanda es complicada; la realidad es que muchos de los que toman decisiones en el campo de la previsión de demanda lo hacen basándose en su instinto y experiencia. La realidad también nos dice que la demanda se establece como un requisito primordial que debe soportar el plan financiero o plan de ventas con lo que, la mayor parte de las veces, el proceso de planificación de la demanda está fuertemente condicionado por elementos que, en un principio, tiene que ver con el mercado y los clientes [Serna, 2003].

Como lo menciona [Gupta and Kumar, 2013], los modelos de pronósticos de ventas presentan avances significativos desde los modelos estadísticos tradicionales como regresión lineal simple y el modelo ARIMA hasta los modelos avanzados Redes Neuronales Artificiales (RNA).

Los pronósticos de ventas en las empresas están a cargo de profesionales expertos en el tema, que realizan análisis de crecimiento en base a resultados anteriores reflejados en reportes históricos de ventas (instinto y experiencia), por otra parte, también se utilizan

métodos estadísticos convencionales como modelos básicos de regresión lineal, análisis de series de tiempo, etc. Para el uso de estas herramientas se necesita información cuantitativa del comportamiento de la demanda a través del tiempo, muchos de los que toman decisiones en el campo de la previsión de demanda lo hacen por medio de instintos y experiencias, las cuales son utilizadas para valorar el impacto de campañas promocionales, de introducción de nuevos productos u otros.

Actualmente, las redes neuronales artificiales son una de las técnicas más usadas para la resolución de pronósticos de este tipo, debido a sus óptimos resultados y su fácil metodología de resolución, es así que se se han realizado varios estudios comparativos de métodos tradicionales y RNA, dando como resultados mayormente satisfactorios los modelos que utilizan RNA cuyos resultados se los puede observar en [Santoyo and González, 2014].

1.2. Justificación.

En la actualidad uno de los principales problemas con el que se enfrentan las empresas es el manejo eficiente del stock de manera que se desea evitar dos casos: tener los productos en exceso que pueda provocar gastos innecesarios de materia prima y mano de obra, y por otra parte la falta de stock que implicaría pérdida de dinero con respecto a clientes y oportunidades de ventas provocando insatisfacción. Hoy por hoy en el mejor de los casos las empresas tienden a utilizar ciertas técnicas de pronósticos para determinar la cantidad que se producirá de determinado producto, dichas técnicas pueden ser cuantitativas o cualitativas. Independientemente de la técnica elegida, el problema real de los pronósticos es su falta de confiabilidad, ya que para la necesidad que surge en la empresa este dato debe ser preciso, el rango de error deberá tender a cero para garantizar una producción exitosa que genere ganancias significativas. En el presente trabajo se plantea una técnica para realizar el pronóstico de ventas para un período de tiempo deseado, esta técnica servirá para tener una estimación lo más acertada posible sobre la producción que permita realizar la toma de decisiones con respecto a la producción de determinado producto, con el fin de incrementar ingresos y disminuir costos, así como también la sobreproducción. Hoy en día los conceptos de Business Intelligence y Redes Neuronales no son desconocidos ya que son herramientas que ayudan a optimizar el proceso de toma de decisiones. Todas las empresas poseen un sistema que permite pronosticar las ventas en determinado período, sin embargo, un algoritmo basado en redes neuronales promete

generar una respuesta eficiente con errores mínimos, esto debido a que una Red Neuronal Artificial (RNA) no ejecuta instrucciones ni programas, por otra parte reaccionan, aprenden y se auto-organizan siendo una ventaja ante los algoritmos tradicionales, por lo antes expuesto se propone una, simulación de pronósticos de ventas en la empresa IMPACTEX mediante Redes Neuronales, el mismo que generará un pronóstico de ventas más exacto y eficaz que algoritmos tradicionales incrementando los ingresos de la empresa y disminuyendo costos con aplicación de técnicas innovadoras como Redes Neuronales y Business Intelligence.

1.3. Objetivos

1.3.1. Objetivo General

- Simular pronósticos de ventas en la empresa IMPACTEX mediante redes neuronales.

1.3.2. Objetivos Específicos

- Indagar conceptos básicos que ayuden al desarrollo del proyecto .
- Gestionar la base de datos obtenida del histórico de ventas para identificar los indicadores claves de rendimiento y definir el grupo de datos que servirán para entrenamiento y testeo de la Red Neuronal.
- Simular el pronóstico de ventas mediante Redes Neuronales y determinar el software para el desarrollo y entrenamiento de la Red Neuronal.
- Evaluar el desempeño de la Red Neuronal comparando los datos de producción generados por la empresa con los obtenidos mediante el modelo propuesto.

2. CAPÍTULO II

ANTECEDENTES INVESTIGATIVOS

2.1. Estado del Arte

En esta sección se muestra trabajos relacionados al problema que se resolverá en el presente trabajo, primero se detalla algunas aplicaciones y áreas que aplican Redes Neuronales Artificiales, seguido de algunos trabajos relacionados con el tema de investigación y finalmente para obtener una visión clara del problema, se tiene la fundamentación teórica.

Aplicaciones

Los pronósticos en la actualidad, son utilizados con mayor frecuencia en empresas que tienen la necesidad de cumplir con la demanda en el mercado de algún producto específico. Teniendo en cuenta un histórico de ventas se puede realizar un pronóstico adecuado que ayude a cumplir con dichas necesidades. Los pronósticos pueden tener aplicaciones en algunas áreas como:

- Alimenticia
- Finanzas
- Industria, Comercio y Transporte
- Medicina
- Economía
- Industria Automotriz
- Recursos Renovables y no Renovables

A continuación se citará algunos trabajos realizados a lo largo del tiempo, cuyo desarrollo se basa en el uso de las Redes Neuronales Artificiales.

2.1.1. Sistema de Pronóstico de la Demanda de Productos Farmacéuticos Basado en Redes Neuronales

De acuerdo con [\[Gil and Rodríguez, 2010\]](#), se realizó una red neuronal (perceptrón multicapa) para el pronóstico de la demanda de productos farmacéuticos, se partió de una fase de aprendizaje con el algoritmo backpropagation que brinda una tasa de error de

3.57 % en el mejor caso encontrado; su implementación se desarrolló bajo la tecnología de MATLAB para la construcción de la red neuronal y del lenguaje JAVA para el diseño de la interfaz gráfica de usuario. La supervivencia en los negocios altamente competitivos de hoy requiere una visión precisa de la demanda para poner en marcha los planes de producción, inventario, distribución y compra dentro de las empresas; el sector farmacéutico no es la excepción, pues los efectos de las temporadas, promociones, cambios de precios, publicidad, productos con bajo o alto nivel de movimiento y datos atípicos en general afectan en la determinación de la misma.

- Los resultados que se obtuvieron en la investigación muestra que, con la poca cantidad de datos históricos se llegó a un margen de error del 3,57 % que comparado con el error del 31,23 % de la persona encargada de las compras (error experto – mediante un sistema) es menor.
- Los márgenes de error del 4,93 % y 6,38 % correspondientes a los productos RG y CH respectivamente son menores a los errores del experto.
- El pronóstico realizado con la red neuronal se acerca a la realidad, por lo que permite tener un stock más acertado de los productos con más demanda.

2.1.2. Pronóstico de demanda por medio de redes neuronales artificiales

Desde el punto de vista de [Salazar Aguilar and Cabrera Ríos, 2007], en toda industria, la planeación es una necesidad, y un objetivo importante de la planeación es tratar de prever lo que puede suceder en el futuro. En este trabajo, se colaboró con una empresa de telecomunicaciones con necesidad de planear a nivel operacional, estratégico y táctico para mantenerse competitiva ante las fluctuaciones de mercado y cursos de acción de sus competidores. El autor plantea una metodología de selección de parámetros de un modelo de RNAs que utiliza técnicas establecidas y confiables y hace entendible la interrelación entre los varios parámetros de la RNA. Se demostró el funcionamiento de la metodología a través de un caso práctico, en el que se utilizaron modelos de RNA con múltiples salidas. Los resultados de este trabajo apoyan la utilización de las RNAs como técnicas confiables de pronóstico y apuntan a la factibilidad de su instauración en la industria.

2.1.3. Modelo de elaboración de pronóstico de ventas mediante el uso de redes neuronales artificiales y SVR

Teniendo en cuenta a [Peralta, 2018], que habla sobre la importancia de pronosticar o prever el nivel de ventas siendo vital para la supervivencia de una empresa. Al no tener un patrón claro, la investigación sobre mejores modelos continúa en desarrollo. A través del avance tecnológico ha sido posible desarrollar aplicaciones basadas en redes neuronales artificiales (RNA), enfocadas en el desarrollo de pronóstico de ventas de productos de consumo frecuente, mejorando la exactitud de los sistemas de pronóstico tradicionales. En éste estudio se compara el desempeño de los modelos tradicionales frente a sistemas más desarrollados de redes neuronales artificiales y máquinas de soporte vectorial o regresión de soporte vectorial (SVM-SVR). Se demostró la importancia de considerar factores externos como indicadores macroeconómicos e internos como los microeconómicos, (por ejemplo, precios de productos relacionados), los cuales afectan el nivel de ventas en una organización.

Algunas conclusiones de este desarrollo fueron las siguientes:

- Una RNA artificial es un avance radical en la forma de anticipar comportamientos y valores, es decir, un entorno fácilmente cuantificable está a la disposición para identificar una tendencia o comportamiento futuro.
- El estudio resalta la eficacia de utilizar RNA para el pronóstico de series temporales considerando tanto factores externos como internos a la organización.
- Los indicadores considerados como entradas del modelo no presentan el mismo nivel de importancia por lo que, es posible determinar según el algoritmo de Garson implementado en el algoritmo de la RNA.

2.1.4. Modelo para el pronóstico de la demanda de agua potable de EMAPA-I aplicando Redes Neuronales Artificiales.

En el Ecuador existen varios trabajos de titulación enfocados en el uso de Redes Neuronales Artificiales y el Pronóstico, uno de ellos es el realizado por [Pavón Valencia, 2019], en el cual realiza un modelo para el pronóstico de demanda de agua potable usando RNAs haciendo uso de la herramienta computacional Neural Network de Matlab. Los pronósticos tienen un papel fundamental, siendo una de las premisas para planificar, organizar y controlar un conjunto de actividades o procesos, es importante señalar que se

debe tener en cuenta en el desarrollo del modelo como por ejemplo: el horizonte temporal del pronóstico para evaluar las variables que serán las entradas en el modelo, ya sean dependientes como el volumen de consumo de agua de los años posteriores al pronóstico y, los meses y años que son las variables independientes.

El desarrollo del modelo llevó a obtener resultados como :

- Error Cuadrático Medio de 4,71 y un coeficiente de correlación del 0,98 que representa un alto nivel de confiabilidad del pronóstico en relación con el comportamiento de los valores reales que se ha contrastado con modelos estadísticos tradicionales.
- El pronóstico obtenido es importante en el estudio técnico y desarrollo de proyectos de la empresa, dicha información es de vital importancia para la empresa por la relación con los departamentos y la influencia del pronóstico en sus actividades.
- El análisis y comparación de los resultados obtenidos con el modelo de Redes Neuronales Artificiales y el modelo estadístico tradicional de Box-Jenkins SRT, con la finalidad de reconocer las diferencias y similitudes en los modelos y facilitar la interpretación de los resultados donde los resultados arrojados por el modelo de RNA presenta mayor similitud en el comportamiento del consumo real.

2.1.5. Pronóstico de ventas de las empresas del sector alimentos: una aplicación de redes neuronales

Según [Morales Castro et al., 2019], se presenta una propuesta metodológica para determinar el comportamiento de las ventas de las siguientes empresas del sector alimentos: Industrias Bachoco, Grupo Bafar, Grupo Bimbo, Gruma, Grupo Herdez, Grupo Lala y Grupo Industrial Maseca. Se comparan modelos lineales y no lineales para determinar aquellos que presenten un mayor ajuste a los datos históricos. Esta investigación se enfocó en un grupo de empresas que cotizan en la Bolsa Mexicana de Valores (BMV) que pertenecen al sector alimentos para realizar el pronóstico de ventas. En esta investigación se decidió estudiar a las empresas que pertenecen al sector de alimentos. Los resultados obtenidos fueron los siguientes:

- Los métodos de redes neuronales artificiales y minería de datos consideran las variables de entrada y ponderan el peso que cada una de ellas tiene sobre las ventas. Mientras que los modelos lineales eliminan variables o periodos de tiempo para ajustar los datos a los criterios de normalidad. En este proceso se puede eliminar información valiosa para determinar el comportamiento de las ventas.

- En este estudio se utilizaron las RNA como una herramienta para cuantificar el impacto de las variables propuestas en las ventas de siete empresas del sector alimentos (Bachoco, Bafar, Bimbo, Gruma, Herdez, Lala y Maseca) y destacar aquellas que presentaron un mayor impacto.
- En esta investigación se tiene como hallazgos que los modelos de regresión lineal presentan mayor precisión en el pronóstico de ventas de seis de siete empresas del sector alimentos analizadas.

2.1.6. Predicción demanda eléctrica española. Implementación de redes neuronales recurrentes en Python

Este proyecto realizado por [Cabezón, 2018], tiene como objetivo principal la predicción de la demanda eléctrica nacional mediante el desarrollo de modelos de redes neuronales recurrentes. Se hace un análisis descriptivo de la demanda eléctrica nacional previo a la implementación en Python de los modelos de redes elegidos. También se estudia el efecto de algunas variables explicativas en la predicción realizada la que se concluyó algunos resultados como:

- Para la predicción el sistema necesita de una gran cantidad de datos para poder obtener resultados confiables.
- Realizó un análisis de la serie de demanda eléctrica nacional de manera analítica, viendo su estacionalidad semanal y su comportamiento frente a festivos, periodos vacacionales y temperatura.
- Comprobaron que con modelos sencillos de redes neuronales recurrentes se mejoran los modelos actuales que disponían.

2.1.7. Modelo basado en Redes Neuronales Recurrentes LSTM para la predicción de la siguiente actividad en procesos de negocios

Para este trabajo [Alcocer et al., 2018] presenta una metodología para la implementación de una red neuronal recurrente para el descubrimiento de modelos de procesos de negocio, en específico predecir la siguiente actividad, como técnica de la minería de procesos. La metodología permite identificar las fases requeridas para predecir la actividad o evento siguiente, mediante la implementación de la red neuronal LSTM. El diseño de un enfoque de método predictivo para datos contenidos en los registros de eventos de procesos de negocio utilizando un enfoque de red neuronal recurrente LSTM. La aplicación

de esta técnica es novedosa en el campo de la minería de procesos de negocio. Al usar registros de eventos para el entrenamiento de la red neuronal LSTM con datos sintéticos y del mundo real, se demuestra que el método predictivo puede ser efectivo, presentando una tasa de exactitud aceptable, en todos los casos de estudio.

2.1.8. Obtención de un modelo neuronal para la estimación de la concentración de etanol en la destilería Héctor Molina

Actualmente la modelación de procesos biológicos utilizando técnicas de inteligencia artificial se ha convertido en tendencia. En este trabajo desarrollado por [de la Maza et al., 2018], se crearon modelos con las redes neuronales artificiales (RNA), perceptrón multicapa específicamente, que lograron estimar la concentración de etanol en los fermentadores de la destilería Héctor Molina, utilizando el software MATLAB 2013. Para este trabajo se utilizó una red neuronal del tipo perceptrón multicapa para la modelación de fermentadores en la destilería Héctor Molina y demostró su capacidad de estimar satisfactoriamente la concentración de etanol al final de la etapa de fermentación. El alto valor del coeficiente de determinación (0,9377) entre los valores simulados y los valores reales, y los bajos valores de error cuadrático medio en la validación del modelo indican la conveniencia de la utilización de redes neuronales en la modelación de la fermentación alcohólica.

2.1.9. Modelado y Simulación de un Sistema de Detección de Intrusos Utilizando Redes Neuronales Recurrentes

Citando a [Urbina et al., 2016], habla sobre el sistema de detección de intrusos basado en la identificación y clasificación de anomalías en el flujo de datos en una red, para esto se utilizó redes neuronales recurrentes basado en el protocolo HTTP (Hypertext Transfer Protocol). El autor desea demostrar que las redes neuronales recurrentes son más rápidas, exactas y adaptables para su aplicación en un sistema de detección de intrusos que los tipos de redes utilizados en una tesis anterior la cual se basó únicamente en redes tipo feedforward y redes Elman (redes parcialmente conectadas). Finalmente se llegó a la conclusión de que las redes neuronales recurrentes son superiores a las demás en velocidad de convergencia y efectividad, alcanzando 94 % en porcentaje de clasificación correcta de ataques a la red perpetrados por intrusos en un tiempo de 60 epochs, a comparación de 106 epochs y menos del 90 % en clasificación para una red Elman. Sin embargo, las redes neuronales recurrentes poseen una complejidad importante como desventaja, la cual

aumenta exponencialmente con el número de neuronas en la red.

2.1.10. Modelos de redes neuronales recurrentes en clasificación de patentes

Este trabajo desarrollado por [Guridi Mateos et al., 2017], se centra en la clasificación automática de patentes usando redes LSTM (Long-Short Term Memory). Las patentes son documentos que recogen y acotan la propiedad intelectual, compuestas principalmente por texto y múltiples etiquetas correspondientes a una clasificación jerárquica de varios niveles (IPC) . Luego de realizar una extensa investigación no se llegó a ningún modelo que esté a la altura del estado del arte de la clasificación de patentes, pero con las pruebas realizadas se ha obtenido un camino a seguir e ideas para llegar a conseguir entrenar un modelo con neuronas lstm para esta tarea de clasificación.

2.1.11. Modelos predictivos basados en Redes Neuronales en tiempo discreto

En la presente investigación de [Pérez-Ortiz, 2002], se estudió la aplicación de distintos modelos de redes neuronales recurrentes de tiempo discreto a diversas tareas de carácter predictivo. Se pudo comprobar que RNR como la red recurrente simple (RRS) o la red parcialmente recurrente (RPR) no tienen tanta memoria como podría presumirse en primera instancia. Aunque trabajan adecuadamente con lenguajes sencillos y altamente estructurados (especialmente lenguajes regulares), surgen diversos problemas a la hora de considerar fuentes secuenciales más elaboradas. Así, las RNR estudiadas no logran capturar en forma acertada la dinámica de los textos en lenguaje natural y son claramente superadas por modelos más sencillos. Por otro lado, los resultados obtenidos con secuencias numéricas correspondientes a señales de voz digitalizada son poco alentadores. Estos experimentos se llevaron a cabo considerando dos algoritmos de entrenamiento: el sencillo descenso por el gradiente y el más complejo filtro de Kalman extendido desacoplado (FKED). Aunque los problemas constatados se siguen produciendo al utilizar el FKED, este saca mayor provecho de la capacidad de memoria teórica de las RNR y supera ampliamente en todos los casos estudiados los resultados del descenso por el gradiente .

2.1.12. Comparación de predicción basada en redes neuronales contra métodos estadísticos en el pronóstico de ventas

El artículo desarrollado por [Santoyo and González, 2014], habla sobre la comparación de pronósticos con métodos estadísticos y Redes Neuronales, y, posteriormente seleccionar el método adecuado y con menor error cuadrático para de esta forma beneficiar a la

empresa con un pronóstico más acertado. Ya que los pronósticos de ventas son datos de entrada a diferentes áreas de la empresa y de ser imprecisos pueden generar gastos para la empresa. Para la comparación entre los métodos estadísticos y las Redes Neuronales, se trabajó con los siguientes métodos estadísticos.

- Método de Holt
- Método de Winters
- Método ARIMA

Luego de evaluar y comparar el error cuadrático medio obtenido por cada método se determino que, la red neuronal artificial fue la que generó menor error cuadrático medio en comparación con los métodos estadísticos de Holt, Winter y ARIMA. Por lo cual se concluye que la red neuronal es una opción potente para el pronóstico de la producción y una correcta planeación de stock de productos.

2.1.13. Predicción de proteínas con redes neuronales: Redes Feedforward vs Redes Recurrentes

De acuerdo con [Quintero, 2007] en su trabajo, se evaluó y se comparó el desempeño de las redes neuronales Feedforward vs las recurrentes ante la predicción de la estructura secundaria de una proteína. Los resultados obtenidos en este trabajo reflejan que las redes neuronales de arquitectura recurrente demostraron tener un mejor desempeño tanto en el entrenamiento como en su funcionamiento en la predicción de la estructura secundaria de proteínas a partir de la secuencia de aminoácidos. La red de arquitectura recurrente tiene un procedimiento de aprendizaje más eficiente y alcanza un nivel de estabilidad más rápido que la red feedforward. Las redes recurrentes alcanzan una efectividad del 73 %, superando a la red feedforward, que presentó una efectividad del 69 %.

2.2. Fundamentación teórica.

2.2.1. Redes Neuronales Artificiales

Según [Salas, 2004] dice que una Red Neuronal Artificial (*RNA*) es un esquema de computación distribuida inspirada en la estructura del sistema nervioso de los seres humanos. La arquitectura de una red neuronal es formada conectando múltiples procesadores elementales, siendo éste un sistema adaptivo que posee un algoritmo para ajustar sus pesos (parámetros libres) para alcanzar los requerimientos de desempeño del problema basado

en muestras representativas. También señala que la propiedad más importantes de las redes neuronales artificiales es su capacidad de aprender a partir de un conjunto de patrones de entrenamientos, es decir, es capaz de encontrar un modelo que ajuste los datos.

Por otra parte, [Caparrini, 2015] define una Red Neuronal Artificial (RNA) como un modelo matemático inspirado en el comportamiento biológico de las neuronas y en cómo se organizan formando la estructura del cerebro. El cerebro puede considerarse un sistema altamente complejo, donde se calcula que hay aproximadamente 100 mil millones neuronas en la corteza cerebral (humana) y que forman un entramado de más de 500 billones de conexiones neuronales (una neurona puede llegar a tener 100 mil conexiones, aunque la media se sitúa entre 5000 y 10000 conexiones).

Respecto a su funcionamiento, el cerebro puede ser visto como un sistema inteligente que lleva a cabo tareas de manera distinta a como lo hacen las computadoras actuales. Si bien estas últimas son muy rápidas en el procesamiento de la información, existen tareas muy complejas, como el reconocimiento y clasificación de patrones, que demandan demasiado tiempo y esfuerzo aún en las computadoras más potentes de la actualidad, pero que el cerebro humano es más apto para resolverlas, muchas veces sin aparente esfuerzo (por ejemplo, el reconocimiento de un rostro familiar entre una multitud de rostros). En la Figura 1 se puede observar las partes de una Neurona Humana.

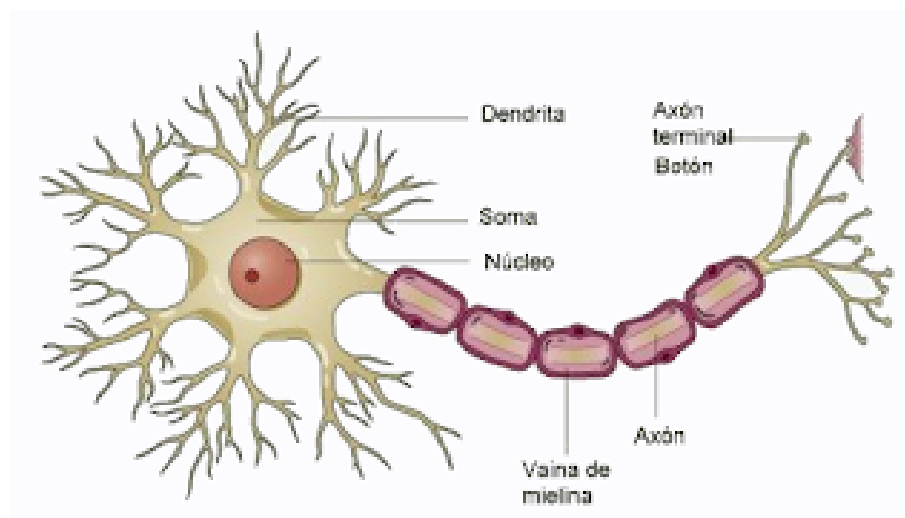


Figura 1: Estructura de una Neurona Biológica

Fuente: [Caparrini, 2015]

El primer modelo matemático de una neurona artificial, creado con el fin de llevar a cabo tareas simples, fue presentado en el año 1943 en un trabajo conjunto entre el psiquiatra y neuroanatomista Warren McCulloch y el matemático Walter Pitts [Caparrini, 2015],

como se muestra en la figura 2. Las redes neuronales artificiales se inspiran en el comportamiento conocido del cerebro humano (principalmente el referido a las neuronas y sus conexiones), trata de crear modelos artificiales que solucionen problemas difíciles de resolver mediante técnicas algorítmicas convencionales.

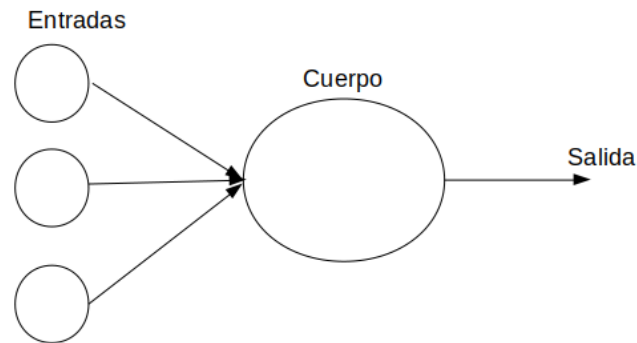


Figura 2: Estructura de una Neurona Artificial.

Fuente: Elaborado por la autora

2.2.2. Modelo Neuronal

Las redes neuronales son sistemas dinámicos autoadaptativos. Son adaptables debido a la capacidad de autoajustarse de los elementos procesales (neuronal) que componen el sistema. Son dinámicos, pues son capaces de estar constantemente cambiando para adaptarse a las nuevas condiciones [Hilera González et al., 2000]. Una Red Neuronal Artificial tiene cuatro elementos básicos que se detallan a continuación:

1. Conjunto de conexiones, pesos o sinapsis que determinan el comportamiento de la neurona.
2. Sumador, se encarga de sumar todas las entradas multiplicadas por su respectiva sinapsis.
3. Función de activación no lineal para limitar la amplitud de la salida de la neurona.
4. Umbral exterior que determina el umbral por encima del cual la neurona se activa.

Esta estructura se encuentra detallada en la figura 3.

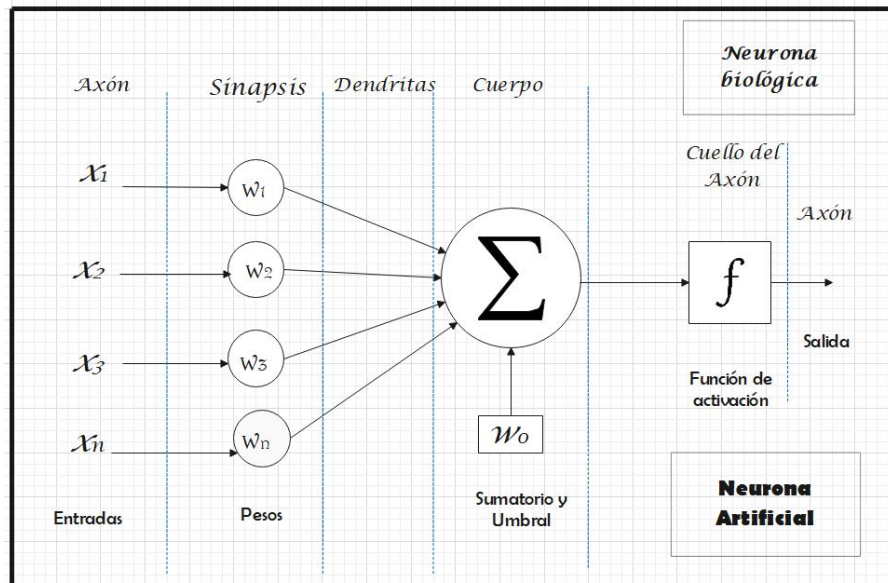


Figura 3: Estructura de una Neurona Artificial

Fuente: Elaborado por la autora

En donde:

- Conjunto de entradas $x_1, x_2, x_3, \dots, x_n$.
- Pesos sinápticos correspondientes a cada entrada $w_1, w_2, w_3, \dots, w_n$
- Función de agregación Σ .
- Función de transferencia f .

Según [Torres, 1994], en toda red neuronal se consideran básicamente tres tipos de unidades:

1. Unidades de Entrada que, como su nombre lo indica, reciben los valores de entrada del sistema y los pasan a otras unidades.
2. Unidades de salida que contienen los valores de respuesta"de la red neuronal después de cada proceso computacional.
3. Unidades ocultas que no tienen comunicación con el “ambiente externo” y que el sistema utiliza para representaciones internas.

Las entradas son el estímulo que la neurona artificial recibe del entorno que la rodea, y la salida es la respuesta a tal estímulo [Parisi F et al., 2003]. La neurona puede adaptarse al medio circundante y aprender de él modificando el valor de sus pesos sinápticos, y por ello son conocidos como los parámetros libres del modelo, ya que pueden ser modificados y adaptados para realizar una tarea determinada. La salida Y está definida por la siguiente expresión:

$$Salida = Y = f\left(\sum_{i=1}^n w_i x_i\right)$$

2.2.3. Arquitecturas Neuronales

Una neurona simple puede conectarse entre sí con más neuronas de similares características dando lugar a estructuras neuronales que se puede clasificar usando ciertos criterios como [Serrano et al., 2009] indica a continuación:

2.2.4. Clasificación según el número de capas

Red Neuronal Monocapa-Perceptron Simple: Corresponde a una red neuronal simple, se compone de una capa de neuronas que proyectan las entradas a una capa de neuronas de salida. La aplicación más común para este tipo de redes es la de memorias asociativas como se observa en la figura 4.

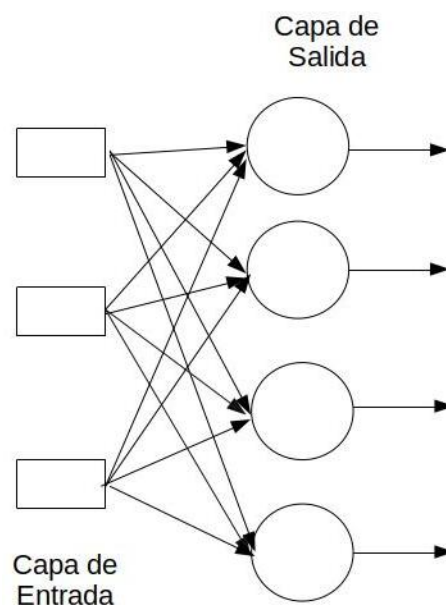


Figura 4: Estructura de una Red Neuronal Monocapa

Fuente: Elaborado por la autora

Red Neuronal Multicapa: En la figura 5 se observa una generalización de la red monocapa, en donde se incrementa el número de capas intermedias entre la capa de entradas y salidas, denominándolas capas ocultas.

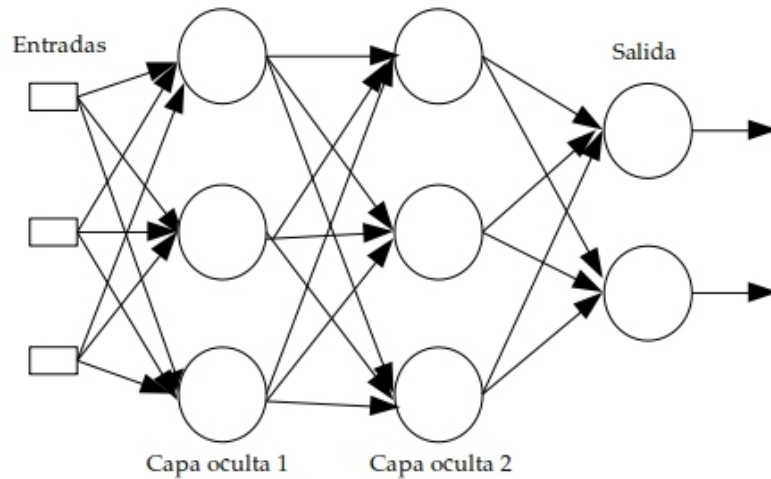


Figura 5: Estructura de una Red Neuronal Multicapa

Fuente: Elaborado por la autora.

2.2.5. Clasificación según el tipo de conexiones

Red Neuronal no Recurrente: Como lo indican [Serrano et al., 2009] en su trabajo en las redes neuronales no recurrentes la propagación de las señales se produce en un solo sentido, sin la posibilidad de la existencia de una retroalimentación, por lo cual no poseen memoria. Esta formado por un conjunto de nodos conocidos como neuronas artificiales que están conectadas y transmiten señales entre sí. Estas señales se transmiten desde la entrada hasta generar una salida. El objetivo principal de este modelo es aprender modificándose automáticamente a si mismo de forma que puede llegar a realizar tareas complejas que no podrían ser realizadas mediante la clásica programación basada en reglas. De esta forma se pueden automatizar funciones que en un principio solo podrían ser realizadas por personas

Red Neuronal Recurrente: No tienen una estructura de capas, sino que permiten conexiones arbitrarias entre las neuronas, incluso pudiendo crear ciclos, con esto se consigue crear la temporalidad, permitiendo que la red tenga memoria. Los datos introducidos en el momento t en la entrada, son transformados y van circulando por la red incluso en los instantes de tiempo siguientes. La figura 6 muestra el diseño de una red neuronal recurrente.

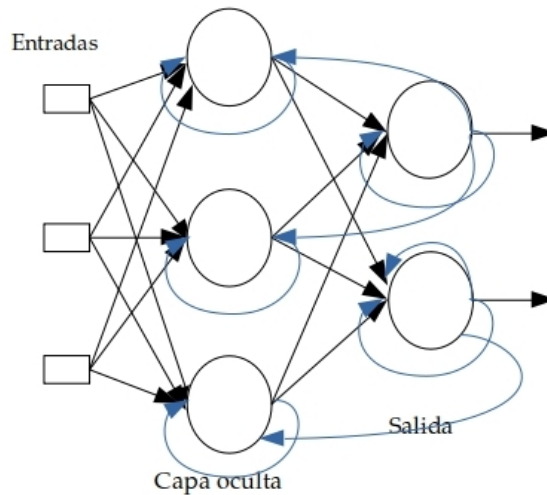


Figura 6: Estructura de una Red Neuronal Recurrente

Fuente: Elaborado por la autora.

Las LSTM (Long Short Term Memory), son un tanto generales como efectivos para capturar dependencias temporales a largo plazo. No sufren los obstáculos de optimización que afectan a las redes recurrentes simples [Greff et al., 2016].

Las RNR (Redes Neuronales Recurrentes), son capaces de realizar una amplia variedad de tareas computacionales incluyendo el tratamiento de secuencias, la continuación de Una trayectoria la predicción no lineal y la modelación de sistemas dinámicos [Cruz et al., 2007].

En las redes neuronales recurrentes tradicionales, durante la fase de retropropagación, la señal de gradiente puede llegar a ser multiplicada un gran número de veces (tantos como cada instante de tiempo) por la matriz de peso asociada a las neuronas de la capa oculta. Esto puede tener un gran efecto en el proceso de aprendizaje de la red puesto que la magnitud de los pesos de la matriz se ven afectados [Montesdeoca Santana, 2016].

Una Red LSTM es capaz de “recordar” un dato relevante en la secuencia y de preservarlo por varios instantes de tiempo. Por tanto, puede tener una memoria tanto de corto plazo (como las Redes Recurrentes básicas) como también de largo plazo. Comparado con una celda de red recurrente básica, la celda LSTM tiene una entrada y una salida adicional. Este elemento adicional se conoce como celda de estado. Esta celda de estado es la clave del funcionamiento de las Redes LSTM. La celda de estado es como una banda transportadora a la que se pueden añadir o de donde se pueden remover datos que no queremos que queden en la memoria de la red [Moreno, 2019].

2.2.6. Clasificación según el tipo de aprendizaje

El proceso de aprendizaje también conocido como entrenamiento de la red puede ser supervisado o no supervisado como lo detalla [Simeone, 2018] a continuación.

Aprendizaje Supervisado: Consiste en entrenar la red a partir de un conjunto de datos o patrones de entrenamiento compuesto por patrones de entrada y salida. El objetivo del algoritmo de aprendizaje es ajustar los pesos de la red de manera tal que la salida generada por la RNA sea lo más cercanamente posible a la verdadera salida dada una cierta entrada. Es decir, la red neuronal trata de encontrar un modelo al proceso desconocido que generó la salida y este aprendizaje se llama supervisado, pues se conoce el patrón de salida el cual hace el papel de supervisor de la red.

El aprendizaje supervisado se suele usar en:

- Problemas de clasificación (identificación de dígitos, diagnósticos, o detección de fraude de identidad).
- Problemas de regresión (predicciones meteorológicas, de expectativa de vida, de crecimiento etc).

Estos dos tipos principales de aprendizaje supervisado, clasificación y regresión, se distinguen por el tipo de variable objetivo. En los casos de clasificación, es de tipo categórico, mientras que, en los casos de regresión, la variable objetivo es de tipo numérico .

Los algoritmos más habituales que aplican para el aprendizaje supervisado para clasificación son:

- Árboles de decisión.
- Clasificación de Naïve Bayes.
- Regresión por mínimos cuadrados.
- Regresión Logística.
- Support Vector Machines (SVM).
- Métodos “Ensemble” (Conjuntos de clasificadores).

Aprendizaje no Supervisado: Tiene lugar cuando no se dispone de datos “etiquetados” para el entrenamiento. Sólo conocemos los datos de entrada, pero no existen datos de salida que correspondan a un determinado input. Por tanto, sólo podemos describir la estructura de los datos, para intentar encontrar algún tipo de organización que simplifique el análisis. Por ello, tienen un carácter exploratorio.

El aprendizaje no supervisado se suele usar en:

- Problemas de clustering
- Agrupamientos de co-ocurrencias
- Perfilado o profiling.

Aprendizaje por corrección de error: Consiste en presentar al sistema un conjunto de datos, representado la entrada y la salida deseada para dicha entrada, luego ajusta los pesos de las conexiones de la red en función del error cometido, es decir la diferencia entre los valores esperados y los obtenidos. Ejemplos de algoritmos:

- Perceptrón
- Delta o Mínimo error cuadrado (LMS Error: Least Mean Squared)
- Backpropagation o Programación hacia atrás (LMS multicapa)

Aprendizaje estocástico Realiza cambios aleatorios sobre los pesos (la matriz de pesos contiene los pesos de los enlaces que llegan a la capa de la red neuronal) va calculando la predicción y mejorando o empeorando con cada uno de los cambios, quedándose con los cambios que mejoren los resultados.

Aprendizaje competitivo y comparativo Permite realizar clasificaciones de los datos de entrada. La forma de actuación consiste en ir añadiendo elementos a una clase, si este nuevo elemento se determina que es de esta clase matiza los pesos, en caso contrario se puede crear una nueva clase con el elemento asociando a una serie de pesos propios.

Aprendizaje por refuerzo Se considera un aprendizaje más lento que el aprendizaje por corrección de errores, en este caso no se dispone de un conjunto completo de los datos exactos de salida (no tenemos una “etiqueta de salida”) sino que se le indica solamente si el dato es aceptable o no, con esto el algoritmo ajusta los pesos basándose en un mecanismo de probabilidades.

2.2.7. Clasificación según su aplicación

De acuerdo a [Charytoniuk et al., 2000], por su aplicación se pueden clasificar en:

Redes de Clasificación. Hacen uso del aprendizaje no supervisado para obtener patrones de clasificación a base de las entradas del problema.

Redes de predicción. Tienen como objetivo modelar una función no lineal. Otros tipos son redes de memoria asociativa, mapeo de características, reconocimiento de patrones y optimización.

2.2.8. Funciones de activación

Una neurona biológica puede estar activa (excitada) o inactiva (no excitada); es decir, que tiene un “estado de activación”. Las neuronas artificiales también tienen diferentes estados de activación; algunas de ellas solamente dos, al igual que las biológicas, pero otras pueden tomar cualquier valor dentro de un conjunto determinado [Match, 2001]. La función de activación devuelve una salida que será generada por la neurona dada una entrada o conjunto de etradas. Cada capa que es parte de la red neuronal tiene una función de activación que permite predecir o construir un evento [Torres, 1994].

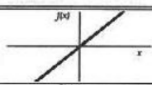
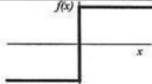
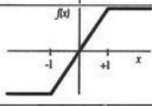

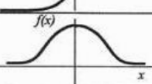
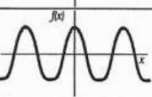
	Función	Rango	Gráfica
Identidad	$y = x$	$[-\infty, +\infty]$	
Escalón	$y = \text{sign}(x)$ $y = H(x)$	$\{-1, +1\}$ $\{0, +1\}$	
Lineal a tramos	$y = \begin{cases} -1, & \text{si } x < -l \\ x, & \text{si } -l \leq x \leq l \\ +1, & \text{si } x > l \end{cases}$	$[-1, +1]$	
	Función	Rango	Gráfica
Sigmoidea	$y = \frac{1}{1 + e^{-x}}$ $y = \text{tgh}(x)$	$[0, +1]$ $[-1, +1]$	
Gaussiana	$y = Ae^{-Bx^2}$	$[0, +1]$	
Sinusoidal	$y = A \text{sen}(ax + \varphi)$	$[-1, +1]$	

Figura 7: Algunas funciones de activación .

Fuente: [Nacelle and Mizraji, 2009].

Función lineal o identidad Se utiliza para hacer que la entrada sea igual a la salida. Por lo tanto, si tiene una red neuronal multicapa y aplica una función lineal, se denomina regresión lineal. Esta función de activación lineal se usa cuando se requiere regresión lineal en la salida. De esta forma, la red neuronal a la que se aplica la función produce un único valor [López et al., 2008].

Función Escalón Esta función se usa cuando se quiere clasificar o cuando se tiene salidas categóricas .

Función Sigmoide Se usa en la última capa para clasificar datos en dos categorías. No es una función muy utilizada debido a que no está centrada y esto afecta en el aprendizaje y entrenamiento de la neurona por lo que influye con el problema de desaparición de gradiente [Saucedo de Miguel, 2020] y [Llano et al., 2007].

Cada neurona de la red tiene su función de activación y normalmente todas las neuronas se suelen disponer con la misma función de activación, dos de las funciones más conocidas son la sigmoide y tangente hiperbólica como se puede observar en la figura 8.

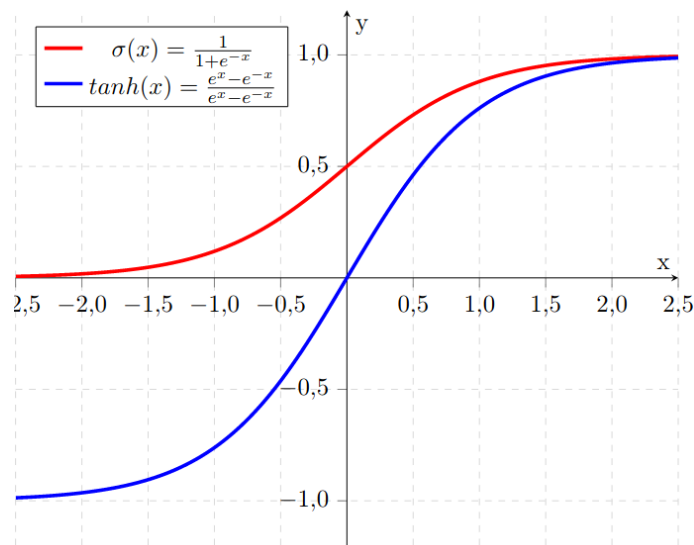


Figura 8: Función de activación tangente hiperbólica y sigmoide

Fuente: [Nebauer, 1998].

Una de las ventajas de estas dos funciones es que su derivada se puede expresar en términos de ella misma. La función tangente hiperbólica tiene el problema de vanishing gradients o desvanecimiento de gradiente que aparece en redes que encadenan muchas capas [Grosse, 2017], la curva se asemeja mucho a una línea horizontal por lo que la derivada en estos puntos será muy cercana a cero. Debido a que en retropropagación se utiliza la regla de la cadena para calcular los gradientes, estos valores cercanos a cero

de la derivada se irán acumulando multiplicativamente haciendo que el gradiente se haga cada vez más y cercano a cero, sobretodo en las primeras capas de la red. Debido a que la precisión de los ordenadores actuales no es infinita, muchas veces un valor tan cercano a cero, acabará siendo representado como cero en un ordenador, haciendo que el gradiente desaparezca por completo [Román Morales, 2018]. Por sus características la función de activación tangente hiperbólica es la más recomendable para redes neuronales recurrentes.

2.3. Principales librerías de Python utilizadas en Machine Learning

A continuación se detallará la utilidad de las librerías utilizadas según [Martínez, 2020].

- **Pandas:** es una de las librerías de python más útiles para los científicos de datos. Las estructuras de datos principales en pandas son Series para datos en una dimensión y DataFrame para datos en dos dimensiones. Destaca por la facilidad del análisis y manejo de datos.
- **Numpy:**proporciona una estructura de datos universal que posibilita el análisis de datos y el intercambio de datos entre distintos algoritmos. Las estructuras de datos que implementa son vectores multidimensionales y matrices con capacidad para gran cantidad de datos. Además, esta librería proporciona funciones matemáticas de alto nivel que operan en estas estructuras de datos.
- **SciPy:** Está formada por una serie de módulos que proporcionan funcionalidad para resolver el cómputo de tareas científicas y anañíticas. Entre estos módulos se puede encontrar soluciones de algebra lineal, integración, optimización, interpolación, procesamiento de señales, procesamiento de imagen, etc. La estructura de datos utilizada por SciPy se trata de arrays multidimensionales proporcionados por el módulo numpy.
- **Matplotlib:** es la librería gráfica de python estándar y la más conocida. Se utiliza para generar gráficos de calidad, crear muchos tipos de gráficos: series temporales, histogramas, espectros de potencia, diagramas de barras, diagramas de errores, etc.
- **Keras:**es un interfaz de alto nivel para trabajar con redes neuronales.
- **Tensorflow:**es una librería de python, desarrollada por Google, para realizar cálculos numéricos mediante diagramas de flujo de datos. Esto puede chocar un poco al principio, porque en vez de codificar un programa, codifica grafos. Los nodos

de este grafo serán operaciones matemáticas y las aristas representan los tensores (matrices de datos multidimensionales). Con esta computación basada en grafos, TensorFlow puede usarse para deep learning y otras aplicaciones de cálculo científico.

2.4. Pronósticos

Para [Nojek et al., 2003] un proceso que consiste en utilizar datos históricos para determinar acontecimientos futuros. Los pronósticos a menudo son utilizados para poder predecir la demanda del consumidor de productos o servicios, aunque se pueden predecir una amplia gama de sucesos futuros que pudieran de manera potencial influir en el éxito. Pronosticar es el arte y la ciencia de predecir los eventos futuros, puede involucrar el manejo de datos históricos para proyectarlos al futuro, mediante algún tipo de modelo matemático, puede ser una predicción del futuro subjetiva, intuitiva o bien una combinación de ambas [Toscano Hernandez et al., 2019].

2.4.1. Pronósticos de ventas y su Clasificación.

El pronóstico tiene mucha importancia dentro de la administración de ventas, en virtud de ser elemento básico de la planeación para la toma de decisiones. El pronóstico de ventas se define como un cálculo estimado de ventas para un período determinado, con el fin de preparar un plan de comercialización como lo detalla en [Mercado, 1998].

Existen 2 tipos de clasificación de pronósticos, a continuación se detalla cada uno.

- **Según el horizonte definido [Adrian, 2002].**
 - **Pronóstico a Corto Plazo:** Este tipo de pronóstico tiene un lapso de hasta un año, pero es generalmente menor a tres meses. Se utiliza para planear las compras, programación de planta, niveles de fuerza laboral, asignaciones de trabajo, niveles de producción, resumiendo decisiones operativas; por ejemplo, la demanda continua de cierto producto.
 - **Pronóstico a Mediano Plazo:** Es un pronóstico de rango mediano o intermedio, generalmente con un lapso de tres meses a tres años. Es valioso en la planeación de producción y presupuestos, planeación de ventas, presupuestos de efectivo, el análisis de varios planes de operación, resumiendo decisiones

tácticas; por ejemplo, el tiempo requerido para prever la sustitución de una línea de productos por otra nueva.

- **Pronóstico a Largo Plazo:** Generalmente con lapsos de tres años o más, los pronósticos a largo plazo se utilizan para planear nuevos productos, desembolsos de capital, localización e instalaciones, la investigación y el desarrollo, resumiendo decisiones estratégicas; por ejemplo, para decidir la ampliación de instalaciones productivas.
- **Según el tipo de pronóstico [Serna, 2003].**
 - **Pronósticos económicos:** Son aquellos pronósticos que marcan el ciclo del negocio al predecir las tasas de inflación, oferta de dinero, nuevas construcciones y otros indicadores de planeación.
 - **Pronósticos tecnológicos:** Éstos tienen que ver con las tasas de progreso tecnológico, que pueden dar por resultado el nacimiento de productos novedosos que requieren nuevas plantas y equipos.
 - **Pronósticos de demanda:** Son proyecciones de la demanda para los productos o servicios de una compañía. Estos pronósticos, también llamados pronósticos de ventas, conducen la producción de una compañía, la capacidad y los sistemas de programación, y sirven como insumos a la planeación financiera de mercado y de personal.

2.5. Empresa Impactex y sus productos:

La empresa Corporacion Impactex Cia. Ltda. se dedica a hilatura y fabricación de hilados e hilos para tejedura o costura, para el comercio o para procesamiento posterior, texturización, retorcido, plegado, cableado y remojo de hilaturas filamentosas de toda clase de fibras animales, vegetales, sintéticas o artificiales. La empresa maneja varios modelos y líneas de productos dentro del mercado de ropa interior, en el catálogo vigente para el 2019, se observa una gran gama de modelos a disposición de la clientela a nivel nacional. Las líneas de productos van acorde a la necesidad del mercado, depende del tipo de tela, actividad que realiza, edad y sexo del cliente, etc, estas líneas de productos se detallan a continuación.

- **Líneas de productos ofertados Línea Slim:** Como se observa en la figura 9 estos productos están diseñados para los deportistas, ya que está elaborada de tela licra

que absorbe el sudor y es altamente resistente para actividades físicas muy exigentes, adaptándose al cuerpo del deportista y al movimiento que realiza.



Figura 9: Línea de productos Mao Slim

Fuente: Departamento de Ventas Catalogo vigentes 2019.

- **Línea MAO Jr:** Estos productos van dirigidos para jóvenes de 12-16 años de edad, puede ser elaborado de tela licra o algodón, algunos productos de esta línea se observa en la figura 10.



Figura 10: Línea de productos Mao Jr

Fuente: Departamento de Ventas Catalogo vigentes 2019.

- **Línea MAO Underaction:** Este tipo de producto va dirigido para todo el público adulto de clase media y baja, está elaborada en tela de algodón con estampados,

además se caracteriza por los colores vivos se los aprecia en la figura 11 tomada del catálogo vigente del 2019.

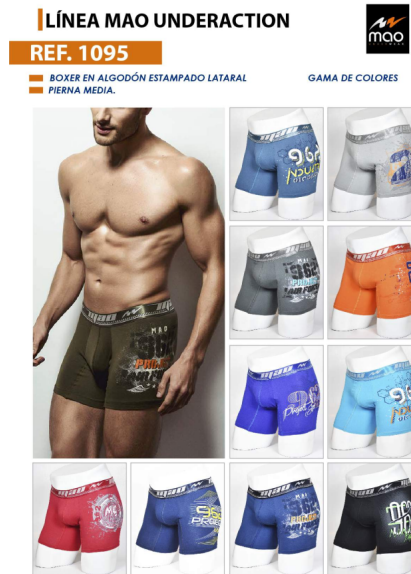


Figura 11: Línea de productos Mao Underaction
Fuente: Departamento de Ventas Catalogo vigentes 2019.

- **Línea MAO Kids:** Productos dirigidos a niños de 5 a 10 años de edad, como se muestra en la figura 12, algunos de los diseños que ofrece esta línea, puede ser elaborado de tela licra o algodón.



Figura 12: Línea de productos Mao Kids
Fuente: Departamento de Ventas Catalogo vigentes 2019.

Código de los Productos Ofertados


Para asignar un código a cada producto se utiliza las letras iniciales o letras que se relacionen con el producto para identificarlos, tal como se muestra en la tabla 1, además se utiliza números que permiten distinguir las variantes de modelos que pertenecen a una misma línea de producción, estos nunca se repiten, ya que varían por el tipo de tela, color y detalles.

Tabla 1: Descripción de códigos de productos.

Código	ITEM	Código	ITEM
BH	Bóxer de hombre	BN	Bóxer de niño
BO	Buso	CH	Camiseta de hombre
CO	Camiseta de cuello redondo	CI	Calzoncillo
BB	Bikini	PN	Pantalón de hombre
PC	Panty	CV	Camiseta de cuello en v
BJ	Brasier	JT	Tanga
CD	Conjunto deportivo	SH	Short
VH	Bividi de hombre	CN	Camiseta de niño
TM	Top de mujer	HH	Hilo
BI	Bicicletero	TS	Terno de baño

Para entender mejor la asignación de código, se muestra la tabla 2 en dónde se realiza un ejemplo con el código de cierto producto y la descripción de lo que significa cada letra y número.

Tabla 2: Ejemplo de descripción de códigos.

BH 7060-38			
BH: Bóxer de hombre	7060 Se refiere a la línea de producción Mao Slim	38 Hace referencia a la talla del producto	

Sin embargo, la empresa tiene diversas líneas de producción, entre ellas se puede nombrar ropa deportiva de mujer, pijamas, camisetas deportivas de hombre, mujer y niños. Además de la línea de ropa interior para adultos, jóvenes y niños, ofrece la línea deportiva enfocada en deportistas de élite, por lo que el código de productos que manejan es extenso. Para realizar el análisis del producto de mayor impacto se necesita detallar el código de todos los productos existentes, lo que se describe a continuación en la tabla 3.

Tabla 3: Tabla códigos de todos los productos ofertados.

Códigos de productos									
Nº	Código	Nº	Código	Nº	Código	Nº	Código	Nº	Código
1	BH 1060	27	BH 1095	53	BN 6618	79	CH 7523	105	CN 174
2	BH 7021	28	BH 7065	54	1111.5	80	CI 580	106	TS 673
3	BH 7060	29	BH 7072	55	BH 7067	81	BH 1073	107	CV 635
4	BH 1056	30	BB 1055	56	CH 185	82	5112.2	108	VH 4101
5	BH 1071	31	BH 6928	57	1111.1	83	BH 201701	109	5112.12
6	BH 6916	32	BH 6928	58	12714.3	84	5112.3	110	VH 470
7	BO 460	33	BH 7033	59	BH 303	85	CV 630	111	CD 9507
8	CO 585	34	1111.6	60	BH 302	86	BO 412	112	23352.1
9	BH 7024	35	1111.6	61	1111.10	87	BH 7059	113	JT 6418
10	BH 1056	36	BN 6611	62	1111.4	88	CO 635	114	SH 702
11	BH 6930	37	BH 7034	63	BH 7068	89	12412.1	115	BN 6518
12	BH 1061	38	1111.12	64	BH 7066	90	BH 7069	116	PN 8910
13	BH 1072	39	1111.7	65	1112.2	91	BH 7064	117	TS 635
14	BH 7053	40	BN 6612	66	BO 470	92	5112.1	118	TS 630
15	CO 580	41	BH 1062	67	1111.13	93	1112.6	119	CO 630
16	1112.3	42	CH 170	68	BJ 6633	94	VH 460	120	CO 620
17	BH 7030	43	1111.9	69	BH 7058	95	1111.11	121	5112.8
18	BH 7073	44	BH 7062	70	BH 7063	96	5112.4	122	CH 7526
19	BH 6925	45	CI 585	71	BO 413	97	BN 202	123	TS 580
20	BH 1065	46	BI 460	72	PN 8913	98	5112.5	124	VH 5103
21	BO 410	47	BH 201703	73	CD 9601	99	5112.7	125	HH 7525

Códigos de productos									
Nº	Código	Nº	Código	Nº	Código	Nº	Código	Nº	Código
22	TS 659	48	23152.1	74	5112.9	100	BH 7071	126	VH 412
23	CH 7521	49	TM 8826	75	VH 413	101	BH 6951	127	TS 640
24	BH 7070	50	BH 7004	76	TS 671	102	BH 7019		
25	3112.1	51	PC 9001	77	VH 410	103	BH 201702		
26	PN 8915	52	TS 585	78	CN 171	104	CH 7522		

3. CAPÍTULO III

MARCO METODOLÓGICO

3.1. Ubicación.

La simulación del pronóstico de venta en la empresa IMPACTEX mediante Redes Neuronales Artificiales se lo realizó con datos proporcionados por la empresa Ubicada en la provincia de Tungurahua ciudad Ambato en las calles 22 de Enero Av Circunvalacion.

3.2. Equipos y materiales.

Hardware:

- Computador portátil intel core i7

Software:

- Python 3.7.
- Pycharm IDE.

3.3. Tipo de investigación.

3.3.1. Investigación bibliográfica.

El presente trabajo se basó en la investigación bibliográfica, ya que, su característica es utilizar datos auxiliares como fuente de información. Con el objetivo de encontrar soluciones a los problemas planteados. Entre las fuentes de información que se utilizará tenemos: libros, revistas, documentos doctorales, artículos científicos, tesis de grado. Los cuales ayudarán a dar cumplimiento con los objetivos planteados para el desarrollo de este trabajo.

3.3.2. Investigación aplicada.

Es de carácter aplicativo ya que busca la generación de conocimiento con aplicación directa a un específico sector tecnológico. Además busca la utilización de conocimientos de varias áreas especializadas de la ingeniería como las Matemáticas, Electrónica y la Programación, con el propósito de implementarlos de forma práctica para desarrollar un pronóstico de venta mediante redes neuronales artificiales.

3.4. Idea a defender.

Es factible simular un pronóstico de venta en la empresa IMPACTEX mediante Redes Neuronales Artificiales.

3.5. Población o muestra.

Por el tipo de investigación empleado para desarrollar este proyecto no requiere de una muestra ya que se trabajó con todos los datos disponibles y proporcionados por la empresa IMPACTEX.

3.6. Recolección de información.

Para el trabajo desarrollado, la recopilación de información se realizó mediante la selección de información relevante, que se obtiene de fuentes como libros, revistas, artículos científicos, tesis doctorales y la propia Internet, información que proporciona materiales para el desarrollo de la investigación. Mientras que los datos utilizados para el desarrollo del pronóstico lo proporcionó la empresa IMPACTEX, obteniendo una base de datos de 12 años con información de todas las ventas de los 127 productos ofertados.

El software con el que se desarrolló el trabajo de investigación se lo obtuvo de las páginas: IDE Pycharm (<https://www.jetbrains.com/es-es/pycharm/>) Python3.7 (<https://www.python.org/>)

En el desarrollo se necesitó de algunas librerías como NumPy, SciPy y Pandas librerías originales de python ideales para análisis de datos y computación numérica. Mientras que para las Redes Neuronales Artificiales se trabajó con librerías como Keras y Tensorflow.

3.7. Procesamiento de la información y análisis estadístico.

Se utilizó datos históricos para realizar el entrenamiento y testeo de las redes neuronales artificiales, destinando un grupo de datos llamado datos entrenamiento, y, para verificar si dicho entrenamiento es óptimo se obtuvo un grupo de datos llamado datos testeo. Para realizar el análisis de la simulación se obtiene el error entre los datos reales y el pronóstico realizado.

3.8. Resultados alcanzados.

Se logró la simulación de pronósticos de venta en la empresa IMPACTEX mediante redes neuronales artificiales, por medio del software libre Python y utilizando los datos

históricos de ventas de la empresa IMPACTEX de los años 2007 al 2019, estos datos se dividieron en dos grupos el primero denominado de entrenamiento con un 75 % de datos totales, y el otro grupo llamado datos de testeo con un 25 % del total de datos. La estructura de la red neuronal que permitió obtener el pronóstico de ventas se obtuvo luego de variar los parámetros de epoch, número de capas y neuronas de la red neuronal, obteniendo como resultado un conjunto de 24 entradas, tres capas internas y una capa de salida. El pronóstico realizado se hizo en base a los productos que tienen mayor demanda, para lo cual se realizó un análisis ABC obteniendo los siguientes resultados: el producto con mayor demanda es el BH1060 siendo un producto tipo A, en segundo lugar el producto BH1070 también perteneciente al grupo A, en el grupo B el producto C585 y en el grupo C el producto 1112.6. Los parámetros seleccionados generaron un error de 2,60 % para el producto tipo A (BH1060), para el producto tipo A (BH1070) 3,64 % , para el producto tipo B (C585) 3,02 % y un error del 3,27 % para el producto C (1112.6).

4. CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

4.1. Análisis de resultados.

4.1.1. Desarrollo de la propuesta.

En el presente capítulo se detalla el procedimiento realizado para la implementación de una "Simulación de pronósticos de ventas en la empresa IMPACTEX mediante redes neuronales", así como también los resultados obtenidos mediante el modelo de pronósticos realizado. Para la construcción del modelo se presenta en la figura 13 el esquema que se siguió como metodología para realizar el pronóstico de ventas.

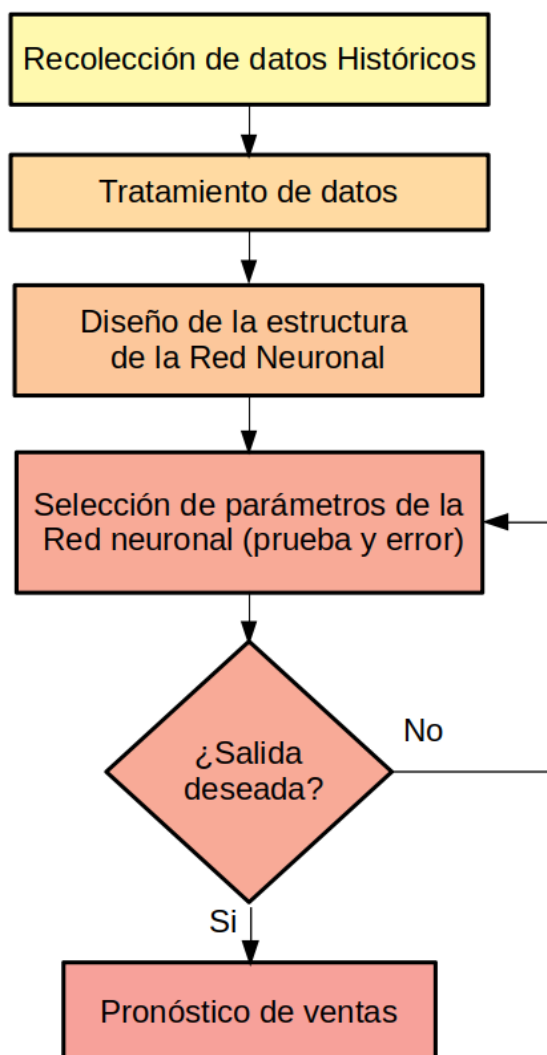


Figura 13: Esquema de la metodología utilizada para el pronóstico de ventas

Fuente: Elaborado por la autora.

4.1.2. Recolección y tratamiento de datos históricos.

La recolección de datos es una parte importante al momento de trabajar con Redes Neuronales, debido a que es la información que servirá posteriormente para obtener un resultado eficaz o no. Los datos recolectados serán destinados para dos propósitos, el primero que será un grupo utilizado para realizar el diseño y construcción de la Red Neuronal, de la cual depende el aprendizaje y que tan acertado es el pronóstico que se obtendrá de la Red Neuronal, mientras que el segundo propósito es igual de importante, aunque en cantidad es un grupo más pequeño de datos y está destinado al testeo de la red, es decir, se utilizará como referencia para validar el resultado obtenido para realizar el pronóstico, ya que se está probando la estimación contra la realidad.

De los datos otorgados por la empresa se detalla a continuación la información registrada.

- Año, mes, día en el que se realizó la venta
- Provincia y ciudad a la que se destinó el producto vendido.
- Código del producto vendido.
- Descripción de la cantidad vendida (unidades o docenas).
- Valor a pagar por la cantidad de producto vendido
- Asesor que atendió la venta.
- Nombre del cliente.

Para el tratamiento de datos se debe filtrar la información útil, ya que toda la información recibida no es de utilidad, para lo cual se realizó algunos cambios que se enumeran a continuación.

1. De la matriz inicial se identificó los indicadores clave, es decir los datos que proporcionarán la información a la red neuronal para su entrenamiento y posterior testeo. En este caso los datos que formarán parte de la nueva matriz de información serán:

- **Año**
- **Código de producto**
- **Cantidad de la venta**

2. Una vez filtrados los datos necesarios como se muestra en la figura 14, se debe uniformizar la cantidad ya que esta expresado en unidades y docenas. Se trabajará solo con docenas.

CORPORACIÓN IMPACTEX			
FECHA: 07/01/2020			
año	Código	Uni.	Cantidad
2019	1111.12	DOCENA	6.00
2019	BB 1055	DOCENA	10.00
2019	BH 1065	DOCENA	33.00
2019	BH 7053	DOCENA	19.00
2019	CI 580	DOCENA	.17
2019	CO 585	DOCENA	13.00
2019	CO 585	DOCENA	.25
2019	CO 585	DOCENA	25.00
2019	BH 1060	DOCENA	13.00
2019	1112.2	DOCENA	.50
2019	BH 1072	DOCENA	1.00

Figura 14: Datos filtrados de la empresa IMPACTEX

Fuente: Elaborado por la autora.

3. La empresa ofrece hasta el año 2019 en total 127 productos, y para esta investigación se realizará el pronóstico de los productos más vendidos para lo cual se realizará un análisis ABC, el cual determinará los productos que se utilizarán para el pronóstico.

Análisis ABC Según [Peiró, 2017] en su artículo indica que el análisis ABC es un sistema que se utiliza para diseñar la distribución de inventarios en almacenes. El objetivo de esta metodología es optimizar la organización de los productos de tal forma que los más atractivos y solicitados por el público se encuentren a un alcance más directo y rápido. Con ello se reducen los tiempos de búsqueda y se aumenta la eficacia.

En el análisis ABC, se clasifican los distintos productos en 3 tipos teniendo en cuenta su importancia como se detalla a continuación:

- **Tipo de artículos A:** Son los más importantes, usados o vendidos. Así mismo, son los que más ingresos generan.

- **Tipo de artículos B:** Tienen una importancia secundaria y los ingresos generados son menores en relación con los artículos A.
- **Tipo de artículos C:** Su importancia es mínima y reportan poco beneficio.

Los artículos de tipo A deben ser sometidos a un estricto control de inventario. Dada la importancia de éstos, deben contar con áreas de almacenamiento mejor aseguradas para su mantenimiento y accesibilidad. Evitar que se agote su stock. Los artículos de tipo B suponen una clasificación intermedia entre los productos A y C. Es importante ejercer un control sobre ellos para determinar si tienden a cambiar de categoría. Mientras que los productos tipo C presentan una baja demanda con un mayor riesgo de costes de inventario excesivos.

Para obtener los productos de mayor impacto se parte de un histórico de ventas, del cual se inicia obteniendo el porcentaje de participación monetaria de todos los productos fabricados en la empresa, para lo cual se emplea la ecuación **1**.

$$\%p = \frac{100}{P} \quad (1)$$

Donde:

p=es la participación monetaria

P=es el número de productos

$$\%p = \frac{100}{127}$$

$$\%p = 0,79$$

Se plantea como valorización, al monto total de ventas por docena en el año de cada producto elaborado en la empresa; para determinar el porcentaje de consumo de cada producto en el año se utiliza la ecuación **2**.

$$\%p = \frac{\text{valor} * 100}{v} \quad (2)$$

Donde:

v=es la valorización total

Una vez establecidos los valores de porcentaje de participación se ordenan de manera descendente (mayor a menor), se aplica las ecuaciones **3** y **4** para obtener el porcentaje de participación acumulada y porcentaje de consumo acumulado respectivamente.

$$\%pa = \%pa_{i-1} + \%pa_i \quad (3)$$

$$\%ca = \%ca_{i-1} + \%ca_i \quad (4)$$

Donde:

pa=es la participación acumulada

ca=es el consumo acumulado

Luego de obtener los valores utilizando las ecuaciones de valorización y participación para todos los productos, se procede a etiquetar con los porcentajes obtenidos el grupo al que pertenece cada producto A, B o C. Un resumen de esto se puede observar en la tabla 4, que muestra los 5 primeros productos de cada grupo.

Tabla 4: Resumen de clasificación ABC

CODIGO	%CONSUMO	%CONSUMO ACUMULADO	%PARTIC ACUM	CLASIFICACIÓN
BH 1060	16.27 %	16.27 %	0.79	A
BH 7021	6.49 %	22.76 %	1.57	A
BH 7060	5.38 %	28.14 %	2.36	A
BH 1056	3.32 %	31.45 %	3.15	A
BH 1071	2.94 %	34.40 %	3.94	A
CI 585	0.67 %	80.54 %	30.71	B
BI 460	0.61 %	81.15 %	31.50	B
BN 6618	0.61 %	81.76 %	32.28	B
1111.5	0.60 %	82.36 %	33.07	B
BH 7067	0.59 %	82.95 %	33.86	B
1112.6	0.27 %	95.06 %	58.27	C
VH 460	0.26 %	95.32 %	59.06	C
1111.11	0.26 %	95.58 %	59.84	C
5112.4	0.25 %	95.82 %	60.63	C
BN 202	0.24 %	96.07 %	61.42	C

Con base a los datos obtenidos del análisis ABC se procede a trabajar el pronóstico utilizando dos productos A que corresponde a los códigos BH 1060 y BH 7021, un producto B con código CI 585 y un producto C que refiere al producto 1112.6.

Una vez identificados los productos con los que se trabajará en el pronóstico, se procede a obtener la información de la base de datos, para lo cual se tiene los datos que se presenta en la tabla 5, se detalla la información de las ventas necesaria para el desarrollo de la Red Neuronal.

Tabla 5: Histórico de ventas de los años 2007 y 2008 de la empresa IMPACTEX

Venta por código de producto						
Año	Mes	Tiempo	BH1060	BH7021	CI585	1112.6
2007	1	1	243.7	112.0	95.2	30.1
	2	2	256.8	118.2	100.3	31.7
	3	3	287.2	132.3	112.2	35.5
	4	4	280.7	129.3	109.7	34.7
	5	5	263.3	121.2	102.9	32.6
	6	6	293.8	135.3	114.8	36.3
	7	7	322	148.3	125.8	39.8
	8	8	311.2	118.3	115.3	37.6
	9	9	295.9	136.3	115.6	36.6
	10	10	258.9	119.2	101.2	32.0
	11	11	226.3	104.2	88.4	28.0
	12	12	256.8	118.2	100.3	31.7
2008	1	13	250.2	115.2	97.8	34.4
	2	14	274.2	126.2	107.1	37.6
	3	15	306.8	141.3	119.9	42.1
	4	16	293.8	135.3	114.8	40.3
	5	17	272	125.2	106.3	37.3
	6	18	324.2	149.3	126.7	44.5
	7	19	369.9	170.3	144.5	50.8
	8	20	359.7	160.1	124.2	44.4
	9	21	343.8	158.3	134.3	47.2
	10	22	289.4	133.3	113.1	39.7
	11	23	248.1	114.2	96.9	34.1
	12	24	304.6	140.3	119	41.8

Los datos de entrada fueron obtenidos de los históricos de ventas de los años 2007-2016 que se convirtieron en datos de entrada para implementar la red neuronal. Este grupo de datos se los llamará datos de entrenamiento, mientras que para evaluar la validez de la Red Neuronal se tomará los históricos de ventas de los años 2017-2019, a este grupo de datos se los llamará datos de testeo.

De acuerdo a [Malik, 2019](#) en su artículo, las redes neuronales con modelos secuenciales tiene 4 tipos de problemas que se numeran a continuación.

1. **Uno a uno:** donde hay una entrada y una salida. Un ejemplo típico de problemas de secuencia uno a uno es el caso en el que tiene una imagen y desea predecir una sola etiqueta para la imagen.
2. **Muchos a uno:** en los problemas de secuencia de muchos a uno, tenemos una secuencia de datos como entrada y tenemos que predecir una única salida.
3. **Uno a muchos:** en los problemas de secuencia de uno a muchos, tenemos una sola entrada y una secuencia de salidas. Un ejemplo típico es una imagen y su descripción correspondiente.
4. **Muchos a muchos :** los problemas de secuencia de muchos a muchos implican una entrada de secuencia y una salida de secuencia.

En la figura [15](#)(a) se puede observar las entradas de la red neuronal, una vez que los datos fueron normalizados. Este array se denomina X_train, y consta de 24 filas y 82 columnas. Es decir el número de entradas de la red neuronal es 24 y consta de 1 salida, según los criterios de [Malik, 2019](#) sería un problema secuencial de muchos a uno.

X_train								
	0	1	2	3	4	5	6	7
0	0.01596	0.02748	0.05408	0.04876	0.03280	0.06028	0.08511	0.08511
1	0.02748	0.05408	0.04876	0.03280	0.06028	0.08511	0.08511	0.06206
2	0.05408	0.04876	0.03280	0.06028	0.08511	0.08511	0.06206	0.02926
3	0.04876	0.03280	0.06028	0.08511	0.08511	0.06206	0.02926	0.00000
4	0.03280	0.06028	0.08511	0.08511	0.06206	0.02926	0.00000	0.02748
5	0.06028	0.08511	0.08511	0.06206	0.02926	0.00000	0.02748	0.02128
6	0.08511	0.08511	0.06206	0.02926	0.00000	0.02748	0.02128	0.04255
7	0.08511	0.06206	0.02926	0.00000	0.02748	0.02128	0.04255	0.07181
8	0.06206	0.02926	0.00000	0.02748	0.02128	0.04255	0.07181	0.06028
9	0.02926	0.00000	0.02748	0.02128	0.04255	0.07181	0.06028	0.04078
10	0.00000	0.02748	0.02128	0.04255	0.07181	0.06028	0.04078	0.08688
11	0.02748	0.02128	0.04255	0.07181	0.06028	0.04078	0.08688	0.12766
12	0.02128	0.04255	0.07181	0.06028	0.04078	0.08688	0.12766	0.12766
13	0.04255	0.07181	0.06028	0.04078	0.08688	0.12766	0.12766	0.10461
14	0.07181	0.06028	0.04078	0.08688	0.12766	0.12766	0.10461	0.05585
15	0.06028	0.04078	0.08688	0.12766	0.12766	0.10461	0.05585	0.01950
16	0.04078	0.08688	0.12766	0.12766	0.10461	0.05585	0.01950	0.07004
17	0.08688	0.12766	0.12766	0.10461	0.05585	0.01950	0.07004	0.07979

(a) Datos de entrenamiento x_train

X_test									
	0	1	2	3	4	5	6	7	
0	0.45567	0.41312	0.49823	0.47074	0.50000	0.63918	0.74645	0.77394	0.
1	0.41312	0.49823	0.47074	0.50000	0.63918	0.74645	0.77394	0.57890	0.
2	0.49823	0.47074	0.50000	0.63918	0.74645	0.77394	0.57890	0.49202	0.
3	0.47074	0.50000	0.63918	0.74645	0.77394	0.57890	0.49202	0.39805	0.
4	0.50000	0.63918	0.74645	0.77394	0.57890	0.49202	0.39805	0.44947	0.
5	0.63918	0.74645	0.77394	0.57890	0.49202	0.39805	0.44947	0.49379	0.
6	0.74645	0.77394	0.57890	0.49202	0.39805	0.44947	0.49379	0.45922	0.
7	0.77394	0.57890	0.49202	0.39805	0.44947	0.49379	0.45922	0.58333	0.
8	0.57890	0.49202	0.39805	0.44947	0.49379	0.45922	0.58333	0.56383	0.
9	0.49202	0.39805	0.44947	0.49379	0.45922	0.58333	0.56383	0.60993	0.
10	0.39805	0.44947	0.49379	0.45922	0.58333	0.56383	0.60993	0.71011	0.

(b) Datos para testeo x_test

y_train									
	0	1	2	3	4	5	6	7	
0	0.07979	0.08865	0.14273	0.11436	0.13121	0.14273	0.18351	0.18351	0.

(c) Etiquetas de resultados esperados y_train

y_test									
	0	1	2	3	4	5	6	7	
0	0.60372	0.55408	0.60816	0.68883	0.71011	0.83156	1.00000	0.96897	0.

(d) Etiquetas de resultados y_test

Figura 15: Ejemplo de la estructura de matrices de datos para entrenamiento y testeo.

Fuente: Elaborado por la autora.

Los datos de entrenamiento se basan en datos históricos que inicia en el año 2007, para lo cual se toma como referencia los dos primeros años es decir el año 2007 y 2008 servirán como base para iniciar el entrenamiento de la red, por lo que se tomará grupos de 24 datos, que representan los 24 meses que se tomaron como referencia inicial.

Para definir el número correcto de datos de entrenamiento, estos deben cumplir las siguientes características.

- Que el conjunto sea lo suficientemente grande como para generar resultados significativos.
- Que sea representativo de todo el conjunto de datos. No elegir un conjunto de prueba con características diferentes al conjunto de entrenamiento.

El porcentaje que se reserva para el conjunto de entrenamiento oscila entre el 70 % al 80 % mientras que el de testeo va del 30 % al 20 % respectivamente, como lo sugiere [Bagnato, 2018]. En este caso se asignó 75 % de los datos para entrenamiento, y el resto 25 % para testeo. Este proceso se puede observar en el siguiente código.

```
training_size = int(len(df1)*0.75)
test_size = len(df1)-training_size
```

Las redes neuronales con una gran cantidad de parámetros son sistemas de aprendizaje automático muy poderosos. Sin embargo, el sobreajuste es un problema grave en este tipo de redes [Srivastava et al., 2014]. La importancia de este procedimiento es evitar la existencia de un sobreajuste o subajuste de la red neuronal, [Marín Marchant, 2020] explica lo siguiente.

Sobreajuste u overfitting es un fenómeno que ocurre a medida que se van generando más iteraciones del algoritmo, donde el modelo se acostumbra tanto a los datos de entrada que la precisión de los resultados deja de ser la esperada, ya que el objetivo del modelo es que interactúe de manera precisa para cualquier conjunto de datos.

Subajuste o underfitting es cuando los resultados del modelo no mejoran a pesar de tener capacidad de mejorar, causas de esto típicamente se asocian a que el modelo no ha sido entrenado lo suficiente o no es lo suficientemente poderoso para completar la tarea. Para combatir el sobreajuste lo mejor es tener más datos de entrenamiento, mientras más datos se tengan, mejor aprenderá el modelo.

Por esta razón los datos de 9 meses serán utilizados en el Data Training, mientras que los 3 últimos meses serán usados en el Data Test, de dónde se obtendrá las siguientes matrices de datos.

- x_train registros para entrenar
- y_train con las “etiquetas” de los resultados esperados de X_train
- x_test registros para test
- y_test con las “etiquetas” de los resultados de X_test

4.1.3. Implementación de la red neuronal artificial.

Una red neuronal es un sistema que permite establecer una relación lineal o no lineal entre las salidas y las entradas. Sus características están inspiradas en el sistema nervioso por lo que poseen varias ventajas como: capacidad de aprendizaje adaptativo, son auto-organizadas, funcionan en paralelo a tiempo real, tolerancia a fallos [Singhal and Swarup, 2011].

Segun [Villanueva Espinoza, 2002] la arquitectura de una red neuronal consiste en la organización y disposición de las neuronas en la red formando capas o agrupaciones de neuronas. Los parámetros fundamentales son: número de capas, número de neuronas por capa, grado de conectividad y tipo de conexión entre neuronas. Mientras que, las fuerzas de las conexiones entre neuronas, conocidas como pesos sinápticos, son usadas para almacenar el conocimiento [Simon et al., 2009]. Uno de los aspectos fundamentales cuando se va enfrentar un proyecto con redes neuronales es disponer de una base de datos histórica de la variable, lo suficientemente grande para garantizar que el proceso de entrenamiento sea mucho más confiable [Villada et al., 2008].

Para iniciar con el diseño de la Red Neuronal primero se necesita del uso de ciertas librerías que permitirán el manejo de datos.

```
import pandas_datareader as pdr
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.preprocessing import MinMaxScaler
import numpy
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import LSTM
import tensorflow as tf
from sklearn.metrics import mean_squared_error
```

```
from numpy import array
import os.path
```

Para poder programar usando código Python se necesita un entorno. Uno de los más usados y el que se ha empleado en este trabajo es Pycharm. Entre sus características se encuentran:Asistencia y análisis de código:Con herramientas como sugerencias para complementación de código o destacando errores de sintaxis. Esto permite que aunque el usuario sea relativamente nuevo programando en ese lenguaje no tenga que estar constantemente buscando sus propios errores.Navegación de proyectos y código:Vistas de proyecto especializadas, vistas de estructuras de archivos o facilidad para saltar entre archivos, clases y métodos [Agustín Granell, 2018].

Redes neuronales recurrentes

Los seres humanos no empiezan a pensar desde cero cada segundo. No tira todo y empieza a pensar todo nuevamente. Los pensamientos tienen persistencia. Las redes neuronales tradicionales no pueden hacer esto y parece ser una deficiencia importante [Jozefowicz et al., 2015].Una red neuronal tradicional está compuesta de una función de activación que actúa en una sola dirección, hacia delante, desde la capa de entrada hacia la capa de salida, una red neuronal recurrente tiene una estructura similar pero incluye conexiones que apuntan hacia atrás con una retroalimentación.

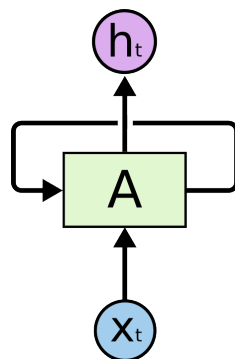


Figura 16: Estructura de una Red Neuronal Recurrente.

Fuente: [Olah, 2015].

La figura 16 muestra la estructura básica de una red neuronal recurrente, en donde x_t es la entrada a la neurona A que a su vez genera una salida h_t , el bucle de retro alimentación cumple la función de memoria de la red neuronal. Sin embargo, esto no significa que sea un tipo de red neuronal diferente en la figura 17 se muestra el interior de la estructura de una red neuronal recurrente, en donde se observa un conjunto de múltiples copias de una misma red cuya función es pasar un mensaje a su sucesor.

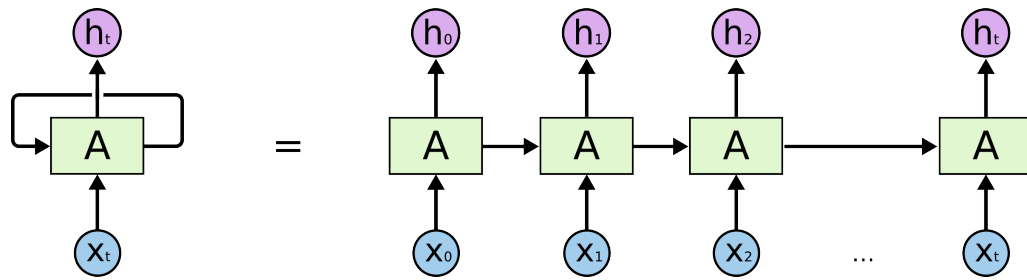


Figura 17: Estructura interna de una Red Neuronal Recurrente.

Fuente: [Olah, 2015].

4.1.4. Selección de parámetros de la red neuronal

Una vez que se declaran las librerías necesarias se procede a crear la estructura de la red neuronal, con el siguiente código.

```
model=Sequential()
model.add(LSTM(1, return_sequences=True, input_shape=(24, 1)))
model.fit(X_train_1, y_train, validation_data=(X_test_1, y_test),
epochs=5, batch_size=5, verbose=1)
```

El primer paso es iniciar el sistema con la llamada a `Sequential()`, que crea una red neuronal vacía a la que se le agrega las capas correspondientes con la instrucción `model.add()`. La siguiente capa es de tipo *LSTM*, el comando `return_sequences = True` se utiliza para que la salida del estado oculto de cada neurona se use como entrada para la siguiente capa LSTM, mientras que 1 es el número de neuronas que va tener cada capa. La segunda línea de código crea una red neuronal de una sola capa que contiene 1 neurona, número de entradas 24 y 1 salida. Para el entrenamiento se selecciona inicialmente 5 *epochs* de entrenamiento, ya que se recomienda iniciar con un número pequeño y luego ir incrementando. Un *epoch* es el número de veces que se ejecuta el algoritmo de la red neuronal. En cada ciclo (epoch) todos los datos de entrenamiento pasan por la red neuronal para que esta aprenda sobre ellos. El resultado de este modelo se observa a continuación.

```
Epoch 1/5
17/17 [=====] - 1s 23ms/step -
loss: 0.3812 - val_loss: 1.6795
Epoch 2/5
17/17 [=====] - 0s 5ms/step -
```

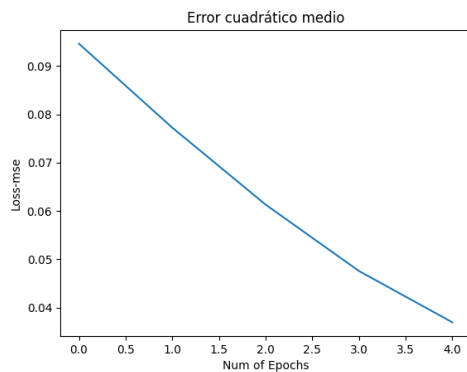
```

loss: 0.2978 - val_loss: 1.4851
Epoch 3/5
17/17 [=====] - 0s 5ms/step -
loss: 0.2341 - val_loss: 1.3191
Epoch 4/5
17/17 [=====] - 0s 5ms/step -
loss: 0.1867 - val_loss: 1.1772
Epoch 5/5
17/17 [=====] - 0s 6ms/step -
loss: 0.1487 - val_loss: 1.0684

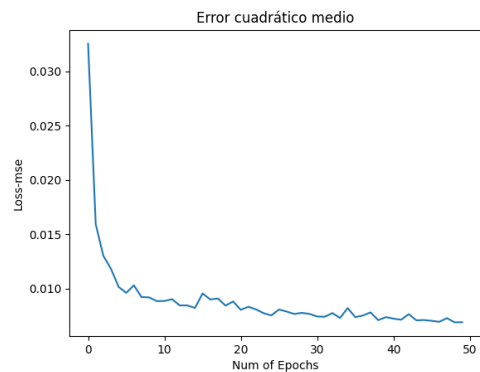
```

El parámetro *loss* es la función de pérdida y representa el error cuadrático medio, en este caso con los parámetros escogidos para la red este error es muy alto, por lo que se debe seguir afinando el modelo de la red.

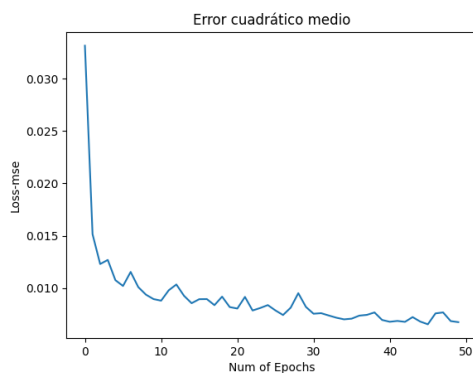
En la figura [18](#) se presenta los resultados de pruebas realizadas con distintos parámetros hasta llegar al modelo que más se ajusta a los requerimientos.



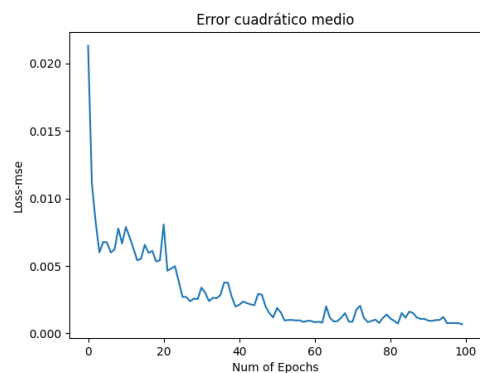
(a) MSE con epoch=5



(b) MSE con epoch=50



(c) MSE con epoch=50 y 2 capas



(d) MSE con epoch=100 y 3 capas

Figura 18: Gráfica de MSE con diferentes parámetros

En la figura 18 (a), se muestra el error cuadrático medio correspondiente a una red neuronal de una sola capa y una neurona con $epoch=5$. Se observa que el error disminuye, pero el número de iteraciones es muy pequeño y no se pudo analizar la estabilización del error. En la figura 18 (b), se muestra el error cuadrático medio correspondiente a una red neuronal de una sola capa y cincuenta neuronas con $epoch=50$. Se puede observar el comportamiento del error que tiende a disminuir pero es necesario incrementar el número de epoch para analizar su tendencia.

En la figura 18 (c y d), se muestra el error cuadrático medio correspondiente a una red neuronal de dos y tres capas respectivamente y $epoch=50$ y 100. El error se comporta de manera similar al de la figura 18 (b) pero el tiempo en el que se aproxima a cero es menor. En la tabla 7 muestra el error de entrenamiento y error de testeo para diferentes valores de $epoch$ y número de capas y neuronas. **El error calculado hace referencia a los datos originales y los datos obtenidos por la red neuronal artificial.**

Tabla 7: Error de entrenamiento y de testeo para el producto BH1060

Capas	Neuronas	epoch	Error entrena	Error test
1	5	5	29.21 %	69.76 %
1	20	10	25.49 %	65.45 %
1	50	20	18.92 %	56.85 %
1	80	50	20.72 %	31.81 %
1	100	100	60.80 %	21.87 %
2	30,20	50	20.70 %	31.84 %
2	50,40	50	20.55 %	32.03 %
2	80,60	80	43.84 %	14.84 %
2	100,90	120	78.26 %	37.08 %
3	50,40,20	50	6.96 %	14.43 %
3	50,40,20	80	4.88 %	20.62 %

Capas	Neuronas	epoch	Error entrena	Error test
3	80,60,50	100	2.72 %	13.72 %
3	100,90,50	120	3.68 %	8.80 %
3	100,90,40	150	2.57 %	7.89 %
3	100,90,40	200	2.36 %	2.69 %
4	100,90,60,50	200	7.81 %	6.29 %
4	100,90,40,30	150	10.16 %	12.76 %

Luego de buscar los parámetros que favorecen a los requerimientos, por medio de prueba y error, se obtiene el siguiente modelo que se muestra en el código que se presenta a continuación:

```

model=Sequential()
model.add(LSTM(100, return_sequences=True, input_shape=(24, 1)))
model.add(LSTM(90, return_sequences=True))
model.add(LSTM(40))
model.add(Dense(1))

```

En la figura 19, se puede ver el resultado obtenido con el código anterior, la figura 19(a) se muestra el MSE obtenido con 200 *epoch* y en la figura 19(b) se puede ver como se ajusta la curva de entrenamiento a la tendencia de los datos de entrada. La fase de aprendizaje se utiliza para ajustar los pesos, los cuales inicialmente son valores aleatorios. El valor del error cuadrático medio sirve para evaluar la corrección de pesos en cada iteración. Por último, la red neuronal artificial aprende el comportamiento por medio de los datos de entrenamiento.

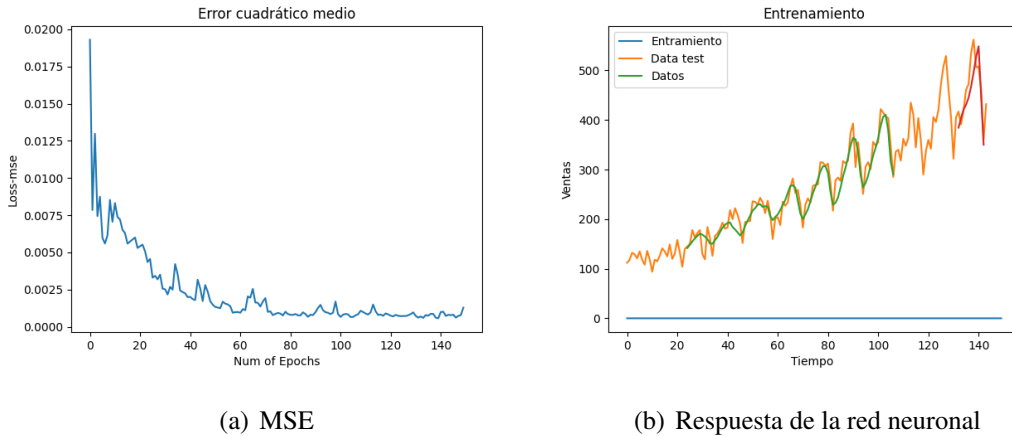


Figura 19: Resultados de los parámetros seleccionados

Fuente: Elaborado por la autora.

En la figura [20](#), se puede observar la estructura de una red neuronal artificial que consta de una capa de entrada, una capa oculta y una de salida, con esta figura se puede tener una idea gráfica de como sería el modelo generado para este trabajo en donde se tiene:

- Capa de entrada: Compuesta por 24 entradas de datos que corresponde a los meses de los dos primeros años que serán tomados como base para el desarrollo del pronóstico.
- capas ocultas: El modelo desarrollado tendrá dos capas ocultas con 90 y 40 neuronas respectivamente en cada capa.
- Capa de salida: el modelo posee una salida, por lo que, sería una red neuronal recurrente con varias entradas y una salida (problema varios a uno).

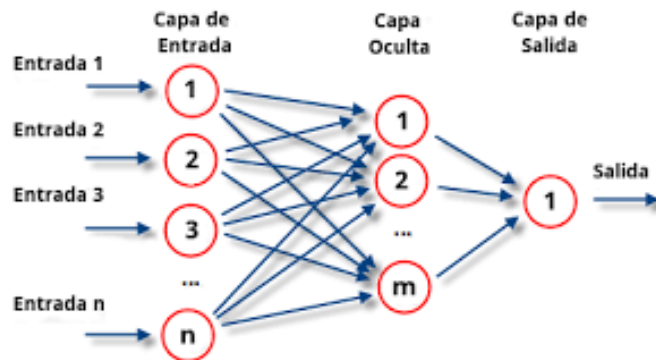
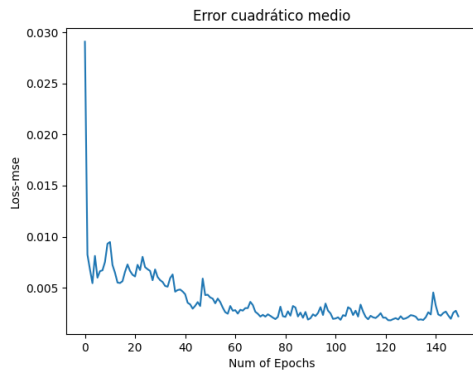


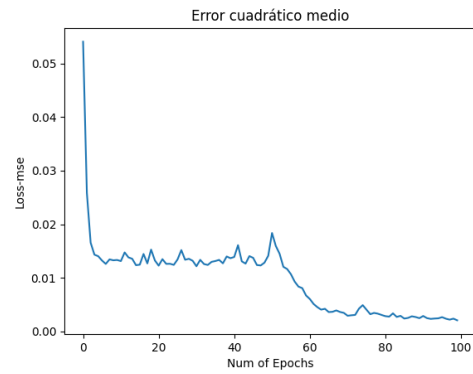
Figura 20: Estructura de una Red Neuronal Recurrente.

Fuente: [\[del Brío and Cinca, 1995\]](#).

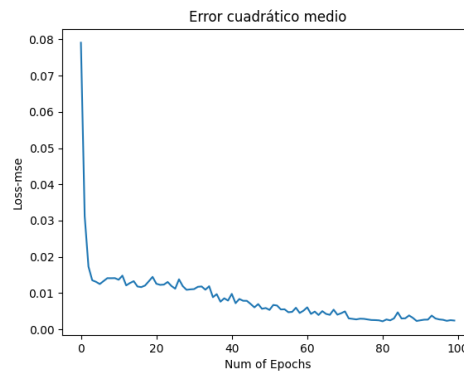
Luego de determinar los parámetros adecuados para el pronóstico del primer producto, se realiza el mismo procedimiento para los tres productos restantes. En la figura 21, se hace referencia a las curvas de error obtenidas luego de ajustar los parámetros para obtener el pronóstico deseado de los diferentes productos.



(a) MSE para el producto BH7010



(b) MSE para el producto C585



(c) MSE para el producto 1112.6

Figura 21: Respuesta del error para los productos BH7010, C585, 1112.6

Fuente: Elaborado por la autora.

La tabla 8, muestra los resultados para diferentes parámetros probados para el entrenamiento del pronóstico del segundo producto BH7010.

Tabla 8: Error de entrenamiento y de testeo para el producto BH7010

Capas	Neuronas	epoch	Error entrena	Error test
3	50,40,20	50	7.20 %	15.19 %
3	50,40,20	80	7.26 %	11.08 %
3	80,60,50	100	4.72 %	10.14 %
3	100,90,40	100	4.54 %	7.04 %
3	100,90,40	150	4.11 %	7.48 %
3	120,90,60	150	4.13 %	9.18 %
3	140,110,90	150	2.26 %	3.64 %
3	150,110,90	150	4.23 %	6.31 %

Tabla 9: Error de entrenamiento y de testeo para el producto C585

Capas	Neuronas	epoch	Error entrena	Error test
3	50,40,20	50	11.97 %	12.28 %
3	50,40,20	80	13.62 %	13.67 %
3	80,60,50	100	12.43 %	11.47 %
3	90,80,80	140	3.87 %	3.02 %
3	100,90,40	150	7.95 %	7.99 %
3	120,90,60	150	5.23 %	8.56 %
3	140,110,90	150	3.91 %	6.71 %
3	150,110,90	150	5.36 %	8.12 %

Tabla 10: Error de entrenamiento y de testeo para el producto 1112.6

Capas	Neuronas	epoch	Error entrena	Error test
3	70,55,20	50	9.33 %	19.92 %
3	80,60,40	80	7.26 %	12.11 %
3	90,60,50	100	6.74 %	18.78 %
3	100,90,40	100	8.51 %	15.22 %
3	100,90,40	150	4.11 %	8.35 %
3	120,90,50	160	2.45 %	3.27 %
3	140,110,90	150	5.29 %	8.46 %
3	150,110,90	190	4.23 %	6.83 %

Las tablas 9 y 10, reflejan los resultados arrojados al probar la red neuronal con diferentes parámetros para los productos C585 y 1112.6 respectivamente. En color verde se indica cuales con los valores de error más bajos obtenidos, mismos que permitirán obtener un pronóstico más acertado.

4.2. Resultados

Tomando en cuenta la opinión de [Yacuzzi and Paggi, 2002], se entiende por pronóstico al conjunto de actividades a través de las cuales, a partir de datos históricos del entorno series cronológicas, experiencia cualitativa, se obtienen escenarios y proyecciones de los valores futuros de las variables bajo análisis.

Una vez seleccionado los parámetros más adecuados para cada red neuronal, se obtiene un porcentaje de error para el pronóstico de la red neuronal, como se detalla en la tabla 11.

Tabla 11: Error para el pronóstico por cada producto

PRODUCTO	ERROR
BH1060	2,60 %
BH7010	3,64 %
C585	3,02 %
1112.6	3,27 %

Luego de realizar la etapa de testeo y obtener el error correspondiente para cada producto, se obtiene el pronóstico para tres años los mismos que se muestran en la tabla 12. En donde cada fila corresponde al año, mes y día del pronóstico y cada columna corresponde al producto en cuestión. La figura 22 muestra el comportamiento de los datos para el pronóstico a tres años del producto BH1060, la curva en rojo representa los datos originales, mientras la curva en azul representa la tendencia del pronóstico obtenido con el modelo.

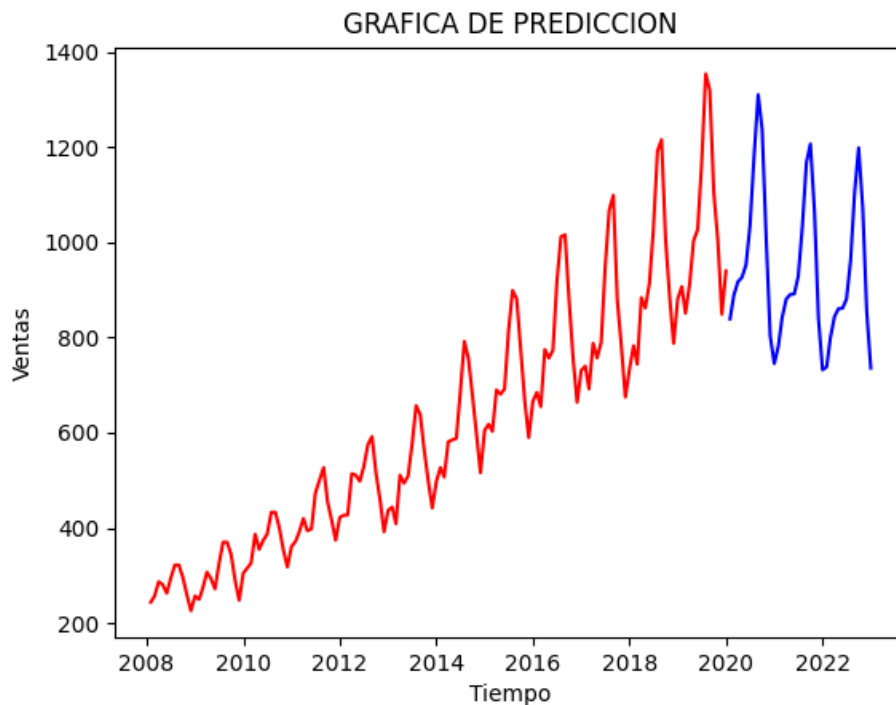


Figura 22: Curva de pronóstico para tres años del producto BH1060 .

Fuente: Elaborado por la autora.

Tabla 12: Pronóstico para tres años de los productos BH1060,BH7010, C55, 1112.6

Fecha	BH1060	BH7010	C585	1112.6
2020-01-31	839	441	220	85
2020-02-29	889	465	223	88
2020-03-31	917	482	226	92
2020-04-30	926	496	231	100
2020-05-31	953	527	248	113
2020-06-30	1,034	600	282	128
2020-07-31	1,181	715	323	136
2020-08-31	1,310	771	343	128
2020-09-30	1,236	658	319	111
2020-10-31	1,005	490	2675	99
2020-11-30	803	426	222	94
2020-12-31	745	447	205	94
2021-01-31	782	480	206	95
2021-02-28	841	504	208	99
2021-03-31	880	520	208	105
2021-04-30	890	533	209	117
2021-05-31	891	557	216	134
2021-06-30	928	628	239	150
2021-07-31	1,032	775	280	154
2021-08-31	1,167	909	320	139
2021-09-30	1,206	826	323	121
2021-10-31	1,065	556	279	110
2021-11-30	840	406	221	106
2021-12-31	732	436	193	107
2022-01-31	738	483	194	110
2022-02-28	799	516	203	116
2022-03-31	843	535	208	127
2022-04-30	860	545	210	145
2022-05-31	861	555	214	168
2022-06-30	880	592	230	182

Fecha	BH1060	BH7010	C585	1112.6
2022-07-31	961	720	265	173
2022-08-31	1,105	953	313	150
2022-09-30	1,198	1,024	338	131
2022-10-31	1,072	715	306	124
2022-11-30	855	396	242	122
2022-12-31	735	393	199	124
Total	33.999	21.075	11.341	4.377

La venta para el año 2020 en condiciones normales se estima en 11.838 docenas para el producto BH1060, el producto BH1070 con 6.518 docenas y los productos C585 y 1112.6 con 5.517 y 1.268 docenas respectivamente. De la misma forma la venta en docenas para el año 2021 en condiciones normales estaría estimada en 11.254 docenas para el producto BH1060, el producto BH1070 con 7.130 docenas y los productos C585 y 1112.6 con 2.902 y 1.547 docenas respectivamente.

Para el año 2022 se pronostica una venta de 10.907 docenas aproximadamente del producto BH1060, 7.427 docenas del producto BH1070 y para los productos C585 y 1112.6 una venta en docenas de 2.922 y 1.672 respectivamente.

Los datos obtenidos para los pronósticos en tres años arrojan un resultado total que se detalla a continuación: Para el producto BH1060 se tendrá una venta de 33.999 docenas, el producto BH1070 se pronostica una venta de 21.075 docenas mientras que para los productos C585 y 1112.6 las ventas serían de 11.341 y 4.377 respectivamente.

5. CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- La mayoría de empresas utilizan métodos estadísticos para el pronóstico de ventas, en este trabajo se presentó una opción diferente, proponiendo un modelo basado en redes neuronales artificiales, que por sus características de adaptabilidad y capacidad autoajustable en la actualidad son una herramienta de gran ayuda en empresas desarrolladas como una técnica de predicción para datos complejos, siendo una opción innovadora para el cumplimiento de sus objetivos.
- Los datos tomados para este trabajo fueron las ventas totales de cada mes ordenadas por tipo de producto. La empresa hasta el año 2019 ofertaba 127 productos para los cuales se realizó un análisis ABC para identificar el producto de mayor consumo, estos productos se dividieron en tres grupos siendo A,B y C según el porcentaje de demanda, de los cuales se tomaron dos productos del grupo A, un producto del grupo B y uno del C para realizar el pronóstico. Luego se definió el grupo de datos que serviría para entrenamiento y para testeo, tomando el 75 % de datos para entrenar el modelo neuronal y el 25 % restante se destinó para testear la red neuronal artificial.
- En la actualidad el software libre tiene características similares a las de un software comercial, por lo que la finalidad fue desarrollar el pronóstico en un software libre como es Python mediante el desarrollador PyCharm. A través del software libre, se utilizó librerías específicas de machine learning como son keras y tensorflow que permitieron el desarrollo óptimo del modelo. Para determinar la estructura adecuada de la red neuronal se realizaron pruebas consecutivas buscando obtener un mínimo error en los pronósticos, en el primer producto BH1060 perteneciente al grupo A, se obtuvo una red neuronal con una estructura de 24 entradas, 3 capas ocultas y una salida, el número de neuronas por capas fueron 100, 90 y 40 respectivamente y un epoch de 200. Para el producto BH7010 perteneciente al grupo A se obtuvo una estructura en la red neuronal que consta de 24 entradas, 3 capas ocultas, número de neuronas 140, 110 y 90 por cada capa respectivamente y epoch 150. El producto C585 correspondiente al grupo B, definió una estructura de 3 capas con 90, 80 y 80 neuronas respectivamente por cada capa con un epoch de 140, y, finalmente para el

producto 1112.6 del grupo C, la estructura fue de 3 capas ocultas con 120, 90 y 50 neuronas en cada capa y epoch 160.

- Una vez determinada la estructura de la red neuronal con hiperparámetros adecuados para cada producto, se realizó un testeo que permitió determinar el porcentaje de error entre los datos reales y los que arrojan los modelos desarrollados por la red neuronal obteniendo los siguientes resultados: Para el producto BH1060 el porcentaje de error en la prueba de testeo fue del 2.6 % , el producto BH7010 obtuvo un porcentaje de error del 3.64 %, mientras que el producto C585 obtuvo como error el 3.02 % y finalmente el producto 1112.6 generó un error del 3.27 %, en base a estos resultados se puede realizar el pronóstico para los 3 años posteriores.
- Luego de realizar el pronóstico de ventas para los productos BH1060, BH1070, C585 y 1112.6, se obtuvo el total de ventas aproximados por docenas de 33.999 correspondiente al producto BH1060, para el producto BH1070 se pronostica una venta de 21.075 docenas, mientras que para los productos C585 y 1112.6 las ventas aproximadas serían de 11.341 y 4.377 respectivamente.

5.2. RECOMENDACIONES

- Para obtener un pronóstico fiable, y que no exista sobreajuste en el entrenamiento se debe tener un histórico de datos cuyos registros sean de al menos 10 años, esto ayuda a que los resultados sean más confiables. Los datos que ingresan como entradas en la red neuronal deben estar normalizados.
- Se sugiere realizar el pronóstico de ventas usando otro tipo de redes neuronales artificiales y con métodos estadísticos, que permitan analizar la eficiencia de cada método realizando una comparativa entre cada método utilizado.
- Se recomienda tener un amplio histórico de ventas para un adecuado pronóstico, cuyos datos deben estar filtrados de forma adecuada.

Glosario

Epoch. Este es el número de veces que se ejecutaran los algoritmos. En cada ciclo (epoch) todos los datos de entrenamiento pasan por la red neuronal para que esta aprenda sobre ellos, si existen 10 ciclos y 1000 datos, cada ciclo los 1000 datos pasarán por la red neuronal [Rodríguez, 2018].

Función de activación. Las funciones de activación se encuentran en cada neurona de una red neuronal y la utilidad más importante que tienen es indicar cuando una neurona se activa o apaga [García, 2020].

Hiperparámetros. Son los parámetros que modificamos manualmente en una red neuronal. Son muy importantes para lograr un buen desempeño [Rodríguez, 2018].

Red neuronal LSTM. las redes LSTM fueron desarrolladas a partir de redes neuronales recurrentes (RNN) y adicional a su diversidad de aplicaciones, evita los problemas de dependencia a largo plazo por su estructura de unidad de almacenamiento única y ayudan a predecir series de tiempo financieras [Hochreiter and Schmidhuber, 1997].

6. BIBLIOGRAFÍA

- [Adrian, 2002] Adrian, E. (2002). Pronosticar la demanda ejercicio de adivinación matemática o fundamento de la planificación operativa. *Universidad de la República*.
- [Agustín Granell, 2018] Agustín Granell, P. (2018). Redes neuronales recurrentes: Una aplicación para los mercados bursátiles. B.S. thesis, Universitat Politècnica de Catalunya.
- [Alcocer et al., 2018] Alcocer, U. M. R., Tello-Leal, E., and Alvarado, A. B. R. (2018). Modelo basado en redes neuronales recurrentes lstm para la predicción de la siguiente actividad en procesos de negocio (lstm recurrent neural network based-model for the prediction of the next activity in business processes). *Pistas Educativas*, 40(130).
- [Bagnato, 2018] Bagnato, J. (2018). Sets de entrenamiento, test y validación.
- [Cabezón, 2018] Cabezón, M. (2018). Predicción demanda eléctrica española. implementación de redes neuronales recurrentes en python.
- [Caparrini, 2015] Caparrini, F. (2015). Redes neuronales: una visión superficial. *Dpto. deficiencias de la computación e inteligencia artificial-Universidad de sevilla*.
- [Charytoniuk et al., 2000] Charytoniuk, W., Box, E. D., Lee, W.-J., Chen, M.-S., Kotas, P., and Van Olinda, P. (2000). Neural-network-based demand forecasting in a deregulated environment. *IEEE Transactions on industry applications*, 36(3):893–898.
- [Cruz et al., 2007] Cruz, I. B., Martínez, S. S., Abed, A. R., Ábalo, R. G., and Lorenzo, M. M. G. (2007). Redes neuronales recurrentes para el análisis de secuencias. *Revista Cubana de Ciencias Informáticas*, 1(4):48–57.
- [de la Maza et al., 2018] de la Maza, L. E. L., de Cárdenas, L. Z., Pérez-Ones, O., and Llanes-Santiago, O. (2018). Obtención de un modelo neuronal para la estimación de la concentración de etanol en la destilería hécctor molina. *Tecnología Química*, 38(2):315–325.
- [del Brío and Cinca, 1995] del Brío, B. M. and Cinca, C. S. (1995). Fundamentos de redes neuronales artificiales: hardware y software. *Scire: representación y organización del conocimiento (ISSNe 2340-7042; ISSN 1135-3716)*, pages 103–125.

- [García, 2020] García, J. D. V. (2020). Redes neuronales desde cero (ii): algo de matemáticas.
- [Gil and Rodríguez, 2010] Gil, E. and Rodríguez, E. (2010). *Sistema de pronóstico de la demanda de productos farmacéuticos basado en redes neuronales*. PhD thesis, Tesis para optar el título profesional de Ingeniero de Sistemas
- [Greff et al., 2016] Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., and Schmidhuber, J. (2016). Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10):2222–2232.
- [Grosse, 2017] Grosse, R. (2017). Lecture 15: Exploding and vanishing gradients. *University of Toronto Computer Science*.
- [Gupta and Kumar, 2013] Gupta, S. and Kumar, N. (2013). A study of advancement in sales forecasting models. *Pranjana: The Journal of Management Awareness*, 16(1):1–10.
- [Guridi Mateos et al., 2017] Guridi Mateos, G. et al. (2017). Modelos de redes neuronales recurrentes en clasificación de patentes. B.S. thesis.
- [Hanke and Wichern, 2006] Hanke, J. E. and Wichern, D. W. (2006). *Pronósticos en los negocios*. Pearson educación.
- [Hilera González et al., 2000] Hilera González, J. R., Martínez Hernando, V. J., et al. (2000). *Redes neuronales artificiales: fundamentos, modelos y aplicaciones*.
- [Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.
- [Jozefowicz et al., 2015] Jozefowicz, R., Zaremba, W., and Sutskever, I. (2015). An empirical exploration of recurrent network architectures. In *International conference on machine learning*, pages 2342–2350. PMLR.
- [Llano et al., 2007] Llano, L., Hoyos, A., Arias, F., and Velásquez, J. (2007). Comparación del desempeño de funciones de activación en redes feedforward para aproximar funciones de datos con y sin ruido. *Avances en Sistemas e Informática*, 4(2).
- [López et al., 2008] López, R. F., Fernandez, J. M., and Fernández, J. M. F. (2008). *Las redes neuronales artificiales*. Netbiblo.

- [Malik, 2019] Malik, U. (2019). Solving sequence problems with lstm in keras.
- [Marín Marchant, 2020] Marín Marchant, M. A. (2020). Redes neuronales convolucionales bayesianas para diagnósticos de falla en activos físicos bajo incertidumbre: caso de análisis en bombas centrifugas.
- [Martínez, 2020] Martínez, J. (2020). Librerías de python para machine learning. Obtenido de <https://www.iartificial.net/librerias-de-python-para-machine-learning/.html>. [Web; accedido el 28-01-2020].
- [Matich, 2001] Matich, D. J. (2001). Redes neuronales: Conceptos básicos y aplicaciones. *Universidad Tecnológica Nacional, México*, 41.
- [Mercado, 1998] Mercado, S. H. (1998). *Mercadotecnia programada: principios y aplicaciones para orientar la empresa hacia el mercado*. Limusa.
- [Montesdeoca Santana, 2016] Montesdeoca Santana, B. (2016). Estudios de predicción en series temporales de datos meteorológicos utilizando redes neuronales recurrentes. B.S. thesis.
- [Morales Castro et al., 2019] Morales Castro, A., Ramírez Reyes, E., and Rodríguez Albor, G. (2019). Pronóstico de ventas de las empresas del sector alimentos: una aplicación de redes neuronales. *Semestre Económico*, 22(52):161–177.
- [Moreno, 2019] Moreno, P. (2019). Solving sequence problems with lstm in keras.
- [Nacelle and Mizraji, 2009] Nacelle, A. and Mizraji, E. (2009). Redes neuronales artificiales. *de Las redes neuronales de la biología en algoritmos de clasificación, Uruguay*.
- [Nebauer, 1998] Nebauer, C. (1998). Evaluation of convolutional neural networks for visual recognition. *IEEE Transactions on Neural Networks*, 9(4):685–696.
- [Nojek et al., 2003] Nojek, S., Britos, P., Rossi, B., and García Martínez, R. (2003). Pronóstico de ventas: Comparación de predicción basada en redes neuronales versus método estadístico. *Reportes Técnicos en Ingeniería del Software*, 5(1):1–12.
- [Olah, 2015] Olah, C. (2015). Understanding lstm networks.
- [Parisi F et al., 2003] Parisi F, A., Parisi F, F., and Guerrero C, J. L. (2003). Modelos predictivos de redes neuronales en índices bursátiles. *El Trimestre Económico*, pages 721–744.

- [Pavón Valencia, 2019] Pavón Valencia, J. F. (2019). Modelo para el pronóstico de la demanda de agua potable de emapa-i aplicando redes neuronales artificiales. B.S. thesis.
- [Peiró, 2017] Peiró, R. (2017). Análisis abc. Obtenido de <https://economipedia.com/definiciones/analisis-abc.html>. [Web; accedido el 28-12-2020].
- [Peralta, 2018] Peralta, J. M. A. (2018). Modelo de elaboración de pronóstico de ventas mediante el uso de redes neuronales artificiales y svr.
- [Pérez-Ortiz, 2002] Pérez-Ortiz, J. A. (2002). Modelos predictivos basados en redes neuronales recurrentes de tiempo discreto.
- [Quintero, 2007] Quintero, B. G. C. (2007). Predicción de proteínas con redes neuronales: Redes feedforward vs. redes recurrentes. *Facultad de Ingeniería*, 16(23):75–88.
- [Rodríguez, 2018] Rodríguez, V. (2018). Conceptos básicos sobre redes neuronales.
- [Román Morales, 2018] Román Morales, J. (2018). Redes recurrentes para el análisis de series temporales. B.S. thesis.
- [Salas, 2004] Salas, R. (2004). Redes neuronales artificiales. *Universidad de Valparaíso. Departamento de Computación*, 1.
- [Salazar Aguilar and Cabrera Ríos, 2007] Salazar Aguilar, M. A. and Cabrera Ríos, M. (2007). Pronóstico de demanda por medio de redes neuronales artificiales. *Ingenierías*, 10(35):6–12.
- [Santoyo and González, 2014] Santoyo, E. A. R. and González, J. A. L. (2014). Comparación de predicción basada en redes neuronales contra métodos estadísticos en pronósticos de ventas. *Ingeniería Industrial. Actualidad y Nuevas Tendencias*, 4(12):91–105.
- [Saucedo de Miguel, 2020] Saucedo de Miguel, P. (2020). Funciones de activación ruidosas en redes neuronales recurrentes. B.S. thesis.
- [Serna, 2003] Serna, P. (2003). Proyección de la demanda: Antecedentes, necesidad e importancia. *Universidad Nacional de Colombia, Manizales*.
- [Serrano et al., 2009] Serrano, A., Soria, E., and Martín, J. (2009). Redes neuronales artificiales. *Escuela Técnica Superior de Ingeniería, Curso de Doctorado*, pages 15–19.

- [Simeone, 2018] Simeone, O. (2018). A very brief introduction to machine learning with applications to communication systems. *IEEE Transactions on Cognitive Communications and Networking*, 4(4):648–664.
- [Simon et al., 2009] Simon, H. et al. (2009). Neural networks and learning machines. *Upper Saddle River: Pearson Education*, 3.
- [Singhal and Swarup, 2011] Singhal, D. and Swarup, K. (2011). Electricity price forecasting using artificial neural networks. *International Journal of Electrical Power & Energy Systems*, 33(3):550–555.
- [Srivastava et al., 2014] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Sakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.
- [Torres, 1994] Torres, L. G. (1994). Redes neuronales y aproximación de funciones. *Boletín de Matemáticas*, 1(2):35–58.
- [Toscano Hernandez et al., 2019] Toscano Hernandez, M. R., CUEVA, N. M. M., and ORTEGA, J. F. C. (2019). Cadena de suministros. In *Colloquium*.
- [Urbina et al., 2016] Urbina, L., Moreno-Armendáriz, M. A., Duchanoy, C. A., and Calvo, H. (2016). Pronóstico del área de contacto de los neumáticos de un vehículo via redes neuronales recurrentes.
- [Villada et al., 2008] Villada, F., Cadavid, D. R., and Molina, J. D. (2008). Pronóstico del precio de la energía eléctrica usando redes neuronales artificiales. *Revista Facultad de Ingeniería Universidad de Antioquia*, (44):111–118.
- [Villanueva Espinoza, 2002] Villanueva Espinoza, M. d. R. (2002). Las redes neuronales artificiales y su importancia como herramienta en la toma de decisiones. In [Villanueva Espinoza, 2002].
- [Yacuzzi and Paggi, 2002] Yacuzzi, E. and Paggi, G. (2002). Diseño e implementación de un sistema de pronóstico de ventas en whirlpool argentina. Technical report, Serie Documentos de Trabajo.

7. ANEXOS

7.1. Código de entrenamiento de la Red Neuronal Artificial

Producto BH1060

```
#####  
#   LIBRERIAS  
#####  
import pandas_datareader as pdr  
import pandas as pd  
import matplotlib.pyplot as plt  
import numpy as np  
from sklearn.preprocessing import MinMaxScaler  
import numpy  
### LIBRERIAS PARA CREAR EL MODELO LSTM  
from tensorflow.keras.models import Sequential  
from tensorflow.keras.layers import Dense  
from tensorflow.keras.layers import LSTM  
import tensorflow as tf  
import math  
from sklearn.metrics import mean_squared_error  
from numpy import array  
import matplotlib.pyplot as plt  
import os.path  
  
# LEER EL ARCHIVO DE DATOS  
df=pd.read_csv('Productos1.csv')  
df.head()  
# LEE LOS DATOS DE UNA COLUMNA ESPECÍFICA  
df1 = df.reset_index()['TIPO A']  
plt.plot(df1)  
# NORMALIZA LOS DATOS PARA SU TRATAMIENTO  
scaler = MinMaxScaler(feature_range=(0,1))  
df1=scaler.fit_transform(np.array(df1).reshape(-1,1))
```

```

##DIVIDE LOS DATOS PARA ENTRENAMIENTO Y TESTEO
training_size = int(len(df1)*0.75)
test_size = len(df1)-training_size
train_data,test_data = df1[0:training_size:],df1[training_size:
len(df1),:1]

# MATRIZ DE DATOS
def create_dataset(dataset, time_step=24):
dataX, dataY = [], []
for i in range(len(dataset)-time_step-1):
a = dataset[i:(i+time_step), 0]
dataX.append(a)
dataY.append(dataset[i + time_step, 0])
return numpy.array(dataX), numpy.array(dataY)
time_step = 24
X_train, y_train = create_dataset(train_data, time_step)
X_test, y_test = create_dataset(test_data, time_step)

#REMODELA LA ENTRADA PARA QUE SEA [muestras, pasos de tiempo]
CARACTERISTICA DE UNA RED LSTM
X_train_1 =X_train.reshape(X_train.shape[0],X_train.shape[1] , 1)
X_test_1 = X_test.reshape(X_test.shape[0],X_test.shape[1] , 1)

# CREACION DE LA ESTRUCTURA DE LA RED
model=Sequential()
model.add(LSTM(100,return_sequences=True,input_shape=(24,1)))
model.add(LSTM(90,return_sequences=True))
model.add(LSTM(60))
model.add(Dense(1))
model.compile(loss=['mean_squared_error'],optimizer='adam')
print(model.summary())

```

```

history=model.fit(X_train_1,y_train,
validation_data=(X_test_1,y_test),
epochs=250,batch_size=10,verbose=1)

plt.figure()
plt.plot(history.history['loss'])
plt.xlabel("Num of Epochs")
plt.ylabel("Loss-mse")
plt.title("Error cuadrático medio")
plt.savefig('loss.png')
if os.path.isfile('modelosave_A.h5') is False:
model.save('modelosave_A.h5')

### REALIZO LAS PREDICCIONES
# Predicciones para el X_train
train_predict=model.predict(X_train_1)
# Predicciones para el x_test
test_predict=model.predict(X_test_1)

### Calcula RMSE
trainScore=math.sqrt(mean_squared_error(y_train,train_predict))
print('Resultado del entrenamiento: %.2f RMSE' % (trainScore*100))
### Test Data RMSE
testScore=math.sqrt(mean_squared_error(y_test,test_predict))
print('Resultado del test: %.2f RMSE' % (testScore*100))

##Traansformo a los datos originales
train_predict=scaler.inverse_transform(train_predict)
test_predict=scaler.inverse_transform(test_predict)

look_back=24
trainPredictPlot = numpy.empty_like(df1)
trainPredictPlot[:, :] = np.nan

```

```

trainPredictPlot[look_back:len(train_predict)+look_back, :] =
    train_predict
testPredictPlot = numpy.empty_like(df1)
testPredictPlot[:, :] = numpy.nan
testPredictPlot[len(train_predict)+(look_back*2)+1:
len(df1)-1, :] = test_predict

plt.figure(0)
plt.plot(scaler.inverse_transform(df1))
plt.plot(trainPredictPlot)
plt.plot(testPredictPlot)
plt.plot(trainPredictPlot,'orange', linewidth = 2)
plt.plot(testPredictPlot,'g', linewidth = 2)
plt.legend( ('Datos', 'Prediccion datos entramiento',
'Prediccion sobre los data test'), loc = 'upper left')
plt.xlabel("Tiempo ")
plt.ylabel("Ventas")
plt.title("Entrenamiento")
plt.savefig('TIPO_A5.png')
plt.show()

```

7.2. Código de implementación para el pronóstico

```
#####  
#   LIBRERIAS  
#####  
import pandas_datareader as pdr  
import pandas as pd  
import numpy as np  
from sklearn.preprocessing import MinMaxScaler  
import math  
from numpy import array  
import os.path  
from tensorflow.keras.models import load_model  
import matplotlib.pyplot as plt  
import matplotlib.dates as mdates  
import tkinter as tk  
from PIL import Image, ImageTk  
import csv  
  
#GENERO LA VENTADA QUE SIRVE DE INTERFACE  
ventana_prin = tk.Tk()  
ventana_prin.title("PRONÓSTICOS IMPACTEX mao")  
ventana_prin.geometry("840x400")  
ventana_prin.configure(background='silver')  
img = Image.open('mao2.png')  
photo = ImageTk.PhotoImage(img)  
fondo=tk.Label(ventana_prin,image=photo).place (x=5,y=325)  
img2 = Image.open('uta.png')  
photo2 = ImageTk.PhotoImage(img2)  
fondo2=tk.Label(ventana_prin,image=photo2).place (x=700,y=250)  
  
#INDICADOR DEL NUMERO DE PREDCCIONES  
Num_predic_A = tk.Label(ventana_prin,text=  
"PRONÓSTICO \n TIPO A1",bg="silver",fg="black")
```



```

Num_predic_A.place(x=18, y=10)

Num_predic_B = tk.Label(ventana_prin,text=
"PRONÓSTICO \n TIPO A2",bg="silver",fg="black")
Num_predic_B.place(x=228, y=10)

Num_predic_C = tk.Label(ventana_prin,text=
"PRONÓSTICO \n TIPO B",bg="silver",fg="black")
Num_predic_C.place(x=438, y=10)

Num_predic_D = tk.Label(ventana_prin,text=
"PRONÓSTICO \n TIPO C",bg="silver",fg="black")
Num_predic_D.place(x=648, y=10)

#numero_predict_1 = tk.StringVar()
entrada1 = tk.Entry(ventana_prin)
#entrada1.pack(side=tk.TOP)
entrada1.place(x=10, y=45)

entrada2 = tk.Entry(ventana_prin)
entrada2.place(x=220, y=45)

entrada3 = tk.Entry(ventana_prin)
entrada3.place(x=430, y=45)

entrada4 = tk.Entry(ventana_prin)
entrada4.place(x=640, y=45)

# REALIZA LA PREDCCION TIPO A
def Prediccion_tipo_A():
# NUMERO DE AÑOS PREDICCHOS
Num_años = int(entrada1.get())

```

```

# CARGO EL MODELO
new_model = load_model('modelosave_A.h5')

# LEO LA BASE DE DATOS
df = pd.read_csv('Productos.csv')
df.head()
df2 = df.reset_index()['TIPO A']

# DATOS PARA LA GRAFICA DE LA BASE DE DATOS
x1 = np.array(df2)
times1 = pd.date_range(start=df.Date[0],
periods=len(df), freq='M')

# ESCALIZAR LA LA BASE DE DATOS
scaler = MinMaxScaler(feature_range=(0, 1))
df1 = scaler.fit_transform(np.array(df2).reshape(-1, 1))
# SELECCIONO 24 DATOS ANTERIORES (2 AÑOS)
    PARA HACER LAS PREDICCIONES
x_input = df1[len(df1) - 24:len(df1)]
lst_output = np.array([])
array1 = np.array([x_input])
i = 0
n_steps = 24

# CICLO WHILE PARA HACER LAS PREDICCIONES (3 AÑOS)
while (i < Num_años * 12):
array1 = array1.reshape((1, n_steps, 1))
yhat = new_model.predict(array1, verbose=1)
lst_output = np.append(lst_output, yhat)
array_aum = np.append(array1, yhat)
array_dis = np.delete(array_aum, 0)
array1 = np.array([array_dis])
i = i + 1

```

```

a = [lst_output]
predic_esc = scaler.inverse_transform(a)

# CREO UN PERIODO AUXILIAR
time_pre_inicial = pd.date_range(start=df.
Date[len(df) - 1], periods=2, freq='M')

# DATOS DE LA PREDICCION
x2 = predic_esc[0]
# times2 = pd.date_range(start='01/1961',
end='01/1964', freq='M')
times2 = pd.date_range(start=time_pre_inicial[1],
periods=12 * Num_años, freq='M')

##GRAFICA DE LOS DATOS
plt.figure()
plt.plot(times1, x1, 'r', times2, x2, 'b')
# xfmt = mdates.DateFormatter('%m-%y')
plt.title("GRAFICA DE PREDICCION")
plt.xlabel("Tiempo")
plt.ylabel("Ventas")
plt.savefig('prediccion_A.png')

name_dict = {
'Fecha': times2,
'Prediccion': x2
}
datos_predic = pd.DataFrame(name_dict)

datos_predic.to_csv('prediccion_A.csv') # relative position
# print(datos_predic)

# REALIZA PA PREDDCION TIPO B
def Prediccion_tipo_B():
# NUMERO DE AÑOS PREDICCHOS

```

```

#     Num_años = 2
Num_años_B = int(entrada2.get())

# CARGO EL MODELO
new_model_B = load_model('modelosave_B.h5')
# new_model.summary()

# LEO LA BASE DE DATOS
df_B = pd.read_csv('Productos.csv')
df_B.head()
df2_B = df_B.reset_index()['TIPO B']

# DATOS PARA LA GRAFICA DE LA BASE DE DATOS
x1_B = np.array(df2_B)
# times1 = pd.date_range(start='01/1949',
end='01/1961', freq='M')
times1_B = pd.date_range(start=df_B.Date[0],
    periods=len(df_B), freq='M')

# ESCALIZAR LA LA BASE DE DATOS
scaler_B = MinMaxScaler(feature_range=(0, 1))
df1_B = scaler_B.fit_transform(np.array(df2_B).reshape(-1, 1))

# len(test_data)
# SELECCIONO 24 DATOS ANTERIORES (2 AÑOS)
    PARA HACER LAS PREDICCIONES
x_input_B = df1_B[len(df1_B) - 24:len(df1_B)]
lst_output_B = np.array([])
array1_B = np.array([x_input_B])
i = 0
n_steps_B = 24

# CICLO WHILE PARA HACER LAS PREDICCIONES (3 AÑOS)
while (i < Num_años_B * 12):
array1_B = array1_B.reshape((1, n_steps_B, 1))

```

```

yhat_B = new_model_B.predict(array1_B, verbose=1)
lst_output_B = np.append(lst_output_B, yhat_B)
array_aum_B = np.append(array1_B, yhat_B)
array_dis_B = np.delete(array_aum_B, 0)
array1_B = np.array([array_dis_B])
i = i + 1
a_B = [lst_output_B]
predic_esc_B = scaler_B.inverse_transform(a_B)

# CREO UN PERIODO AUXILIAR
time_pre_inicial_B = pd.date_range(start=df_B.
Date[len(df_B) - 1], periods=2, freq='M')

# DATOS DE LA PREDICCION
x2_B = predic_esc_B[0]
# times2 = pd.date_range(start='01/1961',
end='01/1964', freq='M')
times2_B = pd.date_range(start=time_pre_inicial_B[1],
periods=12 * Num_años_B, freq='M')

##GRAFICA DE LOS DATOS
plt.figure()
plt.plot(times1_B, x1_B, 'r', times2_B, x2_B, 'b')
# xfmt = mdates.DateFormatter('%m-%y')
plt.title("GRAFICA DE PREDICCION")
plt.xlabel("Tiempo")
plt.ylabel("Ventas")
plt.savefig('prediccion_B.png')

name_dict_B = {
'Fecha': times2_B,
'Prediccion': x2_B
}
datos_predic_B = pd.DataFrame(name_dict_B)

```

```

datos_predic_B.to_csv('prediccion_B.csv')

# REALIZA PA PREDDCION TIPO C
def Prediccion_tipo_C():
# NUMERO DE AÑOS PREDICCHOS
#     Num_años = 2
Num_años_C = int(entrada3.get())

# CARGO EL MODELO
new_model_C = load_model('modelosave_C.h5')
# new_model.summary()

# LEO LA BASE DE DATOS
df_C = pd.read_csv('Productos.csv')
df_C.head()
df2_C = df_C.reset_index()['TIPO C']

# DATOS PARA LA GRAFICA DE LA BASE DE DATOS
x1_C = np.array(df2_C)
# times1 = pd.date_range(start='01/1949',
#     end='01/1961', freq='M')
times1_C = pd.date_range(start=df_C.Date[0],
# periods=len(df_C), freq='M')

# ESCALIZAR LA LA BASE DE DATOS
scaler_C = MinMaxScaler(feature_range=(0, 1))
df1_C = scaler_C.fit_transform(np.array(df2_C).
# reshape(-1, 1))

# len(test_data)
# SELECCIONO 24 DATOS ANTERIORES (2 AÑOS)
# PARA HACER LAS PREDICCIONES
x_input_C = df1_C[len(df1_C) - 24:len(df1_C)]
lst_output_C = np.array([])
array1_C = np.array([x_input_C])

```

```

i = 0
n_steps_C = 24

# CICLO WHILE PARA HACER LAS PREDICCIONES (3 AÑOS)
while (i < Num_años_C * 12):
array1_C = array1_C.reshape((1, n_steps_C, 1))
yhat_C = new_model_C.predict(array1_C, verbose=1)
lst_output_C = np.append(lst_output_C, yhat_C)
array_aum_C = np.append(array1_C, yhat_C)
array_dis_C = np.delete(array_aum_C, 0)
array1_C = np.array([array_dis_C])
i = i + 1

a_C = [lst_output_C]
predic_esc_C = scaler_C.inverse_transform(a_C)

# CREO UN PERIODO AUXILIAR
time_pre_inicial_C = pd.date_range(start=df_C.
Date[len(df_C) - 1], periods=2, freq='M')

# DATOS DE LA PREDICCION
x2_C = predic_esc_C[0]
# times2 = pd.date_range(start='01/1961',
end='01/1964', freq='M')
times2_C = pd.date_range(start=time_pre_inicial_C[1],
periods=12 * Num_años_C, freq='M')

##GRAFICA DE LOS DATOS
plt.figure()
plt.plot(times1_C, x1_C, 'r', times2_C, x2_C, 'b')
# xfmt = mdates.DateFormatter('%m-%y')
plt.title("GRAFICA DE PREDICCION")
plt.xlabel("Tiempo")
plt.ylabel("Ventas")
plt.savefig('prediccion_C.png')

```

```

name_dict_C = {
'Fecha': times2_C,
'Prediccion': x2_C
}
datos_predic_C = pd.DataFrame(name_dict_C)
datos_predic_C.to_csv('prediccion_C.csv')
# REALIZA LA PREDDCION TIPO C
def Prediccion_tipo_D():
# NUMERO DE AÑOS PREDICCHOS
#     Num_años = 2
Num_años_D = int(entrada4.get())
# CARGO EL MODELO
new_model_D = load_model('modelosave_D.h5')
# new_model.summary()
# LEO LA BASE DE DATOS
df_D = pd.read_csv('Productos.csv')
df_D.head()
df2_D = df_D.reset_index()['TIPO D']
# DATOS PARA LA GRAFICA DE LA BASE DE DATOS
x1_D = np.array(df2_D)
# times1 = pd.date_range(start='01/1949',
end='01/1961', freq='M')
times1_D = pd.date_range(start=df_D.Date[0],
    periods=len(df_D), freq='M')
# ESCALIZAR LA LA BASE DE DATOS
scaler_D = MinMaxScaler(feature_range=(0, 1))
df1_D = scaler_D.fit_transform(np.array(df2_D).
reshape(-1, 1))
# len(test_data)
# SELECCIONO 24 DATOS ANTERIORES (2 AÑOS)
PARA HACER LAS PREDICCIONES
x_input_D = df1_D[len(df1_D) - 24:len(df1_D)]
lst_output_D = np.array([])
array1_D = np.array([x_input_D])
i = 0

```



```

n_steps_D = 24
# CICLO WHILE PARA HACER LAS PREDICCIONES (3 AÑOS)
while (i < Num_años_D * 12):
array1_D = array1_D.reshape((1, n_steps_D, 1))
yhat_D = new_model_D.predict(array1_D, verbose=1)
lst_output_D = np.append(lst_output_D, yhat_D)
array_aum_D = np.append(array1_D, yhat_D)
array_dis_D = np.delete(array_aum_D, 0)
array1_D = np.array([array_dis_D])
i = i + 1
a_D = [lst_output_D]
predic_esc_D = scaler_D.inverse_transform(a_D)
# CREO UN PERIODO AUXILIAR
time_pre_inicial_D = pd.date_range(start=df_D.
Date[len(df_D) - 1], periods=2, freq='M')

# DATOS DE LA PREDICCION
x2_D = predic_esc_D[0]
# times2 = pd.date_range(start='01/1961',
end='01/1964', freq='M')
times2_D = pd.date_range(start=time_pre_inicial_D[1],
periods=12 * Num_años_D, freq='M')
##GRAFICA DE LOS DATOS
plt.figure()
plt.plot(times1_D, x1_D, 'r', times2_D, x2_D, 'b')
# xfmt = mdates.DateFormatter('%m-%y')
plt.title("GRAFICA DE PREDICCION")
plt.xlabel("Tiempo")
plt.ylabel("Ventas")
plt.savefig('prediccion_D.png')

name_dict_D = {
'Fecha': times2_D,
'Prediccion': x2_D
}

```

```

datos_predic_D = pd.DataFrame(name_dict_D)

datos_predic_D.to_csv('prediccion_D.csv')
#DESTRUYE LA PRIMERA GRAFICA
def cerrar_prin():
ventana_prin.destroy()
# FUNCION APARECER LA GRAFICA A
def cargar_foto_A():
ventanal = tk.Tk()
ventanal.geometry("900x495")
ventanal.configure(bg="dark turquoise")
ventanal.title("GRAFICA DE PREDICCION")
imagen1 = tk.PhotoImage(file="prediccion_A.png",
    master=ventanal)
fondol = tk.Label(ventanal, image=imagen1).
place(x=250, y=5)
#    fondol.pack()

# SUBIR EL ARCHIVO CSV
File = open("prediccion_A.csv")
Reader = csv.reader(File)
Data = list(Reader)
#    print(Data[1])
list_of_entries = []
for x in list(range(0, len(Data))):
list_of_entries.append(Data[x][1:3])
print(list_of_entries)
listbox1 = tk.Listbox(ventanal, height = 27, width=30)
for x,y in enumerate(list_of_entries):
listbox1.insert(x,y)
listbox1.grid(row=0, column=0)

# BOTON CERRAR PRIMERA PREDICCION
#    botoncierra_ven_A = tk.Button(ventanal,
    text="CERRAR", fg="blue", command=ventanal.destroy)

```

```

# botoncierra_ven_A.pack(side=tk.BOTTOM)
fondo1.pack()

# FUNCION APARECER LA GRAFICA B
def cargar_foto_B():
ventana2 = tk.Tk()
ventana2.geometry("900x495")
ventana2.configure(bg="dark turquoise")
ventana2.title("GRAFICA DE PREDICCION")

imagen1_B = tk.PhotoImage(file="prediccion_B.png",
    master=ventana2)
fondo1_B = tk.Label(ventana2, image=imagen1_B).
place(x=250, y=5)
# fondo1.pack()

# SUBIR EL ARCHIVO CSV
File_B = open("prediccion_B.csv")
Reader_B = csv.reader(File_B)
Data_B = list(Reader_B)
# print(Data[1])
list_of_entries_B = []
for x in list(range(0, len(Data_B))):
list_of_entries_B.append(Data_B[x][1:3])
print(list_of_entries_B)
listbox1_B = tk.Listbox(ventana2, height = 27,
width=30)
for x,y in enumerate(list_of_entries_B):
listbox1_B.insert(x,y)

listbox1_B.grid(row=0, column=0)

# BOTON CERRAR PRIMERA PREDICCION
# botoncierra_ven_A = tk.Button(ventana1,
text="CERRAR", fg="blue", command=ventana1.destroy)

```

```

# botoncierra_ven_A.pack(side=tk.BOTTOM)
fondo1_B.pack()

# FUNCION APARECER LA GRAFICA C
def cargar_foto_C():
ventana3 = tk.Tk()
ventana3.geometry("900x495")
ventana3.configure(bg="dark turquoise")
ventana3.title("GRAFICA DE PREDICCION")

imagen1_C = tk.PhotoImage(file="prediccion_C.png",
master=ventana3)
fondo1_C = tk.Label(ventana3, image=imagen1_C).
place(x=250, y=5)
# fondo1.pack()

# SUBIR EL ARCHIVO CSV
File_C = open("prediccion_C.csv")
Reader_C = csv.reader(File_C)
Data_C = list(Reader_C)
# print(Data[1])
list_of_entries_C = []
for x in list(range(0, len(Data_C))):
list_of_entries_C.append(Data_C[x][1:3])
print(list_of_entries_C)

listbox1_C = tk.Listbox(ventana3, height = 27, width=30)
for x,y in enumerate(list_of_entries_C):
listbox1_C.insert(x,y)

listbox1_C.grid(row=0, column=0)

# BOTON CERRAR PRIMERA PREDICCION
# botoncierra_ven_A = tk.Button(ventana1,
text="CERRAR", fg="blue", command=ventana1.destroy)

```

```

# botoncierra_ven_A.pack(side=tk.BOTTOM)
fondo1_C.pack()
# FUNCION APARECER LA GRAFICA D
def cargar_foto_D():
ventana4 = tk.Tk()
ventana4.geometry("900x495")
ventana4.configure(bg="dark turquoise")
ventana4.title("GRAFICA DE PREDICCION")

imagen1_D = tk.PhotoImage(file="prediccion_D.png",
master=ventana4)
fondo1_D = tk.Label(ventana4, image=imagen1_D).
place(x=250, y=5)
# fondo1.pack()

# SUBIR EL ARCHIVO CSV
File_D = open("prediccion_D.csv")
Reader_D = csv.reader(File_D)
Data_D = list(Reader_D)
# print(Data[1])
list_of_entries_D = []
for x in list(range(0,len(Data_D))):
list_of_entries_D.append(Data_D[x][1:3])
print(list_of_entries_D)

listbox1_D = tk.Listbox(ventana4,height = 27,
width=30)
for x,y in enumerate(list_of_entries_D):
listbox1_D.insert(x,y)

listbox1_D.grid(row=0,column=0)

# BOTON CERRAR PRIMERA PREDICCION
# botoncierra_ven_A = tk.Button(ventana1,
text="CERRAR", fg="blue", command=ventana1.destroy)

```

```

# botoncierra_ven_A.pack(side=tk.BOTTOM)
fondo1_D.pack()

#BOTON SELECCION PREDICCION A
boton_predic_A = tk.Button(ventana_prin,text=
"PREDICCIÓN",fg="blue",command=Prediccion_tipo_A)
boton_predic_A.place(x=35, y=80)

#BOTON SELECCION GRAFICA A
boton_graf_A = tk.Button(ventana_prin,text=
"GRÁFICA",fg="blue",command=cargar_foto_A)
boton_graf_A.place(x=45, y=120)

#BOTON SELECCION PREDICCION B
boton_predic_B = tk.Button(ventana_prin,text="PREDICCIÓN",
fg="blue",command=Prediccion_tipo_B)
boton_predic_B.place(x=245, y=80)

#BOTON SELECCION GRAFICA B
boton_graf_B = tk.Button(ventana_prin,text="GRÁFICA",
fg="blue",command=cargar_foto_B)
boton_graf_B.place(x=255, y=120)

#BOTON SELECCION PREDICCION C
boton_predic_C = tk.Button(ventana_prin,text=
"PREDICCIÓN",fg="blue",command=Prediccion_tipo_C)
boton_predic_C.place(x=455, y=80)

#BOTON SELECCION GRAFICA C
boton_graf_C = tk.Button(ventana_prin,text="GRÁFICA",
fg="blue",command=cargar_foto_C)
boton_graf_C.place(x=465, y=120)

#BOTON SELECCION PREDICCION D
boton_predic_D = tk.Button(ventana_prin,text=

```

```
"PREDICCIÓN", fg="blue", command=Prediccion_tipo_D)
boton_predic_D.place(x=665, y=80)

#BOTON SELECCION GRAFICA C
boton_graf_D = tk.Button(ventana_prin, text="GRÁFICA",
fg="blue", command=cargar_foto_D)
boton_graf_D.place(x=675, y=120)

#BOTON CERRAR PRINCIPAL
boton_cerrar= tk.Button(ventana_prin, text="CERRAR",
fg="red", command=cerrar_prin)
#boton_cerrar.pack(side=tk.TOP)
boton_cerrar.place(x=710, y=225)

ventana_prin.mainloop()
```

7.3. Interface creada para elegir el número de años del pronóstico

luego de realizar el entrenamiento de todos los productos, la red neuronal artificial puede realizar el pronóstico para un número de años indicado. Para esto se creó una interface que permite seleccionar el número de años como se muestra en la figura 23. La ventana emergente que se despliega permite introducir el número de años para los cuales se desea realizar el pronóstico así como una opción que permite ver los datos generados con su respectiva gráfica. Los productos Tipo A1 hace referencia al producto BH1060, TipoA2 al producto BH7010, TipoB al producto C585 y el TipoC al producto 1112.6.

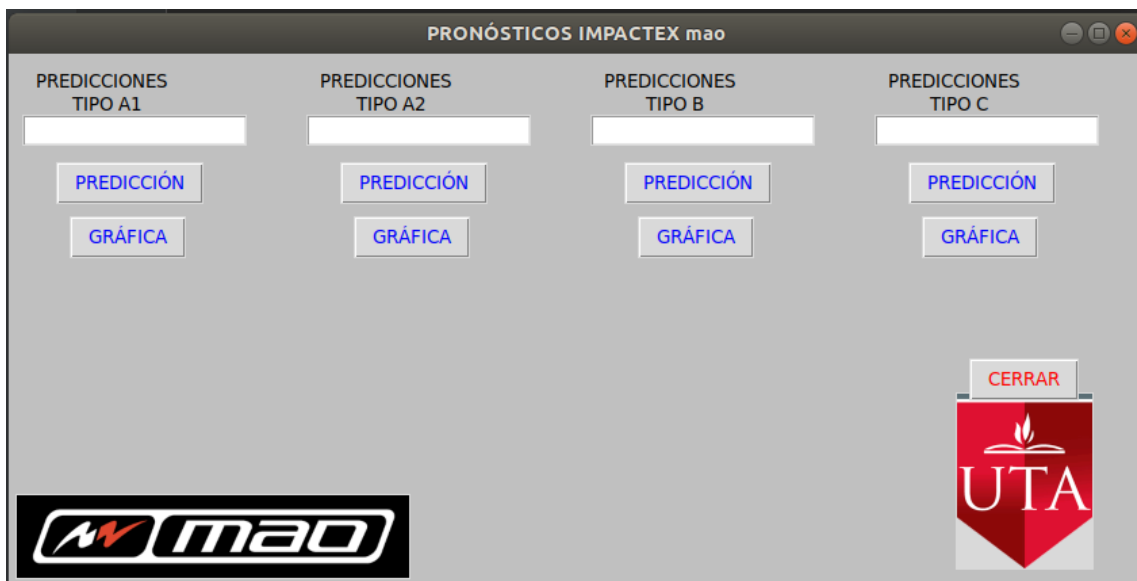


Figura 23: Interface para elegir el número de años para el pronóstico.

7.4. Tabla de pronóstico para 5 años.

Tabla 13: Pronóstico para cinco años de los productos BH1060, BH7010, C55, 1112.6

Fecha	BH1060	BH7010	C585	1112.6
2020-01-31	839.12	441.59	220.64	85.84
2020-02-29	889.11	465.13	223.44	88.01
2020-03-31	917.52	482.20	226.59	92.62
2020-04-30	926.40	496.66	231.75	100.58
2020-05-31	953.08	527.97	248.84	113.40
2020-06-30	1,034.62	600.92	282.11	128.43
2020-07-31	1,181.72	715.04	323.46	136.22
2020-08-31	1,310.39	771.43	343.45	128.28
2020-09-30	1,236.97	658.62	319.30	111.83
2020-10-31	1,005.41	490.28	267.95	99.93
2020-11-30	803.19	426.19	222.81	94.89
2020-12-31	745.33	447.78	205.99	94.12
2021-01-31	782.53	480.30	206.01	95.59
2021-02-28	841.77	504.78	208.02	99.16
2021-03-31	880.95	520.90	208.93	105.97
2021-04-30	890.35	533.20	209.69	117.61
2021-05-31	891.74	557.22	216.61	134.42
2021-06-30	928.45	628.64	239.57	150.99
2021-07-31	1,032.98	775.91	280.76	154.33
2021-08-31	1,167.77	909.27	320.67	139.79
2021-09-30	1,206.80	826.33	323.40	121.23
2021-10-31	1,065.38	556.29	279.34	110.43
2021-11-30	840.02	406.24	221.04	106.76
2021-12-31	732.20	436.88	193.59	107.23
2022-01-31	738.24	483.53	194.98	110.19
2022-02-28	799.14	516.47	203.77	116.34

Fecha	BH1060	BH7010	C585	1112.6
2022-03-31	843.83	535.64	208.68	127.46
2022-04-30	860.65	545.46	210.49	145.57
2022-05-31	861.55	555.37	214.94	168.12
2022-06-30	880.64	592.95	230.73	182.01
2022-07-31	961.42	720.29	265.85	173.43
2022-08-31	1,105.95	953.35	313.75	150.23
2022-09-30	1,198.64	1,024.85	338.61	131.97
2022-10-31	1,072.91	715.16	306.81	124.04
2022-11-30	855.23	396.08	242.95	122.48
2022-12-31	735.76	393.55	199.17	124.56
2023-01-31	727.47	459.89	193.59	130.13
2023-02-28	772.22	505.65	203.18	141.09
2023-03-31	816.19	533.77	210.57	160.46
2023-04-30	838.24	548.38	213.52	187.96
2023-05-31	842.16	554.59	216.57	210.72
2023-06-30	852.90	565.60	228.27	207.93
2023-07-31	910.12	628.36	258.89	181.21
2023-08-31	1,032.92	839.76	309.50	155.18
2023-09-30	1,156.42	1,109.87	351.12	141.95
2023-10-31	1,111.14	969.89	336.02	137.63
2023-11-30	898.32	526.83	270.31	138.13
2023-12-31	746.16	336.11	210.67	142.67
2024-01-31	711.91	415.41	214.94	153.14
2024-02-29	747.40	478.68	230.73	173.30
2024-03-31	793.58	519.46	265.85	204.54
2024-04-30	821.97	543.09	313.75	234.00
2024-05-31	829.96	554.09	338.61	235.47
2024-06-30	836.18	558.15	306.81	205.89

Fecha	BH1060	BH7010	C585	1112.6
2024-07-31	876.99	576.07	242.95	173.35
2024-08-31	980.69	685.28	199.17	155.28
2024-09-30	1,108.86	995.14	193.59	148.25
2024-10-31	1,113.96	1,159.81	216.57	147.11
2024-11-30	940.90	804.71	228.27	150.50
2024-12-31	774.57	381.36	351.12	160.27

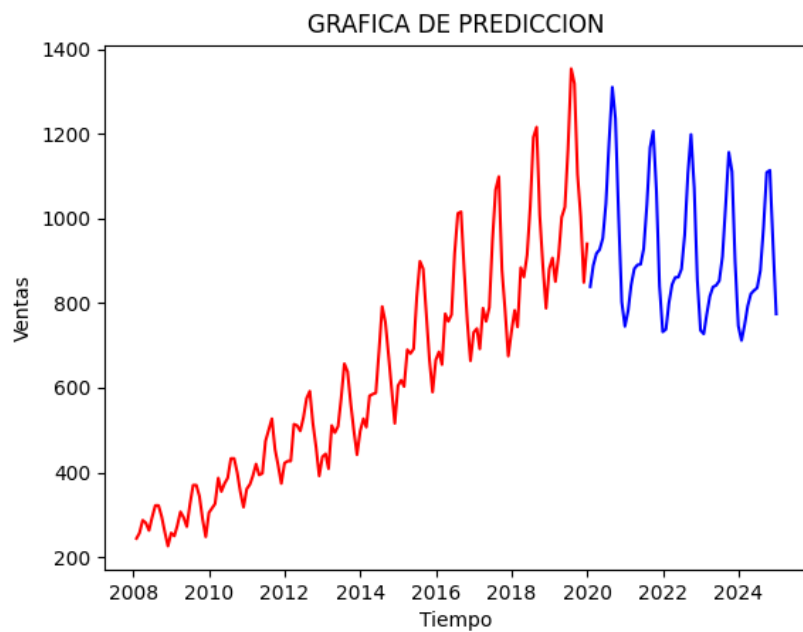


Figura 24: Pronóstico para 5 años del producto BH1060

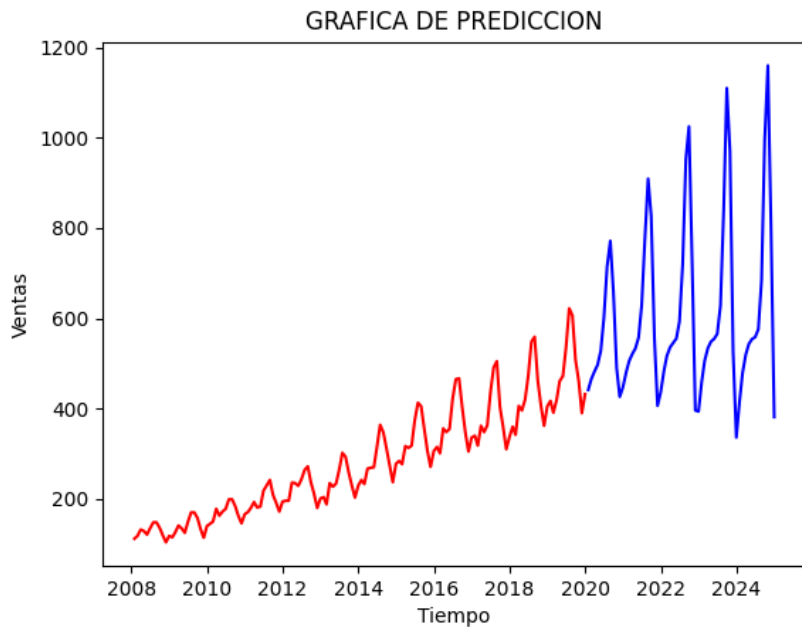


Figura 25: Pronóstico para 5 años del producto BH7010

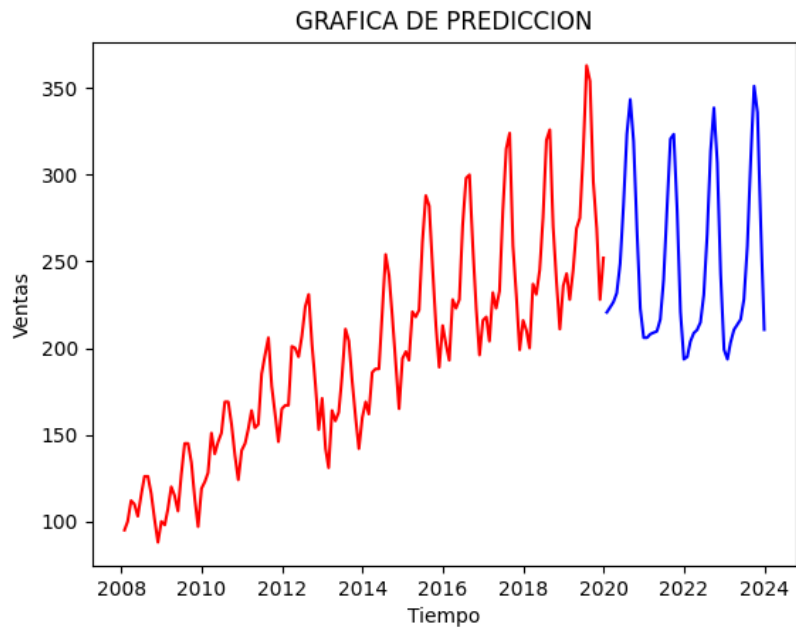


Figura 26: Pronóstico para 5 años del producto C585