

NOMBRE DEL TRABAJO

Sistema inteligente para mejorar la predicción de tendencia en la empresa textil R &D Corporación S.A

AUTOR

Russbell Ronny Castillo Sanca

RECUENTO DE PALABRAS

19435 Words

RECUENTO DE CARACTERES

107128 Characters

RECUENTO DE PÁGINAS

107 Pages

TAMAÑO DEL ARCHIVO

2.9MB

FECHA DE ENTREGA

Apr 18, 2024 3:27 PM GMT-5

FECHA DEL INFORME

Apr 18, 2024 3:29 PM GMT-5

● 16% de similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para cada base de datos.

- 16% Base de datos de Internet
- Base de datos de Crossref
- 0% Base de datos de trabajos entregados
- 4% Base de datos de publicaciones
- Base de datos de contenido publicado de Crossref

● Excluir del Reporte de Similitud

- Material bibliográfico
- Material citado
- Material citado
- Coincidencia baja (menos de 8 palabras)



**FORMULARIO DE AUTORIZACIÓN PARA LA
PUBLICACIÓN DE TRABAJOS DE INVESTIGACIÓN EN
EL REPOSITORIO INSTITUCIONAL DE LA UNTELS
(Art. 45° de la ley N° 30220 – Ley)**

Autorización de la propiedad intelectual del autor para la publicación de tesis en el Repositorio Institucional de la Universidad Nacional Tecnológica de Lima Sur (<https://repositorio.unfels.edu.pe>), de conformidad con el Decreto Legislativo N° 822, sobre la Ley de los Derechos de Autor, Ley N° 30035 del Repositorio Nacional Digital de Ciencia, Tecnología e Innovación de Acceso Abierto, Art. 10° del Rgto. Nacional de Trabajos de Investigación para optar grados académicos y títulos profesionales en las universidades – RENATI Res. N° 084-2022-SUNEDU/CD, publicado en El Peruano el 16 de agosto de 2022; y la RCO N° 061-2023-UNTELS del 01 marzo 2023.

TIPO DE TRABAJO DE INVESTIGACIÓN

- 1). TESIS () 2). TRABAJO DE SUFICIENCIA PROFESIONAL (X)

DATOS PERSONALES

Apellidos y Nombres:	Castillo Sanca Russbell Ronny
D.N.I.:	72327314
Otro Documento:	
Nacionalidad:	Peruano
Teléfono:	917625997
e-mail:	rcastillosanca@gmail.com

DATOS ACADÉMICOS

Pregrado

Facultad:	Facultad de Ingeniería y Gestión
Programa Académico:	Trabajo de suficiencia profesional
Título Profesional otorgado:	Ingeniero de Sistemas

Postgrado

Universidad de Procedencia:	
País:	
Grado Académico otorgado:	

Datos de trabajo de investigación

Título:	Sistema inteligente para mejorar la predicción de tendencias en la empresa textil R&D CORPORACION SAC
Fecha de Sustentación:	16 de diciembre del 2023
Calificación:	Aprobado por Unanimidad
Año de Publicación:	2024

AUTORIZACIÓN DE PUBLICACIÓN EN VERSIÓN ELECTRÓNICA

A través de la presente, autorizo la publicación del texto completo de la tesis, en el Repositorio Institucional de la UNTELS especificando los siguientes términos:

Marcar con una X su elección.

- 1) Usted otorga una licencia especial para publicación de obras en el REPOSITORIO INSTITUCIONAL DE LA UNIVERSIDAD NACIONAL TECNOLÓGICA DE LIMA SUR.

Si autorizo No autorizo

- 2) Usted autoriza para que la obra sea puesta a disposición del público conservando los derechos de autor y para ello se elige el siguiente tipo de acceso.

Derechos de autor		
TIPO DE ACCESO	ATRIBUCIONES DE ACCESO	ELECCIÓN
ACCESO ABIERTO 12.1(*)	info:eu-repo/semantics/openAccess (Para documentos en acceso abierto)	(X)

- 3) Si usted dispone de una **PATENTE** puede elegir el tipo de **ACCESO RESTRINGIDO** como derecho de autor y en el marco de confiabilidad dispuesto por los numerales 5.2 y 6.7 de la directiva N° 004-2016-CONCYTEC DEGC que regula el Repositorio Nacional Digital de CONCYTEC (Se colgará únicamente datos del autor y el resumen del trabajo de investigación).

Derechos de autor		
TIPO DE ACCESO	ATRIBUCIONES DE ACCESO	ELECCIÓN
ACCESO RESTRINGIDO	info:eu-repo/semantics/restrictedAccess (Para documentos restringidos)	()
	info:eu-repo/semantics/embargoedAccess (Para documentos con períodos de embargo. Se debe especificar las fechas de embargo)	()
	info:eu-repo/semantics/closedAccess (para documentos confidenciales)	()

(*) <http://renati.sunedu.gob.pe>



Rellene la siguiente información si su trabajo de investigación es de acceso restringido:

Atribuciones de acceso restringido:

Motivos de la elección del acceso restringido:

Castillo Sanca Russbell Ronny

APELLIDOS Y NOMBRES

72327314

DNI



Firma y huella:

Lima, 17 de Junio del 20 24

UNIVERSIDAD NACIONAL TECNOLÓGICA DE LIMA SUR

**FACULTAD DE INGENIERÍA Y GESTIÓN
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**



**“SISTEMA INTELIGENTE PARA MEJORAR LA PREDICCIÓN DE
TENDENCIAS EN LA EMPRESA TEXTIL R&D CORPORACIÓN S.A.C.”**

TRABAJO DE SUFICIENCIA PROFESIONAL

Para optar el Título Profesional de

INGENIERO DE SISTEMAS

PRESENTADO POR EL BACHILLER

CASTILLO SANCA, RUSSBELL RONNY

ORCID: 0009-0001-7536-7881

ASESOR

HUAROTE ZEGARRA, RAÚL EDUARDO

ORCID: 0000-0001-7466-7404

Villa El Salvador

2023



VI Programa de Titulación por la Modalidad de Trabajo de Suficiencia Profesional
Decanato de la Facultad de Ingeniería y Gestión

ACTA DE SUSTENTACIÓN DE TRABAJO DE SUFICIENCIA PROFESIONAL PARA OBTENER EL
TÍTULO PROFESIONAL DE INGENIERO DE SISTEMAS

En Villa El Salvador, siendo las 10:40:00 am, del día sábado 16 de diciembre de 2023, se reunieron en la Sala de Sustentación, los miembros del Jurado Evaluador del Trabajo de Suficiencia Profesional integrado por:

Presidente	: Dr. Alfredo Cesar Larios Franco	CIP. N° 78376
Secretario	: Dr. Julio Elvis Valero Cajahuanca	CIP. N° 87161
Vocal	: Mg. Manuel Abelardo Alcantara Ramirez	CIP. N° 149488

Designados con Resolución de Decanato de la Facultad de Ingeniería y Gestión N° 984-2023-UNTELS-R-D, de fecha 13 de diciembre de 2023.

Se da inició al acto público de sustentación y evaluación del Trabajo de Suficiencia Profesional, para obtener el Título Profesional de Ingeniero de Sistemas, bajo la modalidad de Titulación por Trabajo de Suficiencia Profesional (Resolución de Consejo Universitario N° 065-2023-UNTELS-CU de fecha 08 de agosto del 2023), en la cual se APRUEBA el "Reglamento, Directiva, Cronograma y Presupuesto del VI Programa de Titulación por la Modalidad de Trabajo de Suficiencia Profesional de la Universidad Nacional Tecnológica de Lima Sur"; siendo que el Art. 4° del precitado Reglamento establece que: "La Modalidad de Titulación prevista consiste en la presentación, aprobación y sustentación de un Trabajo de Suficiencia Profesional que dé cuenta de la experiencia profesional y además permita demostrar el logro de las competencias adquiridas en el desarrollo de los estudios de pregrado que califican para el ejercicio de la profesión correspondiente. Quienes participen en esta modalidad no podrán tramitar simultáneamente otras modalidades de titulación. Además, los participantes inscritos en esta modalidad, deberán acreditar un mínimo de dos (02) años de experiencia laboral, de acuerdo a lo establecido en la Resolución N° 174-2019- SUNEDU/CD y al anexo 1 sobre Glosario de Términos en el punto veinte (20)...", en el cual;

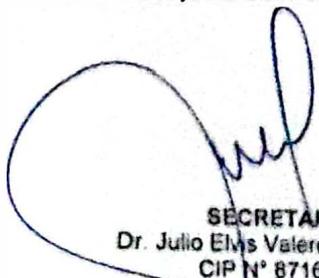
El Bachiller: **RUSBELL RONNY CASTILLO SANCA**

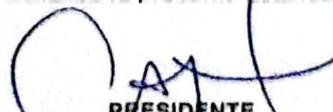
Sustentó su Trabajo de Suficiencia Profesional: "**SISTEMA INTELIGENTE PARA MEJORAR LA PREDICCIÓN DE TENDENCIAS EN LA EMPRESA TEXTIL R&D CORPORACIÓN S.A.C.**".

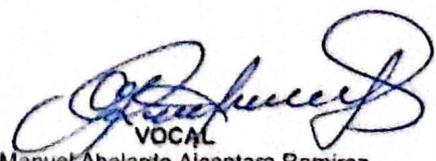
Concluida la Sustentación del Trabajo de Suficiencia Profesional, se procedió a la calificación correspondiente según el siguiente detalle:

Condición Aprobado por unanimidad Equivalencia Bueno de acuerdo al Art. 65° del Reglamento General para el Otorgamiento de Grado Académico y Título Profesional de la UNTELS vigente.

Siendo las 11:20 am del día 16 de diciembre de 2023, se dio por concluido el acto de sustentación del Trabajo de Suficiencia Profesional, firmando la presente acta los miembros del Jurado.


SECRETARIO
Dr. Julio Elvis Valero Cajahuanca
CIP N° 87161


PRESIDENTE
Dr. Alfredo Cesar Larios Franco
CIP. N° 78376


VOCAL
Mg. Manuel Abelardo Alcantara Ramirez
CIP N° 149488

Nota: Art. 14° - La sustentación del Trabajo de Suficiencia Profesional se realizará en un acto público. En caso de faltar algún miembro del Jurado, la sustentación procederá con los dos integrantes presentes. En caso de ausencia del Presidente del Jurado, asumirá la presidencia el representante de mayor categoría y antigüedad. En caso de ausencia de dos o más miembros del Jurado, la sustentación será reprogramada durante los 05 días siguientes.

DEDICATORIA

Trabajo de investigación dedicado primeramente a mis Padres, por todo el apoyo, esfuerzo y aliento que hicieron por mí durante todos estos años de vida. A mis hermanos, quienes son uno de los motivos de superarme cada día más.

AGRADECIMIENTO

Expresar mi profundo agradecimiento a las mis familiares y amigos de haber contribuido de manera significativa a la realización de esta tesis.

Expreso mi sincera gratitud, a todos los profesores que me acompañaron en toda mi etapa universitaria, por su orientación, apoyo, conocimientos y experiencia, fueron fundamentales en cada proceso de este proyecto.

Agradezco a mi padre Edilberto castillo por su apoyo inquebrantable, comprensión y paciencia a lo largo de este viaje. Sus palabras de aliento y motivación constante que me sirvió como motivación para lograr mi objetivo, así como a todos aquellos que, de manera, compartieron su conocimiento y recursos. A todos ustedes, les estoy profundamente agradecido.

ÍNDICE

ÍNDICE.....	IV
RESUMEN	XI
INTRODUCCIÓN.....	XII
CAPÍTULO I.	13
I. ASPECTOS GENERALES	13
1.1 Contexto	13
1.1.1. <i>Misión.</i>	14
1.1.2. <i>Visión.</i>	14
1.1.3. <i>Valores.</i>	15
1.2. Justificación	15
1.3. Delimitación temporal y espacial del trabajo.....	16
1.3.1 <i>Temporal</i>	16
1.3.2 <i>Espacial</i>	16
1.4. Objetivos	16
1.3.1 Objetivo General	16
1.3.2 Objetivo Específico	17
CAPÍTULO II.	18
II. MARCO TEÓRICO.....	18
2.1. Antecedentes de la Investigación	18
2.1.1. <i>Antecedentes Internacionales.</i>	18
2.1.2. <i>Antecedentes Nacionales.</i>	20
2.2. Bases teóricas:.....	22
2.2.1 <i>Sistema inteligente</i>	22
2.2.1.1. Características del pronóstico	24
2.2.1.2 Métodos de pronósticos cualitativos	25
2.2.1.2. Métodos de pronósticos cuantitativos	26
2.2.2. Redes Neuronales Artificiales.....	27
2.2.2.1. Estructura básica de una Red Neuronal Artificial.....	29
2.2.2.2 Tipos de Redes Neuronales Artificiales	30
2.2.2.3 Función de activación de una Red Neuronal Artificial	31
2.2.2.4 Mecanismos de aprendizaje.....	34
2.2.3 Aplicaciones de las Redes Neuronales.	35
2.2.4 Metodología	36

2.2.4.1	Metodología de Kaastra y Boyd	36
2.2.4.1.6	39
2.3	Definición de términos básicos:	40
2.3.1	Aprendizaje	40
2.3.2	Backpropagation	41
2.3.3	Predicción.....	41
2.3.4	Redes Neuronales de tipo biológico.....	41
2.3.5	Redes Neuronales Artificiales (ANN).....	42
CAPÍTULO III	44
III.	DESARROLLO DEL TRABAJO PROFESIONAL	44
3.1	Determinación y Análisis del Problema.....	44
3.2	Aplicación de modelo de solución.....	44
3.2.1	Proceso 1: Selección de las variables	44
3.2.2	Proceso 2: Recolección de los datos	45
3.2.3	Proceso 3: Preparación de los datos	45
3.2.4	Proceso 4: Conjunto de entrenamiento, prueba y validación	45
3.2.5	Proceso 5: Aplicación de los paradigmas de las redes neuronales.	45
3.2.6	Proceso 6: Criterio de evaluación.....	47
3.2.7	Proceso 7: Entrenamiento de la RNA	47
3.2.8	Proceso 4: Implementación.....	47
3.2.8.1	Fase de entrenamiento (Blusas).....	54
3.2.8.2	Fase de entrenamiento (Cartera).....	60
3.2.8.3	Fase de entrenamiento (Casaca).....	65
3.2.8.4	Fase de entrenamiento (Chompas).....	71
3.2.8.6	Fase de entrenamiento (pantalón)	76
3.2.8.7	Fase de entrenamiento (Sacos).....	82
3.2.8.9	Fase de entrenamiento (short)	87
3.2.8.11	Fase de entrenamiento (vestidos)	93
3.3	Resultados	100
CONCLUSIONES	102
RECOMENDACIONES	103
REFERENCIAS BIBLIOGRÁFICAS	104
ANEXO	106

Listado de figuras

Figura 1- Organigrama de la empresa.	14
Figura 2- Ubicación de la empresa R&D Corporación.....	16
Figura 3- Elementos presentes al pronosticar	25
Figura 4- Comparación de una Neurona Biológica y una Artificial	29
Figura 5- Red Neuronal Artificial	29
Figura 6- Estructura del Perceptrón Multicapa	31
Figura 7- Esquema de la función lineal	32
Figura 8- Esquema de la función sigmoidea	33
Figura 9- Esquema de la Función tangente hiperbólica	33
Figura 10- Tipos de normalizaciones	37
Figura 11- Función de errores	39
Figura 12 - Esquema de una neurona.....	42
Figura 13- Secuencia de tiempo.....	46
Figura 14 – Colab Google, Python, Librería, Keras, Panda, Numpy	47
Figura 15 - Importación de librería Pandas	49
Figura 16- Importación y carga de datos a analizar	49
Figura 17- Llenado de valores con la media.....	50
Figura 18- Normalizar los datos	50
Figura 19- Identificar los input y target de la Red Neuronal	51
Figura 20- Normalizar la salida.....	52
Figura 21- Definición de la estructura de la Red Neuronal para su aprendizaje	52
Figura 22- Creación de las capas ocultas de la Red Neuronal Artificial	53
Figura 23- Estructura de la Red Neuronal Artificial	53
Figura 24- Configuración de los parámetros de la red	53
Figura 25- Realizar el proceso de entrenamiento de la Red Neuronal.....	54
Figura 26- Resultado de Entrenamiento.....	54
Figura 27- Grafico de Perdida Vs Épocas	55
Figura 28- Grafico de Entrenamiento de la Red Neuronal (blusas).....	57
Figura 29- Resultados con los Datos de Validación	58
Figura 30- Grafico de los Datos del Entrenamiento.....	59
de la Red Neuronal (blusas).....	59

Figura 31- Pronostico	59
Figura 32- Realizar el proceso de entrenamiento de la Red Neuronal.....	60
Figura 33- Resultado de Entrenamiento.....	60
Figura 34- Grafico de Perdida Vs Épocas	61
Figura 35- Grafico de Entrenamiento de la Red Neuronal (Cartera)	63
Figura 36- Resultados con los Datos de Validación	63
Figura 37- Grafico de los Datos del Entrenamiento.....	64
de la Red Neuronal (cartera).....	64
Figura 38- Pronostico	65
Figura 39- Realizar el proceso de entrenamiento de la	65
Red Neuronal	65
Figura 40- Resultado de Entrenamiento.....	66
Figura 41- Grafico de Perdida Vs Épocas	66
Figura 42- Grafico de Entrenamiento de la Red Neuronal (casaca).....	68
Figura 43- Resultados con los Datos de Validación	69
Figura 44- Grafico de los Datos del Entrenamiento.....	70
de la Red Neuronal (Casaca)	70
Figura 45- Pronostico	70
Figura 46- Realizar el proceso de entrenamiento de la	71
Red Neuronal	71
Figura 47- Resultado de Entrenamiento.....	71
Figura 48- Grafico de Perdida Vs Épocas	72
Figura 49- Grafico de Entrenamiento de la Red	74
Neuronal (chompas).....	74
Figura 50- Resultados con los Datos de Validación	74
Figura 51- Grafico de los Datos del Entrenamiento.....	75
de la Red Neuronal (chompas).....	75
Figura 52- Pronostico	76
Figura 53- Realizar el proceso de entrenamiento de la	76
Red Neuronal	76
Figura 54- Resultado de Entrenamiento.....	77
Figura 55- Grafico de Perdida Vs Épocas	77
Figura 56- Grafico de Entrenamiento de la Red Neuronal	79

(pantalon)	79
Figura 57- Resultados con los Datos de Validación	80
Figura 58- Grafico de los Datos del Entrenamiento.....	81
de la Red Neuronal (pantalon)	81
Figura 59- Pronostico	81
Figura 60- Realizar el proceso de entrenamiento de la.....	82
Red Neuronal	82
Figura 61- Resultado de Entrenamiento.....	82
Figura 62- Grafico de Perdida Vs Épocas	83
Figura 63- Grafico de Entrenamiento de la Red Neuronal	85
(sacos)	85
Figura 64- Resultados con los Datos de Validación	85
Figura 65- Grafico de los Datos del Entrenamiento.....	86
de la Red Neuronal (Sacos)	86
Figura 66- Pronostico	87
Figura 67- Realizar el proceso de entrenamiento de la.....	87
Red Neuronal	87
Figura 68- Resultado de Entrenamiento.....	88
Figura 69- Grafico de Perdida Vs Épocas	88
Figura 70- Grafico de Entrenamiento de la Red Neuronal	90
(short).....	90
Figura 71- Resultados con los Datos de Validación	91
Figura 72- Grafico de los Datos del Entrenamiento.....	92
de la Red Neuronal (short)	92
Figura 73- Pronostico	92
Figura 74- Realizar el proceso de entrenamiento de la.....	93
Red Neuronal	93
Figura 75- Resultado de Entrenamiento.....	93
Figura 76- Grafico de Perdida Vs Épocas	94
Figura 77- Grafico de Entrenamiento de la Red Neuronal	96
(vestidos).....	96
Figura 78- Resultados con los Datos de Validación	96
Figura 79- Grafico de los Datos del Entrenamiento.....	97

de la Red Neuronal (vestidos)	97
Figura 80- Pronostico	98
Figura 80- Histórico de ventas de los meses de agosto	99

Listado de figuras

Tabla 1. Ventas anuales por prenda R&D CORPORACION	44
Tabla 2. Entrenamiento de la Red Neuronal (blusas)	56
Tabla 3. Datos de Validación (blusas)	58
Tabla 4. Datos del Entrenamiento de la Red Neuronal (cartera)	61
Tabla 5. Datos de Validación (cartera).....	64
Tabla 6. Datos del Entrenamiento de la Red Neuronal (casaca)	67
Tabla 7. Datos de Validación (casaca)	69
Tabla 8. Datos del Entrenamiento de la Red Neuronal (chompas).....	72
Tabla 9. Datos de Validación (chompas)	75
Tabla 10. Datos del Entrenamiento de la Red Neuronal (pantalon).....	78
Tabla 11. Datos de Validación (pantalon)	80
Tabla 12. Datos del Entrenamiento de la Red Neuronal (sacos)	83
Tabla 13. Datos de Validación (sacos)	86
Tabla 14. Datos del Entrenamiento de la Red Neuronal (short)	89
Tabla 15. Datos de Validación (short).....	91
Tabla 16. Datos del Entrenamiento de la Red Neuronal (vestidos)	94
Tabla 17. Datos de Validación (vestidos).....	97
Tabla 18. Histórico de Ventas de los Meses de Agosto.....	98
Tabla 19. Resumen de la predicción de venta para el mes de agosto	99
Tabla 20. Costos para la implementación del proyecto.....	103

RESUMEN

El análisis de datos ha emergido como una herramienta fundamental para comprender patrones y tendencias cambiantes, permitiendo a la empresa anticipar y adaptarse a las demandas del mercado.

La investigación se aplicó a la empresa R&D Corporaciones, se recomendó implementar un sistema de inteligencia artificial, encargada de utilizar métodos de análisis de datos y aprendizaje automático para anticipar las direcciones futuras en la industria de la moda, mejorando de esta forma la formulación de decisiones estratégicas.

Según la entrevista que se realizó al jefe de almacén se llegó a la conclusión que hay una mala gestión de inventarios que provoca pérdidas del 20% debido a que se realizan promociones o descuentos constantemente para que no haya productos de más y así puedan lanzar nuevas colecciones por temporada. Con la implementación del sistema, se refleja que hay una disminución de pérdidas solo en 5% también se pudo observar anticipadamente de manera más precisa las preferencias del consumidor, no solo identifica tendencias actuales, sino que también pronostica posibles evoluciones futuras, permitiendo que la empresa tome decisiones informadas sobre diseño, producción y comercialización

Palabras Clave: Redes neuronales, predicción, sinapsis

INTRODUCCIÓN

La industria de la moda, ha experimentado un cambio significativo en los últimos tiempos. En este escenario de constante cambio, las empresas de moda se enfrentan al desafío de no solo seguir el ritmo de las tendencias emergentes, sino también de anticiparlas para tomar decisiones estratégicas efectivas.

La empresa R&D Corporación, se dedica a la industria textil de la producción y comercialización de prendas formales y casuales para damas, con diseños propios e innovadores. Actualmente cuentan con 7 puntos de venta repartidos en la ciudad de Lima. La empresa inició a funcionar a partir del 2002 y hasta la fecha va posicionándose entre los mejores del sector, en el mercado competitivo, aportando desde la calidad e imagen. El cliente de R&D Corporación actualmente utilizan un sistema de venta “bsale”, pero no cuentan con una herramienta que les permita recopilar, procesar y analizar datos heterogéneos para revelar tendencias cambiantes y patrones de comportamiento del consumidor.

Como solución se propuso implementar un sistema haciendo uso con la Inteligencia Artificial (IA), utilizando técnicas avanzadas como redes neuronales computacionales y algoritmos de aprendizaje automático, se pretende no solo identificar tendencias actuales, sino también prever posibles direcciones futuras con un grado significativo de precisión.

La investigación consta de 3 capítulos, en el capítulo I se define los datos de la empresa, las problemáticas y los objetivos de investigación, en el capítulo II, se redacta todo referente al marco teórico para realizar la investigación y el capítulo III, se explica el desarrollo de la ejecución del trabajo con las herramientas empleadas.

CAPÍTULO I.

I. ASPECTOS GENERALES

1.1 Contexto

R&D Corporación, es una empresa que fabrica y vende ropas de vestir (específicamente prendas para mujer), calzado, artículos, accesorios (bolsos y correas), es la entidad en la cual se brindara el servicio de desarrollo del sistema. La empresa empezó a funcionar desde el mes de mayo del 2002 y hasta la fecha va posicionándose entre los mejores del sector textil, y apuesta a continuar con su historia. Su propósito es ofrecerles a sus clientes un estilo de ropa en el cual se sientan cómodas y seguras, aportando desde la calidad e imagen, la mejora de su confianza.

Ofrecen colecciones de ropa, blusas, sacos, faltas, pantalones, polos, entre otras prendas de vestir, formal y casual, transmitiendo el concepto y el estilo de la marca de calidad garantizada, su propósito es que sus clientes, cree sus propios looks partiendo de un buen básico. Los productos son atemporales para que formen parte del día a día de la mujer de hoy. Los diseñan, producen y cuidan con especial detalle para llevarles calidad a sus clientes, garantizándoles comodidad y durabilidad.

Las prendas van dirigidas para el sector juvenil y persona adulta mayor para satisfacer las necesidades al gusto del mercado potencial, acompañando en las experiencias diarias con moda accesible.

El trabajo se desarrolló para la empresa R&D Corporación, que inicio con la marca, Camila Viali y años después apertura la marca Affari, la implementación del sistema permitió como solución reducir los excedentes de productos fuera de temporada, evitando realizar así muchas promociones de reducción de precio y ayudo a tomar decisiones informadas sobre diseño, producción y comercialización.

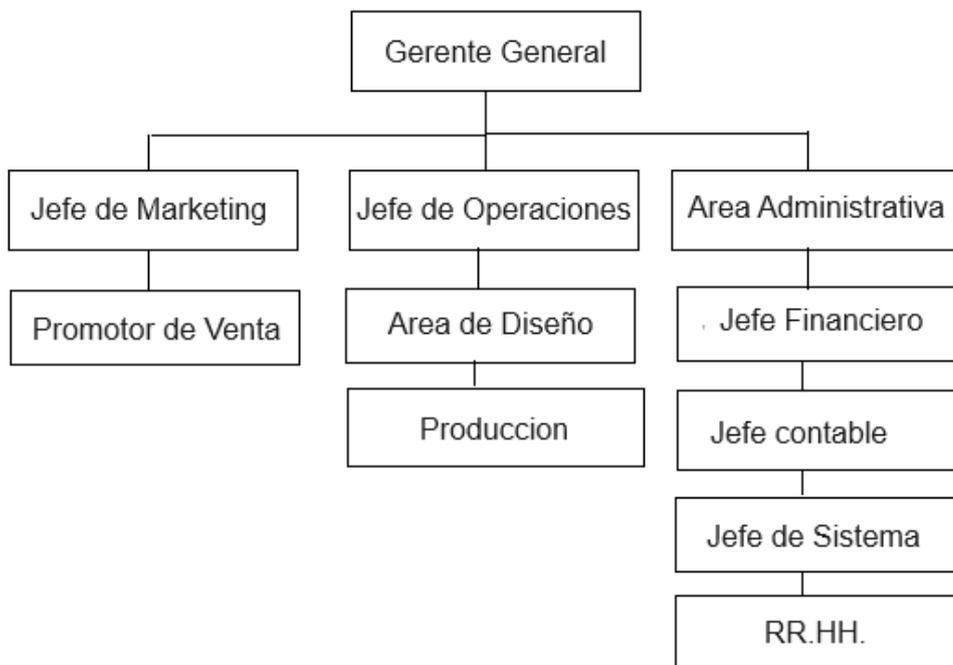
Con la implementación del sistema inteligente, se pudo observar una gran mejora en la empresa. Según el jefe de operaciones se detectó que en el inventario había tipos de prendas que no se vendían mucho y eso ocasionaba que se genere descuentos masivos provocando así perdidas en la empresa, así mismo se detectó que había prendas que sí se vendían en su totalidad, pero la demanda de los clientes no cubría con las cantidades producidas y esto como

consecuencia se originaba perdidas de varias ventas. De igual manera, se visualizó que algunos modelos de accesorios (bolsos y/o carteras) estuvieron guardados por mucho tiempo ocasionando rajaduras o deterioro que se echaron todo a perder, eso se originaba porque no se realizaba un estudio histórico de las ventas locales, sino que la producción se basaba en modelos de productos internacionales que no eran en su totalidad a gusto de los sus clientes locales. Analizando las metas de las ventas propuestas, se llegó a la conclusión que después de implementar el sistema, mejoro en 4.2%

En el siguiente organigrama mostrado en la Figura 1 se visualiza la organización de la empresa R&D Corporación.

Figura 1

Organigrama de la empresa



FUENTE: R&D Corporación, 2018

1.1.1. Misión.

Brindar a nuestros clientes un estilo de ropa con el cual se sientan cómodas y seguras, aportando desde la calidad e imagen, la mejora de su confianza

1.1.2. Visión.

Ser una empresa de moda posicionándose entre los mejores del sector en el mercado competitivo, y apostamos continuar con nuestra historia

1.1.3. Valores.

- Puntualidad
- Respeto
- Honestidad
- Confianza
- Compromiso

1.2. Justificación

La implementación de un Sistema Inteligente de Predicción de Tendencias es una necesidad que se presenta en la empresa R&D Corporación, debido a que es un mercado altamente competitivo y sujeto a cambios constantes en las preferencias de los consumidores. La incapacidad para anticipar con precisión estas tendencias tiene varias repercusiones, como la producción ineficiente, la acumulación de inventario no vendido, lo que a su vez afecta negativamente la utilidad de la empresa y su aptitud para atender las necesidades del mercado.

Este sistema inteligente de predicción de tendencias, aplicado en el método de las redes neuronales, aprovecha tecnologías avanzadas, como el aprendizaje automático, que su función es analizar grandes conjuntos de datos y descubrir patrones que serían difíciles de detectar mediante métodos tradicionales. Esto posibilitará que la compañía tome decisiones respaldadas por información precisa acerca de las tendencias de moda y las inclinaciones de los clientes.

No solo mejorará la toma de decisiones, sino que la introducción de este sistema también generará efectos positivos en la eficiencia de la cadena de suministro de la empresa. Desde la obtención de materias primas hasta la distribución de productos finales, la coordinación se volverá más eficaz, lo que conducirá a una mejor utilización de los recursos y una disminución de los gastos operativos.

No solo se potenciará la toma de decisiones, sino que la implementación de este sistema también acarreará impactos positivos en la eficiencia del inventario de la empresa. Desde la adquisición de materias primas hasta la distribución de productos finales, la coordinación se optimizará, resultando en

una utilización más efectiva de los recursos y una reducción de los costos operativos.

Esto no solo es una ventaja estratégica en términos de imagen de marca, sino que también puede generar ahorros a largo plazo.

1.3. Delimitación temporal y espacial del trabajo

1.3.1 Temporal

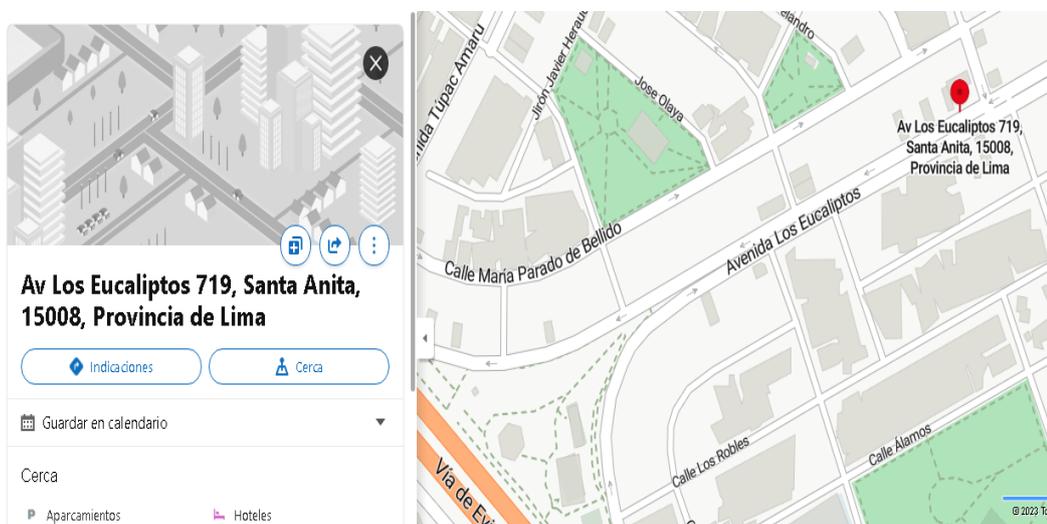
El trabajo de investigación tendrá 3 meses y medio de duración entre el periodo del 12 de abril 2023 y finalizará el 25 de julio del 2023

1.2.2 Espacial

La ejecución de la investigación para este proyecto se realizó en la sede de la empresa R&D Corporaciones, situada en la Av. Los Eucaliptos N° 719, altura de Los Ficus, en el distrito de Santa Anita, Lima.

Figura 2

Ubicación de la empresa R&D Corporación



Fuente 2: R&D Corporación 2023

1.4. Objetivos

1.3.1 Objetivo General

Implementación de un Sistema Inteligente basado en redes neuronales artificial para prever las tendencias en la industria de la moda de la empresa R&D Corporaciones.

1.3.2 Objetivo Específico

- Diseñar un modelo de redes neuronales adecuado para prever tendencias de moda basado en los datos recopilados.
- Entrenar un modelo de redes neuronales adecuado para prever tendencias de moda basado en los datos recopilados.
- Evaluar el impacto del sistema en las decisiones comerciales y la rentabilidad de R&D Corporaciones.

CAPÍTULO II.

II. MARCO TEÓRICO

2.1. Antecedentes de la Investigación

2.1.1. Antecedentes Internacionales.

- Padilla y Bonifaz (2021), autores de la tesis titulada “Desarrollo de un sistema de recomendación inteligente para selección de prendas utilizando análisis de imágenes con redes neuronales.”

Surgió la necesidad de desarrollar un sistema que optimizara la selección de productos de moda para los usuarios. Este sistema no solo mejoraría la elección de prendas de moda al proporcionar una experiencia personalizada que simplifica la selección, sino que también contribuiría a reducir los costos de producción de las temporadas.

La solución propuesta consistió en un sistema de recomendación de moda basado en las preferencias del usuario. Se llevaron a cabo dos fases para evaluar su capacidad de adaptación. En la primera fase, se creó una memoria a corto plazo que se actualiza constantemente a partir de las interacciones del usuario. La segunda fase generó una memoria a largo plazo utilizando una Red Neuronal Profunda. El sistema de recomendación se dividió en tres fases: Generación de Bases de Datos, Clasificación de Modelos y Perfilado Implícito.

Los experimentos basados en la interacción con el usuario demostraron que el sistema es capaz de adaptarse al sugerir conjuntos similares a las selecciones anteriores y ajustarse a las preferencias del usuario durante las interacciones hombre-máquina. Cumple así con los requisitos necesarios para predecir las preferencias del cliente. El desarrollo del sistema incorpora recomendaciones que aprovechan tanto memorias a corto como a largo plazo para mejorar la experiencia del usuario.

- Laura (2020), autor de la tesis titulada “Modelo predictivo de demanda basado en redes neuronales para aprovisionamiento en empresas textiles” En la actualidad, se ha

observado que las empresas emplean análisis estadísticos de ventas con el fin de anticipar la demanda futura, pero estos análisis suelen ser inexactos debido a la falta de consideración de factores externos y al uso de métodos estadísticos convencionales. Tomaban decisiones basadas en su intuición y experiencia en lugar de emplear modelos de predicción avanzados. Estas dificultades resultaban en una planificación deficiente de la adquisición de materias primas, lo que generaba costos adicionales debido a compras excesivas y almacenamiento innecesario, además de pérdida de ventas por falta de suministros disponibles. Por lo tanto, decidieron desarrollar un modelo predictivo de demanda que se basara en una red neuronal artificial para mejorar la gestión de suministros en empresas del sector textil. Su objetivo principal consistía en crear y entrenar una red neuronal artificial utilizando variables clave que influyen en la demanda en el contexto de empresas textiles, con el propósito de lograr un rendimiento adecuado que permitiera la implementación de esta red en el proceso de aprovisionamiento de materiales en empresas textiles. La intención era determinar si las redes neuronales artificiales son una técnica fiable para la predicción y si pueden ser adoptadas de manera efectiva por empresas textiles nacionales. Además, se pretende evaluar si los resultados generados por esta red neuronal mejoran el proceso de aprovisionamiento de textiles de manera efectiva. Este modelo pudo utilizarse directamente para simplificar el proceso de pronóstico. Los resultados respaldan el uso de redes neuronales artificiales como técnicas fiables para el pronóstico en empresas textiles y se ha desarrollado un software de predicción basado en esta técnica.

- Chuquimia (2019), autor de la tesis titulada “modelación de redes neuronales para la toma de decisiones de impacto económico, caso de estudio: sector productivo oleaginoso”, donde vieron sobre la necesidad de utilizar modelos sofisticados para tomar decisiones que reduzcan el riesgo en los

pronósticos, especialmente en situaciones en las que diferentes sistemas interactúan de manera compleja. Se enfoca en sectores que impulsan la economía y utiliza la matriz de Leontief para identificar sectores estratégicos e independientes.

El análisis se realiza desde la perspectiva de oferta y demanda, considerando cómo ciertos sectores influyen en otros, debido a la cantidad de insumos que utilizan. Esto se conoce como encadenamientos hacia atrás o hacia adelante. Los coeficientes tecnológicos de demanda y oferta se utilizan para calcular estadísticas de encadenamiento, lo que ayuda a clasificar actividades económicas en sectores clave, estratégicos, independientes e impulsores.

La investigación se realizó mediante un enfoque descriptivo con el objetivo de caracterizar un fenómeno de arbitrariedad presente en los datos analizados. El método de investigación que aplicaron es el experimental, que se considera el más complejo y efectivo en términos de métodos empíricos.

El diseño de la investigación fue causal comparativa, que tiene como objetivo descubrir cómo y por qué ocurre un fenómeno al comparar similitudes y diferencias entre diferentes fenómenos. Llegando a la conclusión que la investigación permitió a las empresas dedicadas a la producción de aceites vegetales, desarrollar estrategias a corto plazo con un alto nivel de confianza del 93%, para hacer frente a cambios en la productividad de los cultivos de soya, fluctuaciones en los precios de exportación, ajustes en los precios locales y dinámicas comerciales.

2.1.2. Antecedentes Nacionales.

- Cotacallapa (2022), autor de la tesis titulada “Redes neuronales convolucionales para datos composicionales: Una aplicación a la industria de la moda”, Sus clientes experimentaban dificultades al comprar prendas de vestir en línea, específicamente

relacionadas con la elección de tallas. Según la directora de la empresa, en años anteriores, se enfrentaron a devoluciones de más de 60 millones de dólares debido a errores en la selección de tallas por parte de los consumidores. La solución propuesta abordó la determinación de tallas de prendas de vestir mediante el desarrollo de un modelo teórico de redes neuronales convolucionales. Este modelo, diseñado para identificar la talla adecuada a partir de imágenes de la silueta de una persona, utilizó veinte puntos de referencia visuales clave, como la cintura y las mangas. Se logró una precisión del 98.6% en la predicción de tallas utilizando modelos como Ridge, SVR, Random Forest y Gradient Boosting, con píxeles de imagen como variables independientes. Aunque los resultados fueron prometedores, con un conjunto de datos de entrenamiento de alrededor de 25 mil imágenes y 4 mil para pruebas, se señaló la posibilidad de mejorar la calidad de la predicción con un conjunto de datos más extenso. Además, las pruebas con imágenes originales de vestidos no cumplieron expectativas debido a la falta de información sobre el tamaño real de las prendas en las imágenes originales.

- Carrión (2018), autor de la tesis pregrado titulado “Aplicación de redes neuronales artificiales para la predicción de la recuperación de planta concentradora en Minsur s.a. – unidad san Rafael”, en el cual la mejora y optimización de los procesos en la extracción de minerales se volvieron cada vez más cruciales debido a la necesidad de abordar la escasez cuantitativa (tonelaje) y cualitativa (bajas leyes) de minerales. Con el objetivo de lograr recuperaciones óptimas al final de cada turno, se implementaron técnicas mejoradas. Durante el entrenamiento de las Redes Neuronales Artificiales (RNA) con datos históricos, se realizaron cuatro pruebas, y en la última, los resultados fueron adecuados y confiables, sin errores notables. En comparación, los métodos estadísticos de regresión lineal presentaron errores inmodificables una vez obtenidos, a diferencia de las RNA, que permiten reducir

los errores, si es necesario, mediante iteraciones adicionales durante el entrenamiento.

- Cantuta y Roncal (2020), autores de la tesis titulada “Redes neuronales recurrentes para la predicción de casos por Covid-19 usando datos abiertos en el Perú, 2020”, Emplearon un sistema de inteligencia para pronosticar casos de COVID-19 utilizando datos disponibles en Perú durante el año 2020. Con el propósito de mejorar la precisión del modelo predictivo, se aplicó un método que evaluó diversas arquitecturas y seleccionó aquella que ofrecía la predicción con el error cuadrático medio más bajo (RMSE). Los resultados revelaron que, en la mayoría de los casos, las arquitecturas híbridas superaron a las no híbridas al proporcionar un RMSE más bajo. El proyecto se desarrolló utilizando la metodología SCRUM. Se observó un aumento promedio en la tasa de mortalidad de COVID-19 por cada 10,000 habitantes entre los datos recopilados y las predicciones, así como una variación media del 0.03% en la tasa de letalidad. Concluyeron que el uso de redes neuronales recurrentes (RNN) híbridas permitió obtener predicciones cercanas a la realidad para casos confirmados y fallecidos, a pesar de la limitada cantidad de datos de entrenamiento.

2.2. Bases teóricas:

2.2.1 Sistema inteligente

Según el autor, (Montemayor,2013), los sistemas inteligentes son un concepto amplio que se refiere a sistemas o dispositivos que pueden aprender, adaptarse y tomar decisiones autónomas basadas en datos y experiencias previas. Estos sistemas utilizan técnicas de inteligencia artificial, como el aprendizaje automático y el procesamiento del lenguaje natural, para realizar tareas que normalmente requerirían la intervención humana. Se muestra algunos aspectos clave del concepto de sistemas inteligentes:

- Aprendizaje automático: Los sistemas inteligentes pueden aprender de datos y experiencias previas para mejorar su desempeño en

tareas específicas. Esto implica la capacidad de reconocer patrones, tomar decisiones y adaptarse a situaciones cambiantes.

- Toma de decisiones autónomas: Estos sistemas tienen la capacidad de tomar decisiones sin intervención humana directa. Pueden evaluar información y tomar medidas en función de sus algoritmos y modelos internos.

- Adaptabilidad: Los sistemas inteligentes pueden adaptarse a nuevas circunstancias y entornos. Pueden ajustar sus comportamientos y decisiones a medida que cambian las condiciones.

- Procesamiento de datos complejos: Los sistemas inteligentes pueden manejar grandes cantidades de datos y realizar análisis avanzados para extraer información valiosa. Esto incluye la capacidad de procesar datos no estructurados, como texto, imágenes y voz.

- Interacción con humanos: Muchos sistemas inteligentes están diseñados para interactuar con humanos de manera natural. Esto puede incluir chatbots, asistentes virtuales y sistemas de recomendación personalizados.

- Aplicaciones diversas: Los sistemas inteligentes se utilizan en una amplia variedad de campos, desde la medicina y la atención al cliente hasta la conducción autónoma, la automatización industrial y la gestión de datos.

- Mejora continua: Estos sistemas tienen la capacidad de mejorar con el tiempo a medida que acumulan más datos y experiencias. El aprendizaje constante es una característica importante de los sistemas inteligentes.

Así mismo están presentes en una variedad de aplicaciones y campos. Aquí tienes algunos ejemplos de sistemas inteligentes

- Asistentes de voz
- Reconocimiento de voz y texto
- Recomendación de contenido:
- Vehículos autónomos:
- Detección de fraudes:

- Diagnóstico médico:
- Robótica avanzada:
- Reconocimiento de imágenes:
- Predicciones a hechos
- Traducción automática
- Monitorización del hogar inteligente, entre otros
- Método de pronósticos.

Según el autor, (Montemayor, 2013), Los métodos de pronóstico buscan prever el valor futuro de una variable para reducir la incertidumbre en la toma de decisiones. Su utilidad radica en que el costo de realizar estos pronósticos sea inferior a los beneficios obtenidos al tomar decisiones con mayor certeza.

Existen dos categorías de métodos de pronóstico: cualitativos y cuantitativos. Los métodos cualitativos dependen de opiniones de expertos y se utilizan cuando no hay datos históricos disponibles. Por otro lado, los métodos cuantitativos, preferidos cuando se tienen datos históricos, emplean enfoques matemáticos y estadísticos para ofrecer pronósticos respaldados por evidencia objetiva (Montemayor, 2013).

2.2.1.1. Características del pronóstico

Según (Montemayor, 2013) Pronosticar implica hacer predicciones sobre el valor futuro de una variable, basándose en datos históricos, la opinión de expertos o una combinación de ambos.

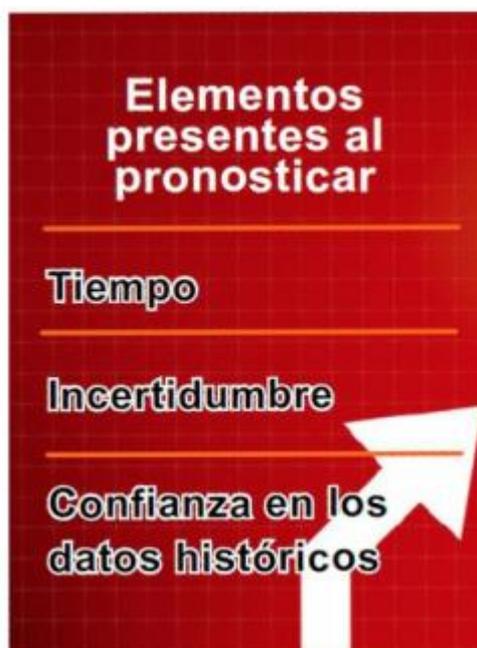
Los pronósticos son herramientas valiosas para la toma de decisiones, ya que contribuyen a reducir la incertidumbre sobre eventos futuros. Sin embargo, su utilidad depende de que los beneficios de tomar decisiones informadas superen los costos de realizar el pronóstico. Cuando se realizan pronósticos, es crucial tener en cuenta factores como el tiempo, la incertidumbre y la confianza en los datos históricos.

En ausencia de datos históricos, se puede recurrir al juicio de expertos, pero es esencial evitar sesgos y mantener un enfoque objetivo. Cuando se disponga de datos históricos, se recomienda utilizar métodos cuantitativos para generar pronósticos objetivos,

aunque aún se puede considerar el aporte de expertos, especialmente cuando los datos son limitados o el pasado no es un buen indicador del futuro.

Figura 3

Elementos presentes al pronosticar



Fuente 3: (autor Montemayor)

2.2.1.2 Métodos de pronósticos cualitativos

2.2.1.2.1 Pronóstico visionario

Según (Montemayor, 2013), estos pronósticos se basan en la información disponible o en la experiencia personal. Por ejemplo, cualquiera puede predecir la temperatura en su ciudad al día siguiente. Sin embargo, hacer pronósticos sobre temas más complejos, como el precio futuro de acciones de una empresa, es más difícil y generalmente se basa en la experiencia y la reputación de un experto en ese campo.

2.2.1.2.2. Analogía histórica

Según (Montemayor, 2013), se enfatiza la utilidad de la experiencia pasada como referencia para pronosticar el rendimiento en un nuevo mercado similar. Por ejemplo, si una empresa tuvo éxito al introducir una marca de panecillos de fresa en una ciudad, puede usar esa experiencia y el conocimiento

acumulado para prever las ventas en un nuevo mercado con características comparables.

2.2.1.2.3. Consenso de un panel

Según (Montemayor, 2013), este enfoque de pronóstico se fundamenta en la colaboración de un conjunto de expertos para llegar a un consenso, lo que puede ser beneficioso, pero también susceptible a sesgos. Para aplicarlo, es necesario reunir a estos expertos y analizar la situación, llegando a un consenso sobre los valores futuros de las variables a predecir. La ventaja de este enfoque es que se aprovecha la diversidad de perspectivas de los expertos, lo que conduce a pronósticos más precisos.

2.2.1.2.4. Método Delphi

Según (Montemayor, 2013), se presenta como una opción al método de consenso de panel, donde un grupo de expertos emite pronósticos de manera anónima y sin interacción mutua. Este procedimiento consta de múltiples etapas iterativas con el objetivo de reducir la variabilidad en los pronósticos y alcanzar un consenso.

En la primera etapa, los expertos envían sus pronósticos al moderador, quien realiza un análisis estadístico y comparte los resultados clave con el grupo. En la segunda etapa, los expertos ajustan sus pronósticos según esta retroalimentación y los reenvían al moderador para análisis adicional. Este proceso iterativo se repite para reducir la variabilidad en los pronósticos y lograr un consenso.

2.2.1.2. Métodos de pronósticos cuantitativos

2.2.3.1. Métodos de suavización

Según (Montemayor, 2013), los métodos de suavización son técnicas empleadas para prever valores futuros de una variable de interés basándose en los patrones históricos de la serie. Estos métodos parten de la premisa de que el valor futuro de la variable en el período $t+1$ se ve afectado por los valores de

la serie en el período actual (t), el período anterior (t-1) y períodos previos.

La fórmula general que subyace a estos métodos es:

$$Y_{T+1} = f(Y_T + Y_{T-1} + Y_{T-2} + Y_{T-3} \dots)$$

Según (Montemayor, 2013), resulta fundamental identificar el patrón presente en la serie, que puede ser horizontal (estacionario), de tendencia, cíclico, estacional o incluir variaciones irregulares (aleatorias). Los métodos de suavización se clasifican en diferentes categorías según el patrón particular de la serie:

- Para el patrón horizontal (estacionario):
 - Promedio móvil simple.
 - Promedio móvil de k periodos.
 - Suavización exponencial simple.
- Para el patrón de tendencia:
 - Promedio móvil doble.
 - Suavización exponencial doble: Método de Brown.
 - Suavización exponencial doble: Método de Holt.
- Para el patrón estacional:
 - Suavización exponencial triple:

2.2.3.2. Método de Winters.

Según (Montemayor, 2013), estos métodos de suavización son herramientas valiosas para pronosticar variables en función de los patrones observados en los datos históricos. Cada método se selecciona en función del tipo de patrón presente en la serie de datos.

2.2.2. Redes Neuronales Artificiales

Según (Oliveira Colabone y colaboradores, 2015), sostienen que estas redes se entrenan utilizando entradas derivadas de situaciones tanto externas como internas del sistema, las cuales son ponderadas por coeficientes asignados de manera aleatoria.

En relación con este tema, Callejas y colaboradores (2013), sostienen que las RNA se inspiran en el procesamiento de información del

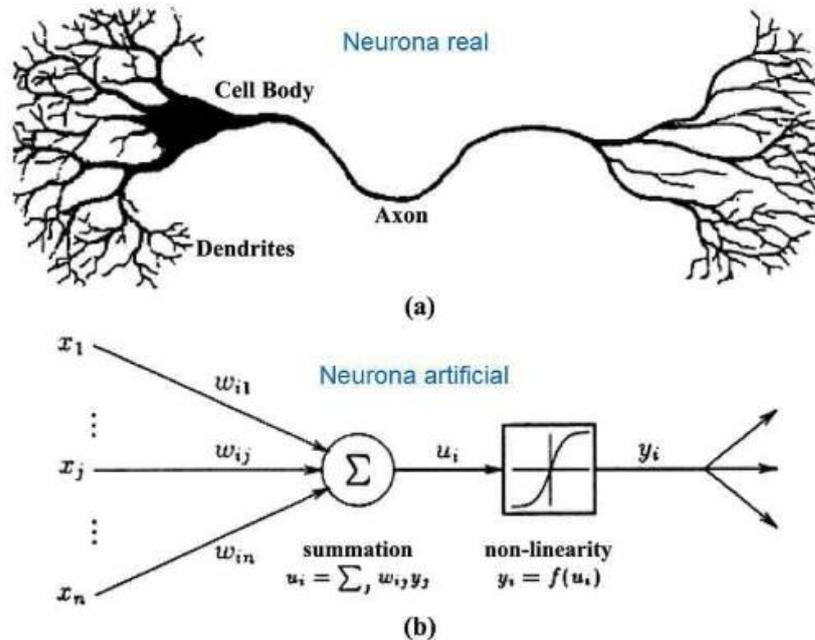
sistema nervioso biológico, imitando así el comportamiento cerebral. Por otro lado, Molino (2014) destaca que estas redes tienen la capacidad de resolver funciones altamente no lineales en periodos cortos, ya que aprenden de datos difíciles de expresar matemáticamente. En consecuencia, se considera que estas redes son herramientas altamente eficaces para el análisis de señales y la modelación de sistemas.

Esteban (2009) señala que las redes neuronales artificiales son programas informáticos diseñados para replicar el funcionamiento de las neuronas biológicas. Estas redes, construidas con múltiples elementos interconectados, tienen una organización jerárquica que les permite interactuar con su entorno. En esencia, son sistemas informáticos que procesan información en respuesta a estímulos externos.

Es importante resaltar que las Redes Neuronales Artificiales son una herramienta versátil para abordar problemas, ya sea de manera independiente o en conjunto con otros enfoques, especialmente en situaciones donde la disponibilidad de datos supera el conocimiento preexistente. Estas redes encuentran aplicaciones en diversas disciplinas, como neurociencias, matemáticas, estadísticas, física, ciencias de la computación e ingeniería. Se emplean en diversos campos, como modelado, análisis de series temporales, reconocimiento de patrones, procesamiento de señales y control. La característica clave de las redes neuronales es su capacidad para aprender de los datos de entrada, ya sea con o sin la orientación de un experto, según lo mencionado por Simón (1999).

Figura 4

Comparación de una Neurona Biológica y una Artificial



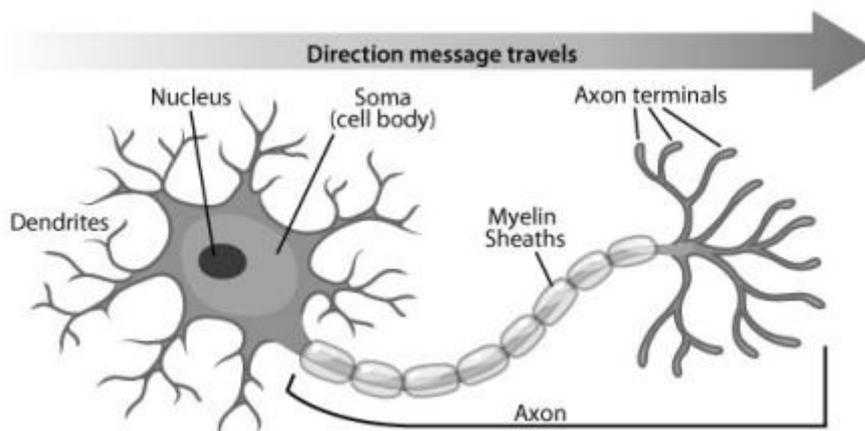
Fuente (autor Alarcón; 2021)

2.2.2.1. Estructura básica de una Red Neuronal Artificial

La composición de una red neuronal incluye diversas partes, entre ellas el axón que representa la salida de la neurona. Este se conecta con las dendritas, que desempeñan el papel de entradas, mediante las sinapsis. Vale la pena resaltar que, durante el proceso de aprendizaje de la red neuronal, estas sinapsis pueden experimentar modificaciones.

Figura 5

Red Neuronal Artificial



Fuente: (Lexly Vanessa Sosa Jerez y Laura Camila Zamora Alvarado, 2022)

2.2.2.2 Tipos de Redes Neuronales Artificiales

2.2.2.2.1 Adaline y madaline

Desarrollados por Bernard Widrows y Marcian Hoff, (Universidad de Stanford). Adaline, cuyo nombre significa "Elemento Lineal Adaptativo", fue diseñado para reconocer patrones binarios. Su operación se basa en la suma de los pesos de las entradas para generar una salida de 0 o 1, dependiendo de si supera un umbral específico. La importancia de Adaline radica en que sirve como base para la construcción de redes neuronales más avanzadas.

Múltiples unidades Adaline se organizan en capas para formar arquitecturas más complejas, dando lugar a una red llamada Madaline, que significa "Elemento Lineal Adaptativo Múltiple". Madaline, una de las primeras redes neuronales multicapa con elementos adaptables, se compone de dos o más unidades Adaline. Originalmente utilizada como un filtro para eliminar el eco en líneas telefónicas, Madaline fue desarrollada posteriormente a Adaline para abordar problemas más complejos que no son linealmente separables.

2.2.2.2.2 Red Backpropagation

De acuerdo con Cruz Beltrán (2008), este tipo de red, establecido en 1986, se basa en el aprendizaje de la asociación de patrones. El proceso de retro propagación tiene la capacidad de generalizar comportamientos al utilizar un conjunto de muestras seleccionadas como patrones de entrenamiento. Esta característica lo hace similar al funcionamiento del cerebro humano, lo que implica que la red debe pasar por una fase de aprendizaje. En otras palabras, esta red necesita identificar una función de comportamiento que se ajuste a los valores muestreados y al comportamiento del sistema, minimizando el error.

2.2.2.2.3 Perceptrón multicapa

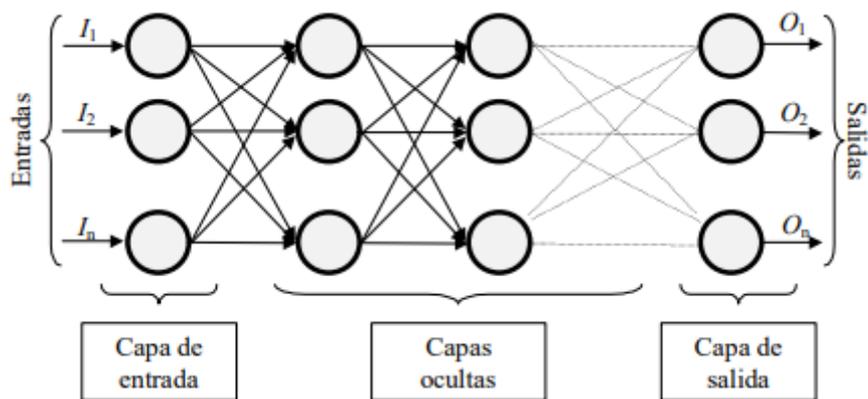
De acuerdo a la investigación realizada por Barbosa y Monzón (2001), el perceptrón consta de un conjunto de neuronas que se comunican mediante el envío de señales entre ellas. Aunque posee un conjunto de conexiones bien desarrollado, su capacidad para reconocer caracteres complejos es limitada.

En cuanto al perceptrón multicapa, presenta ciertos aspectos de relevancia que se describen a continuación:

- El perceptrón se compone de unidades de procesamiento, semejantes a células o neuronas.
- Cada unidad de procesamiento posee un estado de activación que influye en la salida de la neurona.
- Conexiones entre las neuronas existen, y cada conexión tiene un peso que determina el impacto de la señal en la neurona receptora.
- El proceso de propagación, regido por una regla específica, establece la entrada efectiva de una neurona a partir de las entradas externas.
- El sistema del perceptrón funciona en un entorno que suministra las señales de entrada con las que la neurona realiza su trabajo.

Figura 6

Estructura del Perceptrón Multicapa



Fuente: (matich, 2001)

Antes de estudiar las redes neuronales, es esencial comprender el funcionamiento de las neuronas y su aplicación en una red neuronal. Se realiza una comparación entre una neurona biológica y una artificial para destacar las similitudes, como la presencia de entradas, pesos y salidas.

2.2.2.3 Función de activación de una Red Neuronal Artificial

Según (Matich, Universidad Tecnológica Nacional, 2001), la función de activación desempeña un papel crucial al calcular el nivel de actividad de una neurona. Su tarea consiste en transformar la entrada

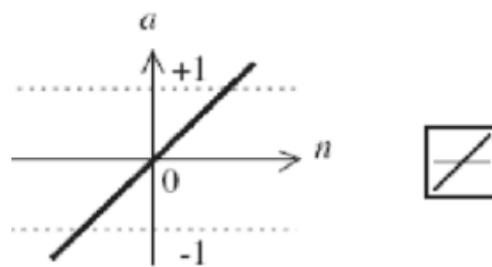
global, que implica restar el umbral (Q), en un valor de activación típicamente dentro del rango de 0 a 1 o de -1 a 1. Esta gama refleja la capacidad de una neurona para estar completamente inactiva (0 o -1) o activa (1). La función de activación se obtiene a partir de la entrada global (G_{in}) menos el umbral (Θ_i).

A continuación, se detallan las funciones de activación más frecuentemente utilizadas.

- **Función lineal:** Produce una salida lineal que depende de la entrada global dentro de un rango específico. Por encima o por debajo de ese rango, la salida es 1 o -1, respectivamente.

Figura 7

Esquema de la función lineal



$$a = \text{purelin}(n)$$

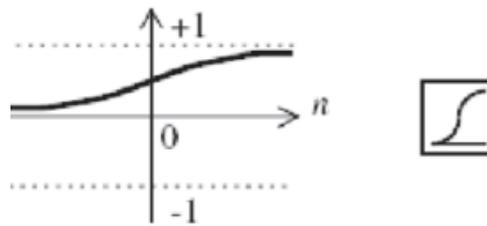
$$a = n$$

Nota. (Francisco cubillos, 2007)

- **Función sigmoidea:** Genera valores de salida en el rango de 0 a 1 y su pendiente puede ajustarse mediante un parámetro. A medida que se modifica este parámetro, se cambia la pendiente de la función.

Figura 8

Esquema de la función sigmoidea



$$a = \text{logsig}(n)$$

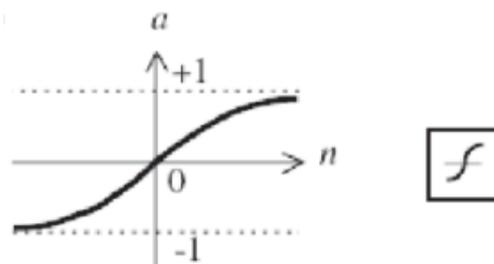
$$a = \frac{1}{1 + e^{-n}}$$

Nota: (Francisco cubillos, 2007)

- **Función tangente hiperbólica:** Produce valores de salida en el rango de -1 a 1, y al igual que la función sigmoidea, la pendiente puede ajustarse variando un parámetro.

Figura 9

Esquema de la Función tangente hiperbólica



$$a = \text{tansig}(n)$$

$$a = 0 \quad n < 0$$

$$a = 1 \quad n \geq 0$$

Nota: (Francisco cubillos, 2007)

La elección de estas funciones de activación se realiza considerando la naturaleza específica del problema a abordar y se comparan a menudo con la aceleración de un automóvil, donde se requiere más potencia para acelerar tanto en velocidades bajas como altas, y menos en el punto intermedio. (Stanislav Korotky, 2023)

2.2.2.4 Mecanismos de aprendizaje.

Según (Francisco cubillos, 2007), en una red neuronal, los datos de entrada se procesan para generar una salida deseada. Durante el proceso de aprendizaje, la estructura de la red y las funciones de las neuronas permanecen invariables, pero los pesos en las conexiones pueden ajustarse en respuesta a la información de entrada. A diferencia de los sistemas biológicos, en las redes neuronales artificiales, la creación de una nueva conexión se refleja en un peso distinto de cero, y una conexión se considera destruida cuando su peso se ajusta a cero. El proceso de aprendizaje implica la modificación de los valores de los pesos en las conexiones para incorporar nueva información.

Existen dos métodos de aprendizaje fundamentales que se pueden identificar:

2.2.2.4.1 Aprendizaje supervisado.

Según (Tang, 2014), En el aprendizaje supervisado, un agente externo guía y controla el entrenamiento de la red neuronal, determinando las respuestas deseadas a partir de entradas específicas.

Cuando la salida de la red no coincide con la respuesta esperada, se ajustan los pesos de las conexiones para acercar la salida a la deseada, corrigiendo y mejorando así el rendimiento de la red. En el aprendizaje supervisado, se emplean tres enfoques principales:

- **Corrección de error.**

En este método, se ajustan los pesos de las conexiones de la red según la discrepancia entre los valores deseados y los valores de salida obtenidos. En otras palabras, los pesos se adaptan en función del error cometido en la salida.

- **Refuerzo.**

Este método de aprendizaje supervisado es más lento que el anterior y se basa en la idea de no tener un ejemplo completo del comportamiento deseado. Durante el entrenamiento, no se especifica de manera exacta la salida esperada de la red para una entrada determinada.

- **Estocástico.**

Este proceso consiste en realizar ajustes aleatorios en los pesos de las conexiones de la red y evaluar cómo estos afectan el resultado deseado, teniendo en cuenta distribuciones de probabilidad.

2.2.2.4.2 Aprendizaje no supervisado.

Según (Flórez & Fernández, 2008), Este método, también denominado auto supervisado, opera sin intervención externa para ajustar sus conexiones neuronales. Estas redes no reciben retroalimentación del entorno que indique si sus respuestas a entradas son correctas. Su tarea principal es descubrir patrones, regularidades o categorías en los datos de entrada sin orientación externa. La interpretación de su salida puede variar según su estructura y algoritmo de aprendizaje, pudiendo representar similitud, categorización o agrupamiento de la información.

2.2.3 Aplicaciones de las Redes Neuronales.

Existen diversas variedades de RN, cada una con su aplicación específica. Cada tipo se adapta mejor a ciertas aplicaciones particulares. Algunos ejemplos de aplicaciones comerciales incluyen:

- **Medicina:** (Alberto Delgado, Universidad Nacional de Colombia, 1999) aplicaron la red neuronal clasificadora para desarrollar un diagnóstico automático de reacciones alérgicas y pseudoalérgicas, que son clínicamente similares a las reacciones alérgicas, lo que agrega complejidad al problema
- **Economía:** (Fernando Villada, Nicolás Muñoz y Edwin García, 2012), para prever los precios de dos de las principales acciones negociadas en el mercado de valores colombiano, se lleva a cabo un estudio enfocado en las acciones de Ecopetrol y Preferencial Bancolombia. Estas empresas cotizan tanto en las bolsas de valores de Colombia como en la de Nueva York.
- **Automotriz:** (Departamento de la universidad de Málaga, 2018) se aplicó para resolver problemas de seguridad en automóviles, el experto explica que se buscó reproducir de forma artificial cómo piensa la mente, para dirigir los movimientos utilizaron un número limitado de neuronas interconectadas, tomando inspiración del funcionamiento humano y emulando patrones naturales. La idea

detrás de estos modelos es imitar comportamientos naturales, donde la naturaleza a menudo opera sin un proceso consciente, como cuando se conduce un automóvil.

- **Reconocimiento facial:** Seguridad: (Felipe, Víctor y Efraín, 2016) Reconocimiento facial mediante redes neuronales, en su estudio, realizaron un análisis comparativo de varios algoritmos, incluyendo Hopfield, Backpropagation y el Algoritmo PCA, concluyendo que la Red Neuronal de Hopfield es la más adecuada para el reconocimiento facial. Esto se basó en su mayor porcentaje de efectividad, alcanzando un 80%.

2.2.4 Metodología

2.2.4.1 Metodología de Kaastra y Boyd

[I. Kaastra, M. Boyd y J. Ebeling, 1996] han presentado una metodología que describe un proceso para la fabricación de un modelo de RN destinado al pronóstico temporal en el ámbito financiero y económico. En la elaboración del modelo que se abordará en este trabajo, se tomará como base este procedimiento, enriqueciéndolo con aportes de investigaciones más actuales. Este enfoque se compone de una secuencia de 8 procesos o fases:

2.2.4.1.1. Fase 1: Selección de las variables

La elección de la frecuencia de los datos está condicionada por los objetivos del investigador. Esto significa que, si el investigador tiene un horizonte a largo plazo, optará por utilizar datos anuales o mensuales. En cambio, si el enfoque es a corto plazo, se recurrirá a datos históricos (Kaastra y Boyd, 1996).

2.2.4.1.2 Fase 2: Recolección de los datos

La fuente de la data, deben provenir de un proveedor confiable, ya que la falta de fiabilidad en la información podría generar dificultades para que la red neuronal produzca con éxito una solución al problema (Kaastra & Boyd, 1996).

2.2.4.1.3 Fase 3: Preparación de los datos.

Se refiere a la manipulación de los datos en bruto de las variables de entrada y salida con el objetivo principal de reducir el ruido. Esto facilita que la red neuronal identifique tendencias y

patrones relevantes de manera más efectiva, como señalan Kaastra y Boyd (1996).

A continuación, se presentan algunas de las técnicas de normalización utilizadas.

Figura 10

Tipos de normalizaciones

Normalización Estadística	$z = \frac{x - \mu}{\sigma} \rightarrow N(0, 1)$	<p>x: es el dato a normalizar u: es la media del conjunto de datos, q: es la desviación típica de los datos z: es el resultado ya estandarizado</p>
Normalización Máximos y Mínimos (Max-Min)	$x_{normalized} = \frac{x - x_{min}}{x_{max} - x_{min}} \in [0,1]$	<p>x: es el dato a estandarizar Xmin: mínimo valor de todo el set de datos Xmax: máximo valor de todo el set de datos Xnormaliz: resultado estandarizado</p>
Normalización Sigmoide	$z = \frac{1}{1 + e^{\frac{(x - \bar{a})}{\sigma}}}$	<p>x: valor a estandarizar ā: media del conjunto de datos q: desviación típica de los datos z: resultado ya estandarizado.</p>

2.2.4.1.4 Fase 4 Conjuntos de entrenamiento, prueba y validación.

Datos de Entrenamiento

Según (Kaastra y Boyd, 1996), La colección de datos empleada para llevar a cabo el proceso de entrenamiento. Es el grupo más extenso de datos que se utiliza para capacitar a la RN, facilitándole la capacidad de reconocer y aprender los patrones que existen en los datos.

Datos de Prueba

Según (Kaastra & Boyd, 1996), se emplea para medir la capacidad de generalización de la RN, es decir, su habilidad para aplicar lo que ha aprendido a datos que no formaron parte del proceso de entrenamiento.

Datos de Validación

Según (Kaastra & Boyd, 1996), cumple el propósito de evaluar el desempeño de una RN que fue previamente

entrenada. Este conjunto se compone de los datos más recientes dentro del conjunto completo.

2.2.4.1.5 Fase 5: Aplicación de los paradigmas de las redes neuronales.

Según [Fu, 1994], [Walczak, 1999] y [Widrow et al, 1994], las redes neuronales han evidenciado su eficacia en la aproximación de funciones y la identificación de relaciones en conjuntos de datos con comportamientos complejos. Específicamente, el enfoque de retro propagación ha evidenciado su utilidad en la predicción de variables de naturaleza financiera.

a. Número de Neuronas de Entrada.

Según (Kaastra & Boyd, 1996), En cuanto al número de neuronas de entrada, se realizaron varias pruebas para encontrar el número que mejor proporcionó la estimación

b. Número de Capas Ocultas.

Según [Fu, 1994], [Walczak, 1999] y [Widrow et al, 1994], si se plantea la adición de más capas ocultas, es esencial tener en cuenta que esto implicará un aumento en el tiempo necesario para que nuestro procesador complete los cálculos.

c. Número de Neuronas Ocultas.

Según (Álvaro Fernando, 2000), es posible comenzar con 1 red que tenga 1 sola capa oculta mínima (con una neurona) y luego ir agregando neuronas a esta capa hasta que la reducción del error ya no sea suficiente o no alcance los objetivos deseados. Cuando esto sucede, se considera la posibilidad de añadir otra capa a la red.

También se puede optar por empezar añadiendo capas ocultas desde el principio. La elección de cuál enfoque seguir depende de la complejidad del problema y de la experiencia del diseñador de la RNA.

d. Número de Neuronas de Salida.

Según (Kaastra & Boyd, 1996), depende del resultado específico que el investigador desea obtener. Por ejemplo, si

se busca pronosticar un conjunto de tiempo en el futuro, será necesario contar con un número equivalente de neuronas de salida al periodo de tiempo que se pretende pronosticar. En cambio, si el objetivo es predecir únicamente el mes siguiente, bastará con una sola neurona de salida.

2.2.4.1.6 Fase 6: Criterios de evaluación.

Según (Kaastra & Boyd, 1996), su propósito es evaluar la habilidad de generalización de una RNA, para lo cual se recurre a funciones de error que señalan la diferencia porcentual entre las predicciones de la red y los datos reales

Las funciones de errores, son:

Figura 11

Función de errores

Error Cuadrático medio	
$\frac{1}{n} \sum_{i=1}^n (\bar{y}_i - y_j)^2$	<u>Calcula la media diferencial absoluta</u> \bar{y}_i : representado dato real y_j : representado por el dato obtenido por la red neuronal
Error Absoluto medio	
$\frac{\sum_{i=1}^n y_i - x_i }{n}$	<u>calcula la media diferencial absoluta</u> y_i : dato real representado x_i : representado por el dato obtenido por la red neuronal
Error Porcentual Absoluto medio	
$\frac{1}{n} \sum_{T=1}^n \left \frac{A_T - F_T}{A_T} \right $	<u>calcula el error porcentual absoluto medio</u> A_t : representado por el dato de real F_t : representado por el dato obtenido por la red neuronal artificial

2.2.4.1.7 Fase 7: Entrenamiento de la red neuronal.

Según (Vidaurre, 2012), El entrenamiento de la red neuronal se fundamenta en el descenso de gradiente mediante el algoritmo de retro propagación. El proceso de entrenamiento concluye cuando se cumplen tres condiciones cruciales: al superar el límite de iteraciones establecido para las épocas, al alcanzar el Error Cuadrático Medio (ECM) definido como objetivo, o al experimentar un aumento

sostenido del error promedio de la función de evaluación durante un número específico de iteraciones (requiriendo la existencia de un conjunto de validación). En cada uno de estos casos, los pesos y sesgos resultantes son aquellos que minimizan el error medido por la función de evaluación

- **Número de Iteraciones (Épocas) Entrenamiento**

Según (Vidaurre, 2012), a veces, aumentar el número de iteraciones en el proceso de entrenamiento puede disminuir el error durante el entrenamiento, aunque esto no asegura necesariamente que se alcanzará el error más bajo en el conjunto de prueba.

- **Tasa de Aprendizaje y Momento**

La tasa de aprendizaje, según (Vidaurre, 2012), es una constante que regula el tamaño de los ajustes de los pesos en una red neuronal, influyendo en la reducción del error. Para acelerar el entrenamiento sin generar oscilaciones, se puede usar un término de momento en la regla de retro propagación, controlando cómo los cambios pasados en los pesos afectan los cambios actuales y contribuyendo a la estabilidad y eficiencia del proceso de entrenamiento.

2.2.4.1.8 Fase 8: Implementación.

Se procede a implementar el modelo utilizando la técnica de RNA. Aplicando a los datos que deseamos analizar en la investigación

Así mismo se hace uso del software para la implementación del modelo

2.3 Definición de términos básicos:

2.3.1. Aprendizaje

Según (Palmer y Montalvo, 1999), cualquier proceso de aprendizaje implica realizar ajustes en estas conexiones. El aprendizaje en una red neuronal implica ajustar los pesos en respuesta a un conjunto de ejemplos llamado conjunto de entrenamiento. Existen diversas reglas de aprendizaje, algoritmos matemáticos, que modifican los pesos para que la red aprenda a resolver problemas específicos. Estas reglas se

dividen principalmente en aprendizaje supervisado, donde se proporcionan ejemplos con salidas deseadas, y aprendizaje no supervisado, donde la red se adapta a patrones sin salidas específicas predefinidas.

2.3.2. Backpropagation

Según (McCord Nelson e Illingworth, 1991), es un método computacional utilizado en algoritmos diseñados para entrenar el funcionamiento de redes neuronales artificiales. En términos generales, las RNA buscan imitar la función de las redes de neuronas biológicas. Al igual que en el ser humano, donde las neuronas están interconectadas, en las redes neuronales artificiales, se produce un fenómeno similar: las neuronas artificiales se conectan y organizan en diferentes niveles conocidos como capas. Estas neuronas o nodos reciben datos desde fuentes externas o de otras neuronas, los procesan y generan valores de salida que se transmiten a otras neuronas o se entregan como resultados fuera de la red. Este proceso de retro propagación de errores se emplea para ajustar los pesos de las conexiones entre las neuronas durante el entrenamiento de la red, permitiendo que la red neuronal aprenda y mejore su capacidad para realizar tareas específicas.

2.3.3. Predicción

Según (Palmer y Montalvo 1999), capacidad de estimar o anticipar un evento futuro o un resultado en función de información disponible en el presente y en el pasado. Implica el uso de datos históricos o actuales, así como de modelos matemáticos, estadísticos o algorítmicos para hacer proyecciones sobre cómo se desarrollará una situación o evento en el futuro. La predicción desempeña un papel fundamental en diversas disciplinas, abarcando desde la meteorología y la economía hasta la inteligencia artificial y la toma estratégica de decisiones en el ámbito empresarial.

2.3.4. Redes Neuronales de tipo biológico

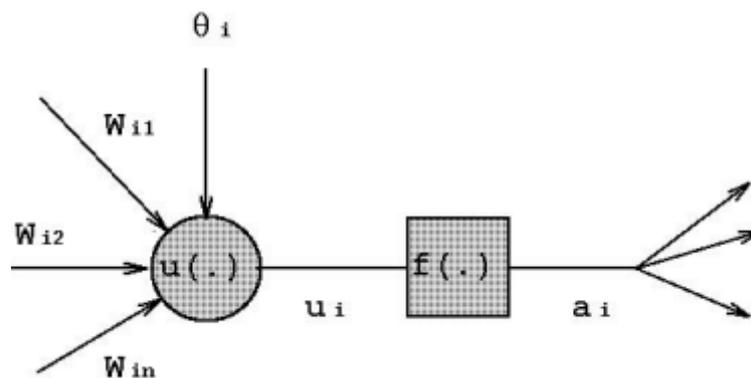
A pesar de que las neuronas biológicas tienen tiempos de respuesta significativamente más lentos en comparación con las computadoras actuales, poseen una conectividad que supera en miles de

veces a la de las neuronas artificiales. El propósito principal de las RN de tipo biológico es realizar operaciones relacionadas con la síntesis y procesamiento de información, inspirándose en los sistemas biológicos (Diazaraque, 2012).

La interacción exacta entre dos neuronas no se comprende completamente y varía según el tipo específico de cada neurona. En términos generales, una neurona transmite su señal de salida a otras a través de su axón, transfiriendo información mediante diferencias de potencial eléctrico (Diazaraque, 2012).

Figura 12

Esquema de una neurona



Nota. (diazaraque, 2012)

2.3.5 Redes Neuronales Artificiales (ANN)

Son sistemas de procesamiento de información inspirados en las redes neuronales biológicas, pero con características específicas que las hacen útiles en diversas aplicaciones, como el aprendizaje automático y la inteligencia artificial (Marín, Madrid 2012).

Características principales de las ANN:

- **Auto-Organización y Adaptabilidad:** Estas redes neuronales emplean algoritmos de aprendizaje adaptativo y autoorganización.
- **Procesamiento no Lineal:** Las ANN pueden realizar un procesamiento no lineal, lo que aumenta su capacidad para aproximar funciones y clasificar patrones.

- **Procesamiento Paralelo:** Por lo general, cuentan con un gran número de nodos de procesamiento que están altamente interconectados.

CAPÍTULO III

III. DESARROLLO DEL TRABAJO PROFESIONAL

3.1 Determinación y Análisis del Problema

La empresa R&D Corporación, se dedica a la producción y venta de productos relacionados a la moda para mujer, cuenta con un sistema de venta alojada en la nube, pero no tienen una herramienta que les pueda ayudar a planificar que modelos lanzar al mercado y así evitar presentar pérdidas y además de tener más control con el inventario.

Se realizó consultas al representante del área de producción donde se identificó algunos requerimientos que nos servirán para el futuro análisis al momento de realizar la configuración del sistema inteligente con las redes neuronales artificiales.

Se muestra las ventas realizadas desde el periodo del 22/10/2018 – 31/07/2023, visualizando el tipo de prendas y cantidades vendidas por mes.

Tabla 1

Ventas anuales por prenda R&D Corporación.

Tipo de prenda Año	Blusa	Carteras	Casacas	Chompa	Pantalón	Saco	Short	Vestido
2018	1795	59			957	604	11	1042
2019	9405	198		12	5755	4575	81	3606
2020	6571	142	189		2620	1943	104	1537
2021	8938	338	1103	1038	5264	5064	436	1870
2022	13107	221	2172	430	7453	7381	887	1830
2023	7892		1076	56	3873	2812	767	1110
Total general	47708	958	4540	1536	25922	22379	2286	10995

3.2. Aplicación de modelo de solución

La metodología que se aplicó para realizar este trabajo profesional se fundamentó en el estudio que se llevó a cabo por Kaastra y Boyd (1995), enfocados a las 8 fases o procesos.

3.2.1 Proceso 1: Selección de las variables

La elección de la variable para este trabajo de investigación a predecir, en relación con los ocho tipos de prendas vendidas por un rango de tiempo, es decir, será de tipo cuantitativa, ya que representa la cantidad de ventas en todas las tiendas de la empresa R&D Corporación.

3.2.2 Proceso 2: Recolección de los datos

Los datos de recolección fueron descargadas del programa de venta “bsale” que cuenta la empresa almacenada en la nube, analizadas desde el periodo del mes de octubre del 2018 a julio 2023 de lima metropolitana, Perú.

3.2.3 Proceso 3: Preparación de los datos

Para tener un rendimiento óptimo en el modelo de la red neuronal artificial, se realizó agrupar los datos en 8 diferentes tipos de prenda (blusas, carteras, casacas, chompas, pantalones, sacos, shorts y vestidos) ventas por mes, y se aplicó la fórmula de estandarización de máximo y mínimo

3.2.4 Proceso 4: Conjunto de entrenamiento, prueba y validación

Se realiza la partición de todos los datos recopilados en tres fases.

Datos de entrenamiento. Los datos de entrenamiento representado fueron de 58 meses del total de registros obtenidos, esto corresponde a la cantidad de 5 años aproximadamente

Datos de prueba. El conjunto de datos de prueba comprende aproximadamente el 20% del total de registros de venta en el sistema

Datos de validación. El conjunto de datos de validación, comprende los datos más recientes, es decir, datos del presente año, representado al 17% del total de Registros utilizados para el entrenamiento, equivalentes a 7 meses.

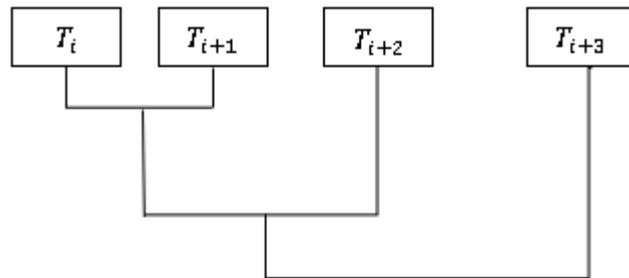
3.2.5 Proceso 5: Aplicación de los paradigmas de las redes neuronales.

Se define la arquitectura de la red neuronal artificial para nuestra investigación, determinando el tamaño de la capa de entrada, la configuración de las capas ocultas, la cantidad de neuronas en estas capas, así como en la capa de salida, y la elección de la función de activación que se utilizará

Número de Neuronas de Entrada

Figura 13

Secuencia de tiempo



En la figura 13, se visualiza los datos T_i , T_{i+1} , T_{i+2} representadas en 3 neuronas en las capas de entrada, en el cual podremos pronosticar los próximos resultados por mes a futuro que está representada por la capa de salida T_{i+3} .

Número de capas ocultas

En la cantidad de capas ocultas, se analizaron con 3 capas ocultas, debido a que su análisis fue más preciso en relación con las cantidades de datos por analizar

Número de Neuronas de la capa oculta

El número de neuronas en la capa oculta son la siguiente: En la Primera capa tiene 16 neuronas y con función de activación ReLu.

En la Segunda capa tiene 16 neuronas y con función ReLu y en la tercera capa tiene 1 neurona y con función de activación lineal, logrando así un buen pronóstico en la red neuronal

Número de Neuronas de Salida

Se eligió utilizar una única neurona en la capa de salida porque es adecuado para la tarea de pronóstico necesaria en este modelo.

Función de Transferencia

La función de activación, como se menciona anteriormente en el número de neuronas ocultas, fue la función de Activación ReLu y lineal.

3.2.6 Proceso 6: Criterio de evaluación

En la evaluación del criterio, se aplicó la evaluación de error cuadrático medio para el estudio del modelo de la RNA.

3.2.7 Proceso 7: Entrenamiento de la RNA

Durante esta fase, se examinan el número de iteraciones y la tasa de aprendizaje para analizar y reducir al mínimo la función de error.

Número de Iteraciones

El número de iteraciones que se analizaron fueron de 1000 iteraciones, debido a que se observó mejor los resultados en el entrenamiento de los datos a evaluar, proporcionó mejores ajustes y la función de errores fue mínima.

Razón de Aprendizaje

Se estableció la métrica de evaluación igual a “mean_absolute_error” y se definió el momento con una cantidad de optimizer='adam' (se utiliza para ajustar los pesos de la red durante el proceso de aprendizaje), como consecuencia se obtuvo en buenos resultados.

3.2.8 Proceso 4: Implementación

Y como último proceso se incluyó el uso de las siguientes herramientas y bibliotecas para la construcción del modelo de la red neuronal artificial:

- Google Colab
- Librerías Numpy, Keras, numpy
- Python

Figura 14

Colab Google, Python, Librería, Keras, Panda, Numpy



Pandas



- Colab: Es una plataforma en línea gratuita de Google que permite escribir y ejecutar código Python en la nube, además de la inclusión de contenido enriquecido. Ofrece la capacidad de configurar recursos computacionales según las necesidades y facilita tareas complejas. También promueve la colaboración al permitir compartir códigos entre equipos de trabajo. (G. L. Baume, 2021)

- Python: Es un lenguaje de programación de alto nivel ampliamente utilizado, conocido por su simplicidad. Es versátil, soporta múltiples paradigmas de programación. Python es especialmente apreciado en áreas como la ciencia de datos, desarrollo web y automatización. (Guido van Rossum, 2015)

- Keras: Biblioteca de código abierto en Python para el desarrollo de redes neuronales y aprendizaje profundo. Es ampliamente utilizado debido a su flexibilidad, capacidad de personalización y compatibilidad con diversos backends como TensorFlow (François Chollet, 2015)

- Panda: Ofrece estructuras de datos flexibles, como DataFrames, que permiten la manipulación y limpieza de datos de manera eficiente. Es esencial para tareas de carga, exploración y transformación de datos en proyectos de análisis y procesamiento de información. (Wes McKinney, 2008)

- Numpy: Proporciona estructuras de datos eficientes y funciones para operaciones numéricas, lo que la convierte en un pilar de la

computación científica y el aprendizaje automático. Permite el procesamiento de datos de manera eficaz, utilizado para análisis de datos. (Travis E. Oliphant, 2006)

Figura 15

Importación de librería Pandas



Se visualiza en la figura 15, las librerías, la cual le pondremos un alias pd y np, a pandas y numpy respectivamente.

Pandas se utiliza para manipular y trabajar con datos tabulares, permite subir todo el registro de la data en formato Excel,

Así mismo, la librería numpy, proporciona soporte para arreglos multidimensionales, funciones matemáticas de alto nivel para operar en estos arreglos y herramientas para trabajar con datos.

Figura 16

Importación y carga de datos a analizar

```
training = pd.read_excel("ventas3.xlsx")
training.head(10)
```

	Año	Fecha	BLUSAS	CARTERA	CASACA	CHOMPAS	ENTERIZO	FALDA	PANTALON	POLO	SACOS	SHORT	VESTIDOS
0	2018	octubre	97	2.0	NaN	NaN	NaN	6	74	NaN	80	NaN	43
1	2018	noviembre	529	21.0	NaN	NaN	NaN	82	348	NaN	323	NaN	326
2	2018	diciembre	1169	36.0	NaN	NaN	NaN	76	535	NaN	201	11.0	673
3	2019	enero	750	14.0	NaN	NaN	NaN	72	401	NaN	104	26.0	432
4	2019	febrero	486	21.0	NaN	NaN	NaN	41	282	NaN	57	22.0	452
5	2019	marzo	635	19.0	NaN	NaN	NaN	41	343	NaN	97	13.0	360
6	2019	abril	502	18.0	NaN	NaN	NaN	75	372	NaN	214	2.0	238
7	2019	mayo	974	14.0	NaN	1.0	NaN	84	616	NaN	454	NaN	204
8	2019	junio	586	6.0	NaN	4.0	NaN	71	576	NaN	733	NaN	125
9	2019	julio	547	17.0	NaN	4.0	NaN	81	469	NaN	679	NaN	87

Carga datos desde un archivo Excel llamado "ventas3.xlsx" en un DataFrame de Pandas llamado training.

Los datos son obtenidos a través de la descarga de los reportes de la web bsale que cuanta la empresa R&D Corporación, fueron analizadas y separadas por tiempo y tipos de prendas (8 en total) que serán analizadas una por una, empezando por la primera prenda "BLUSAS"

Después de la partición de los datos, procederemos a la estandarización de los mismos. Posteriormente, se llevará a cabo un análisis detallado de las entradas y salidas de la red.

Figura 17

Llenado de valores con la media

```
# Llenar los valores NaN con la media de la columna "BLUSAS"  
training["BLUSAS"].fillna(training["BLUSAS"].mean(), inplace=True)
```

En la figura 17, se reemplaza la columna "BLUSAS" del DataFrame training con su media antes de realizar análisis o entrenar modelos, con el fin de evitar complicaciones durante el entrenamiento, ya que muchos algoritmos no pueden manejar de manera efectiva valores nulos.

Figura 18

Normalizar los datos

```
from sklearn.preprocessing import MinMaxScaler  
  
# Normalizar los datos  
scaler = MinMaxScaler()  
input_normalized = scaler.fit_transform(training[['BLUSAS']])
```

En la figura 18, el código crea un objeto **MinMaxScaler** y lo utiliza para normalizar los datos en la columna "BLUSAS" de training, asegurando que estos valores estén en una escala común entre 0 y 1. La normalización es útil en muchos casos, especialmente al trabajar con algoritmos de machine learning y redes neuronales, para evitar que ciertos atributos dominen sobre otros, debido a sus magnitudes originales.

El **MinMaxScaler** es una técnica de escalamiento que transforma los datos de manera que estén en un rango específico, que por defecto es entre 0 y 1. Este código importa la clase **MinMaxScaler** desde el módulo `sklearn.preprocessing` en la biblioteca `scikit-learn`.

Figura 19

Identificar el input y target de la Red Neuronal

```
# Identificar el conjuntos de entrada y salida
# Inicializar listas vacías para almacenar los conjuntos de entrada y salida
x_input = [] # Conjunto de entrada
y_target = [] # Conjunto de salida

# Identificamos a través de los datos normalizados
for i in range(len(input_normalizado) - 3):
    x_input.append(input_normalizado[i:i+3, 0])# Agregar secuencias de longitud 3 como entrada
    y_target.append(input_normalizado[i+3, 0])# Agregar el siguiente valor como salida

# Convertir las listas a arrays numpy
x_input = np.array(x_input)
y_target = np.array(y_target)

print(x_input)
print(y_target)
print(len(x_input))
print(len(y_target))
```

```
[[[0.03201302 0.26641346 0.61367336]
 [0.26641346 0.61367336 0.38632664]
 [0.61367336 0.38632664 0.24308193]]
```

Como se aprecia en la figura, se identifica los inputs y target. Este proceso se lleva a cabo con el objetivo de preparar datos secuenciales para entrenar un modelo capaz de predecir el siguiente valor en función de secuencias

Se procede a la inicialización de las listas `x_input` y `y_target` con el propósito de almacenar los conjuntos de entrada y salida, respectivamente.

Continuamente, se lleva a cabo una iteración a lo largo de los datos normalizados mediante un bucle `for` que abarca desde 0 hasta la longitud de los datos menos 3.

En cada ciclo de la iteración, se incorpora una secuencia de longitud 3 como conjunto de entrada (`x_input`), y el valor que sigue a dicha secuencia se designa como conjunto de salida (`y_target`).

Posteriormente, las listas son transformadas en arrays `numpy`, preparándolas para su implementación en modelos de aprendizaje automático.

Y al finalizar, se imprime la información completa, incluyendo los conjuntos de entrada y salida, junto con sus respectivas longitudes.

Figura 20

Normalizar la salida

```
03 ✓ ▶ # Normalizar las salidas (y_target)
y_target_normalized = scaler.transform(training[['BLUSAS']].iloc[3:])
y_target_normalized = y_target_normalized.flatten()
```

En la figura 19, el código se encarga de normalizar las salidas de la columna "BLUSAS" del DataFrame training y luego aplana el resultado para obtener un array unidimensional y_target_normalized. Este array representa las salidas normalizadas, listas para ser utilizadas en el entrenamiento de un modelo de predicción

Figura 21

Definición de la estructura de la Red Neuronal para su aprendizaje

```
▶ from keras.models import Sequential
from keras.layers import Dense
```

Luego, se importan las clases y funciones necesarias de Keras para construir una red neuronal.

Sequential, es un tipo de modelo que permite construir una red neuronal de manera secuencial, capa por capa, lo que es común en la mayoría de las redes neuronales.

keras.layers: Se importa dos clases clave de Keras para la construcción de capas en la red neuronal:

Dense: Esta clase se utiliza para crear capas densamente conectadas, también conocidas como capas completamente conectadas. En estas capas, cada neurona está conectada a todas las neuronas de la capa anterior.

Figura

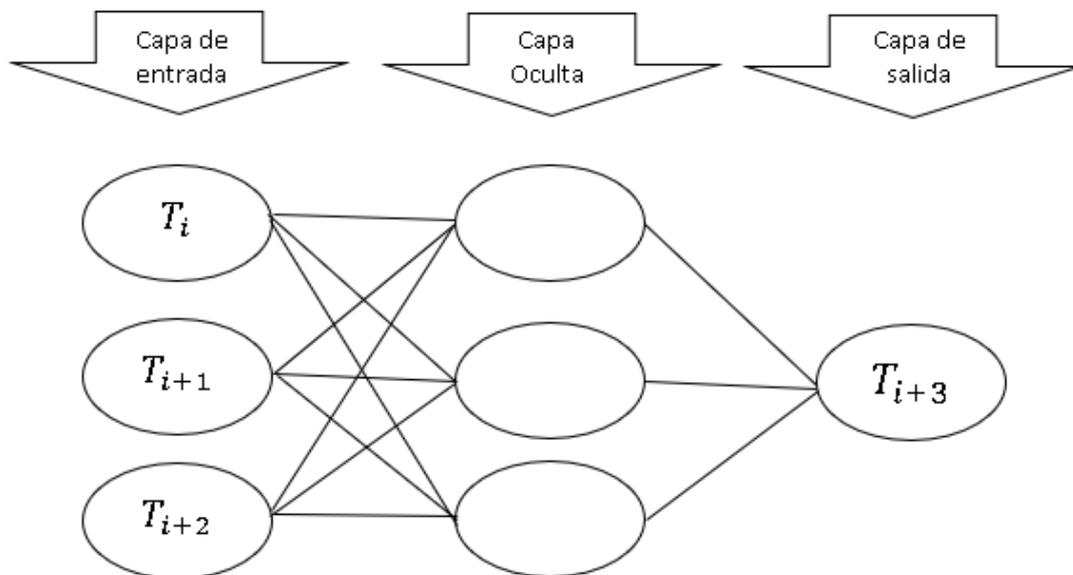
Implementación de las capas ocultas de la Red Neuronal Artificial

```
# Crear modelo
model = Sequential() # Crea un modelo de red neuronal secuencial.
model.add(Dense(16, input_dim=3, activation='relu')) # el modelo tien 16 neuronas con 3 entradas
model.add(Dense(16, activation='relu')) # la capa oculta tiene 16 neuronas en la capas ocultas
model.add(Dense(1, activation='linear')) # añade una salida y utiliza la activacion lineal
```

Se crea un modelo secuencial llamado `model`, las capas se apilan una encima de la otra en un orden secuencial (totalmente conectadas). La primera capa tiene 16 neuronas con una dimensión de entrada de 3 y una función de activación ReLU, la segunda capa también tiene 16 neuronas y de activación ReLU, y la tercera capa tiene 1 neurona (ya que se está realizando una regresión) con una función de activación lineal.

Figura 23

Estructura de la Red Neuronal Artificial



Como se aprecia en la figura 23, se modela la estructura de la red artificial

Figura 24

Configuración de los parámetros de la red

```
▶ #Configurar los parametros de la red
model.compile(loss='mean_squared_error', optimizer='adam', metrics='mae')
```

En el proceso de compilación del modelo, se utiliza la función `model.compile ()`. En este paso, se define la función de pérdida que medirá la discrepancia entre las predicciones del modelo y los valores reales objetivo. En este caso, se ha seleccionado la función de pérdida de error cuadrático medio. Además, se utiliza el optimizador "adam" y se mide la precisión utilizando la métrica `mean_absolute_percentage_error`, que evalúa el error porcentual absoluto medio entre las predicciones y los valores objetivos. El objetivo durante el entrenamiento es minimizar esta métrica para mejorar la precisión del modelo.

3.2.8.1. Fase de entrenamiento (Blusas)

Figura 25

Realizar el proceso de entrenamiento de la Red Neuronal

```
# Entrenar el modelo (BLUSAS)
history= model.fit(x_input, y_target_normalized, epochs=1000)
```

Finalmente, el modelo se entrena con los datos de entrada `x_input` y las etiquetas `y_target` durante 1000 épocas utilizando `model.fit ()`

Figura 26

Resultado de Entrenamiento

```
2/2 [=====] - 0s 7ms/step - loss: 0.0038 - mae: 0.0336
Epoch 994/1000
2/2 [=====] - 0s 7ms/step - loss: 0.0038 - mae: 0.0341
Epoch 995/1000
2/2 [=====] - 0s 8ms/step - loss: 0.0038 - mae: 0.0338
Epoch 996/1000
2/2 [=====] - 0s 9ms/step - loss: 0.0038 - mae: 0.0337
Epoch 997/1000
2/2 [=====] - 0s 7ms/step - loss: 0.0038 - mae: 0.0329
Epoch 998/1000
2/2 [=====] - 0s 7ms/step - loss: 0.0038 - mae: 0.0346
Epoch 999/1000
2/2 [=====] - 0s 9ms/step - loss: 0.0037 - mae: 0.0330
Epoch 1000/1000
2/2 [=====] - 0s 7ms/step - loss: 0.0039 - mae: 0.0351
```

Como se evidencia en la Figura 26, observamos una reducción gradual en nuestro error a lo largo del tiempo, alcanzando un error del 0.38% después de 1000 épocas. El proceso de aprendizaje concluyó debido a que se alcanzó el límite máximo de épocas.

El valor de pérdida Loss (0.0038) es una medida del error. Este valor debería ser lo más bajo posible, indicando que el modelo ha aprendido a hacer predicciones cercanas a los valores reales.

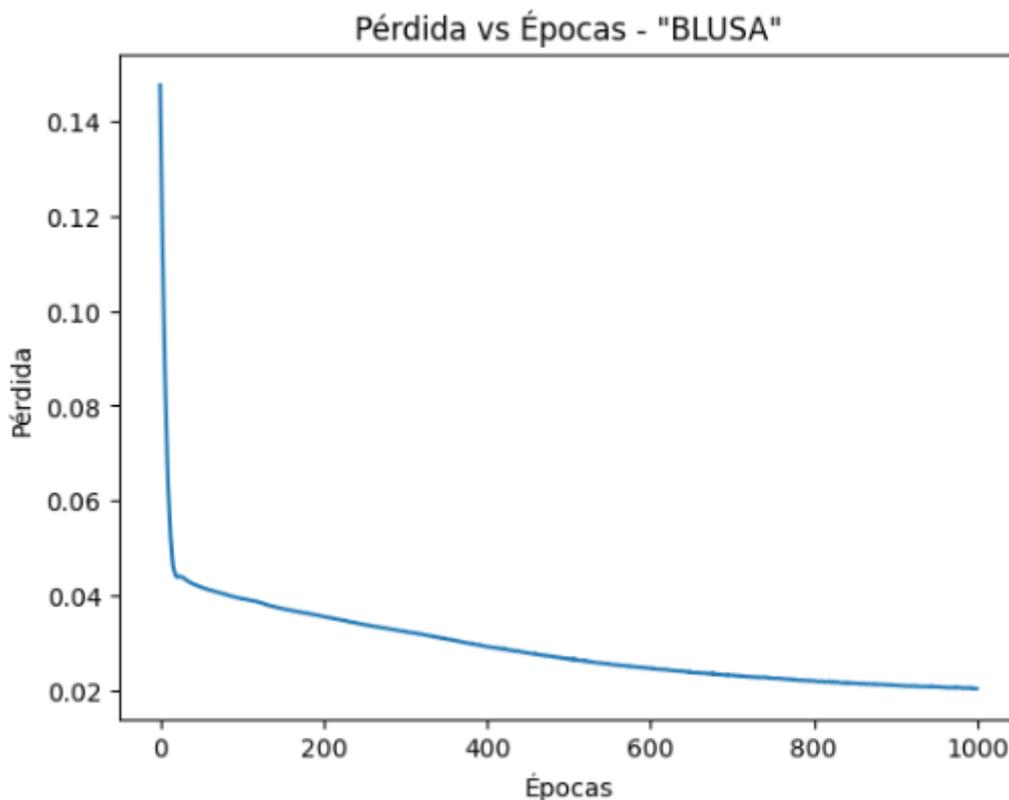
Además, el Mean Absolute Error (MAE) de 0.0351 representa la media de las diferencias absolutas entre las predicciones y los valores reales, siendo un indicador bajo que sugiere que las predicciones son cercanas a los valores reales en el conjunto de evaluación.

Interpretación específica:

En este contexto, parece que tu modelo ha aprendido bien a hacer predicciones en el conjunto de evaluación, ya que tanto la pérdida como el MAE son bajos. Sin embargo, estos resultados son específicos al conjunto de evaluación y no necesariamente indican cómo el modelo se desempeñará en datos que no ha visto antes (conjunto de prueba).

Figura 27

Gráfico de Perdida vs Épocas



En la Figura 27, se espera ver una disminución en la pérdida a medida que avanzan las épocas. Esto nos indica que el modelo está aprendiendo y mejorando su capacidad para hacer predicciones precisas.

Después de cierto punto, se observa que la pérdida deja de disminuir significativamente. Esto significa que el modelo ha alcanzado un punto donde ya no mejora de manera significativa.

Tabla 2

Entrenamiento de la Red Neuronal (blusas)

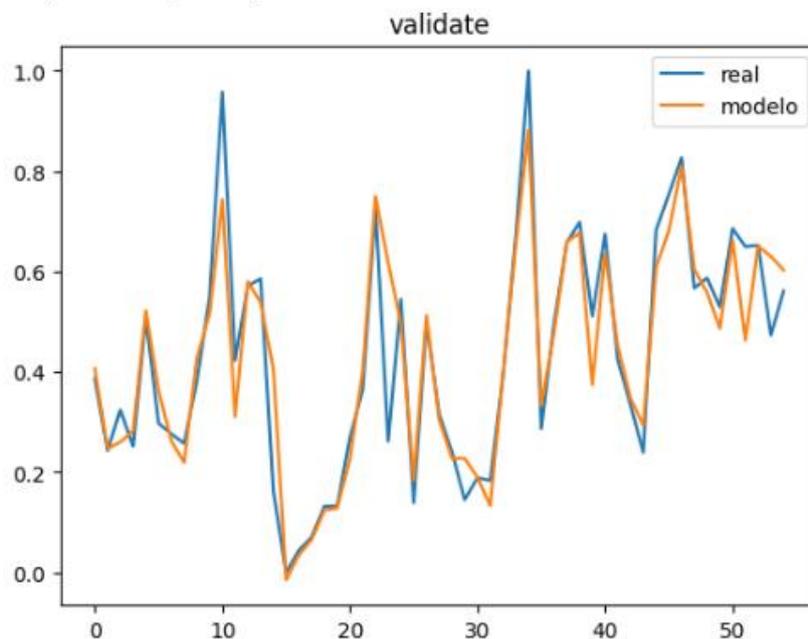
Venta Real	Ventas Desnormalizada	Pronóstico de la Red Neuronal	Error
750	0.386327	0.472557	0,086230
486	0.243082	0.481059	0.237977
635	0.323928	0.21471	0.109218
502	0.251763	0.299399	0.047636
974	0.507868	0.553388	0.04552
586	0.297341	0.508524	0.211182
547	0.27618	0.495359	0.219179
512	0.257189	0.212607	0.044582
742	0.381986	0.452538	0.070552
1050	0.549105	0.544882	0.004223
1804	0.95822	0.719275	0.238946
817	0.42268	0.446693	0.024012
1091	0.571351	0.518586	0.052765
1117	0.585458	0.57287	0.012589
336	0.161693	0.630612	0.468919
38	0	0.062683	0.062683
119	0.04395	0.05436	0.01041
168	0.070537	0.034981	0.035556
281	0.13185	0.091	0.04085
282	0.132393	0.149782	0.017389
534	0.269126	0.263209	0.005918
710	0.364623	0.375332	0.010709
1375	0.725448	0.588821	0.136626
520	0.26153	0.620195	0.358665
1042	0.544764	0.475208	0.069556
294	0.138904	0.271772	0.132868
958	0.499186	0.316436	0.18275
619	0.315247	0.188828	0.126419
481	0.240369	0.262149	0.02178
305	0.144872	0.208268	0.063396
386	0.188823	0.26727	0.078448
376	0.183397	0.192081	0.008685
758	0.390667	0.395861	0.005194
1272	0.66956	0.529124	0.140437
1881	1	0.668362	0.331638
566	0.286489	0.407994	0.121504
953	0.496473	0.462278	0.034195
1250	0.657623	0.605904	0.05172
1325	0.698318	0.645822	0.052496
980	0.511123	0.545348	0.034225
1281	0.674444	0.571574	0.10287

819	0.423766	0.51243	0.088665
653	0.333695	0.545629	0.211934
479	0.239284	0.324043	0.084759
1295	0.68204	0.418582	0.263458
1427	0.753663	0.635306	0.118357
1562	0.826913	0.598367	0.228545
1083	0.56701	0.409872	0.157138
1120	0.587086	0.572039	0.015047
1013	0.529029	0.518756	0.010272
1302	0.685838	0.603792	0.082046
1235	0.649485	0.528519	0.120966
1240	0.652198	0.554613	0.097584
910	0.473142	0.513577	0.040435
1072	0.561042	0.564709	0.003667

Se visualiza en la tabla 2 la cantidad de venta de blusas (real y des normalizada), se ve muy próximo a la cantidad que pronostica la red neuronal y consecuentemente el error disminuye significativamente

Figura 28

Gráfico de entrenamiento de la Red Neuronal (blusas)



Nota: En la figura 28 se visualiza la proximidad muy cercana de la red real, al modelo de la red neuronal

Validación con los datos de prueba

Para realizar la validación, separamos nuestros datos en dos, los datos de entrada en `x_input` y `y_target`, para validar la capacidad de la red neuronal los datos deben de estar normalizados.

Figura 29

Resultados con los Datos de Validación

```
[147] score = model.evaluate(x_input,y_target)
      print("\n %s: %.2f%%" % (model.metrics_names[0],score[0]*100))

2/2 [=====] - 0s 14ms/step - loss: 0.0154 - mae: 0.0816

      loss: 1.54%
```

Nombrando `score` emplearemos el método "evaluate", este método requiere dos parámetros de entrada: los datos de entrada almacenados en la variable "`x_input`" y los datos de salida en la variable "`y_target`" de nuestro conjunto de datos de prueba. Como se visualiza en la Figura 32 el error que tenemos para pronosticar con los datos de prueba es del 1.54%.

Tabla 3

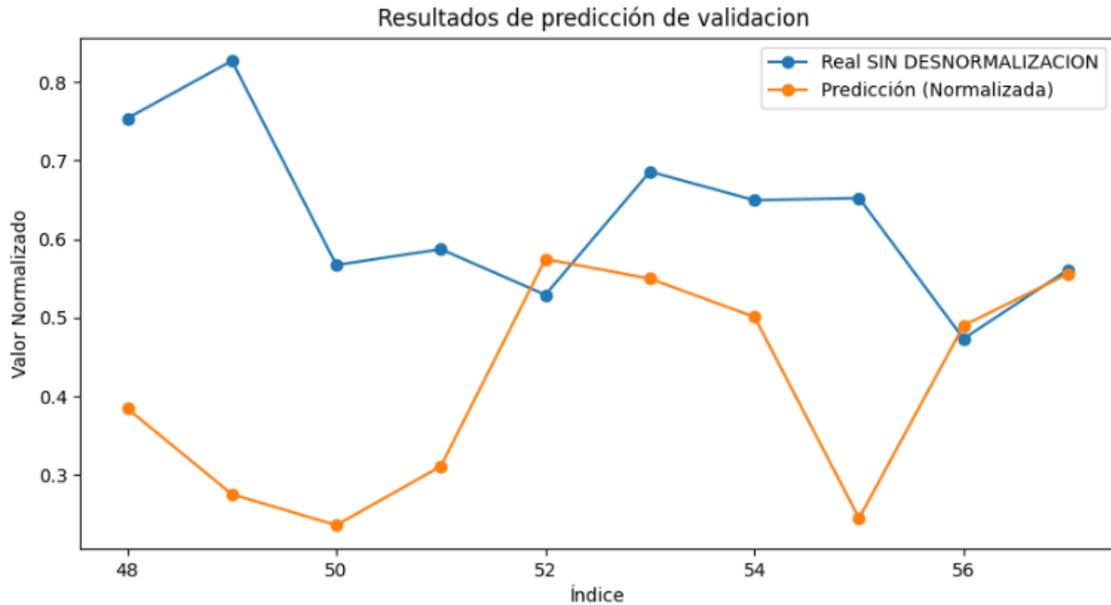
Datos de Validación (blusas)

Venta Real	Ventas Desnormalizada	Pronóstico de la Red Neuronal	Error
1562	0.826913	0.598367	0.228545
1083	0.56701	0.409872	0.157138
1120	0.587086	0.572039	0.015047
1013	0.529029	0.518756	0.010272
1302	0.685838	0.603792	0.082046
1235	0.649485	0.528519	0.120966
1240	0.652198	0.554613	0.097584
910	0.473142	0.513577	0.040435
1072	0.561042	0.564709	0.003667

En la Tabla 3, realizamos una partición de los 10 últimos meses de venta de blusas (real y des normalizada), se ve muy próximo a la cantidad que pronostica la red neuronal y consecuentemente el error disminuye significativamente

Figura 30

Gráfico de los Datos del Entrenamiento de la Red Neuronal (blusas)



Nota: En la figura 30 se visualiza la proximidad muy cercana de la red real, al modelo de la red neuronal de los datos de validación.

Pronóstico

Figura 31

Pronóstico de la prenda blusa

```
✓ [18] respuesta = model.predict([[1240, 910, 1072]])
0s print(respuesta.round())

1/1 [=====] - 0s 47ms/step
[[1000.]]
```

(Ver Figura 31) Al pronosticar utilizaremos el método llamado “**model.predict**”, se utiliza para obtener las predicciones del modelo, en el cual, como datos de entrada, enviaremos como parámetros los tres últimos meses, es decir, de abril, junio y julio del 2023, se está pasando una secuencia de tres valores, que parece coincidir con el patrón de entrada que el modelo espera, para poder predecir las ventas que podrían generar en el mes de agosto. El cual nos da como resultado redondeado “**respuesta.round()**” ya des estandarización 1000 blusas vendidas para el mes de agosto 2023.

3.2.8.2. Fase de entrenamiento (Cartera)

Figura 32

Realizar el proceso de entrenamiento de la Red Neuronal

```
# Entrenar el modelo (CARTERA)
history= model.fit(x_input, y_target_normalized, epochs=1000)
```

El modelo se entrena con los datos de entrada **x_input** y las etiquetas **y_target** durante 1000 épocas utilizando **model.fit ()**

Figura 33

Resultado de Entrenamiento

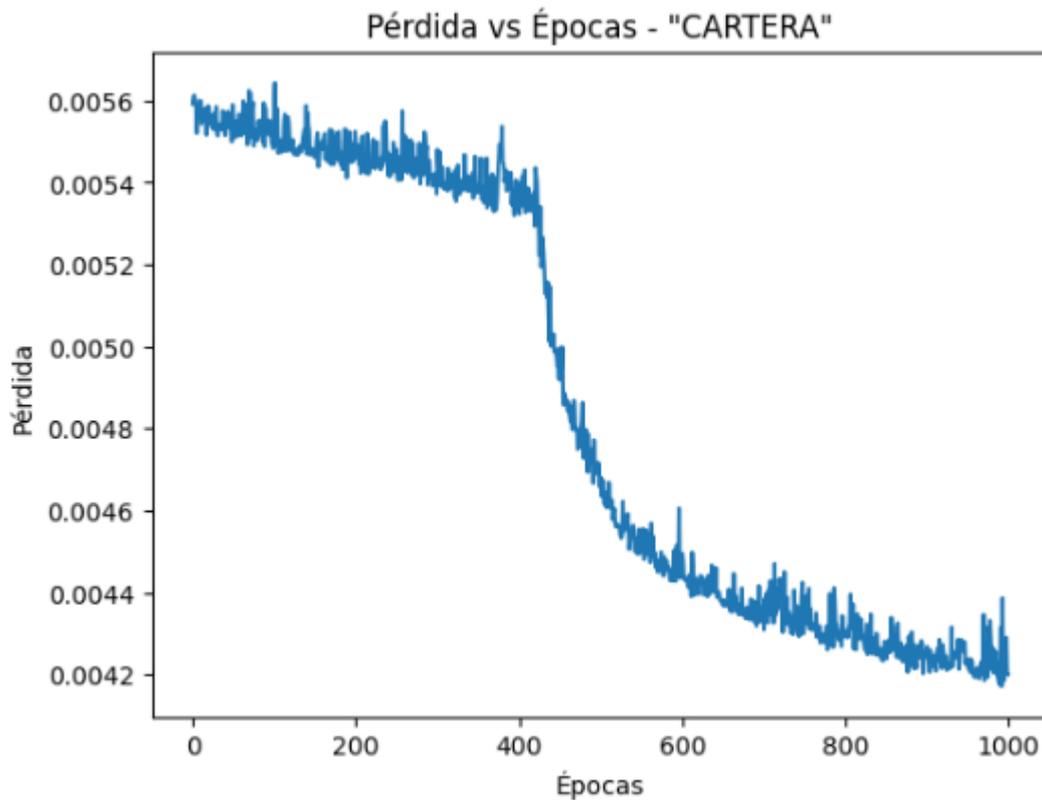
```
2/2 [=====] - 0s 6ms/step - loss: 0.0042 - mae: 0.0468
[57] Epoch 991/1000
2/2 [=====] - 0s 8ms/step - loss: 0.0043 - mae: 0.0469
Epoch 992/1000
2/2 [=====] - 0s 7ms/step - loss: 0.0042 - mae: 0.0460
Epoch 993/1000
2/2 [=====] - 0s 12ms/step - loss: 0.0044 - mae: 0.0473
Epoch 994/1000
2/2 [=====] - 0s 8ms/step - loss: 0.0042 - mae: 0.0457
Epoch 995/1000
2/2 [=====] - 0s 8ms/step - loss: 0.0042 - mae: 0.0458
Epoch 996/1000
2/2 [=====] - 0s 11ms/step - loss: 0.0043 - mae: 0.0471
Epoch 997/1000
2/2 [=====] - 0s 10ms/step - loss: 0.0043 - mae: 0.0471
Epoch 998/1000
2/2 [=====] - 0s 12ms/step - loss: 0.0043 - mae: 0.0473
Epoch 999/1000
2/2 [=====] - 0s 7ms/step - loss: 0.0042 - mae: 0.0457
Epoch 1000/1000
2/2 [=====] - 0s 13ms/step - loss: 0.0042 - mae: 0.0455
```

Como se visualiza en la Figura 33, observamos una reducción en nuestro error a lo largo del tiempo, alcanzando un error del 0.42%, debido a que se alcanzó el límite máximo de 1000 épocas.

El valor de pérdida (Loss) de 0.0042 indica el error entre las predicciones del modelo y los valores reales en el conjunto de evaluación, siendo más bajo deseable. Además, el Mean Absolute Error (MAE) de 0.455 representa la media de las diferencias absolutas entre las predicciones y los valores reales, siendo un indicador bajo que sugiere que las predicciones son cercanas a los valores reales en el conjunto de evaluación.

Figura 34

Gráfico de Pérdida vs Épocas



En la figura 34, se espera ver una disminución en la pérdida a medida que avanzan las épocas. Esto nos indica que el modelo está aprendiendo y mejorando su capacidad para hacer predicciones precisas. Después de cierto punto, se observa que la pérdida deja de disminuir significativamente. Esto significa que el modelo ha alcanzado un punto donde ya no mejora de manera significativa.

Tabla 4

Datos del Entrenamiento de la Red Neuronal (cartera)

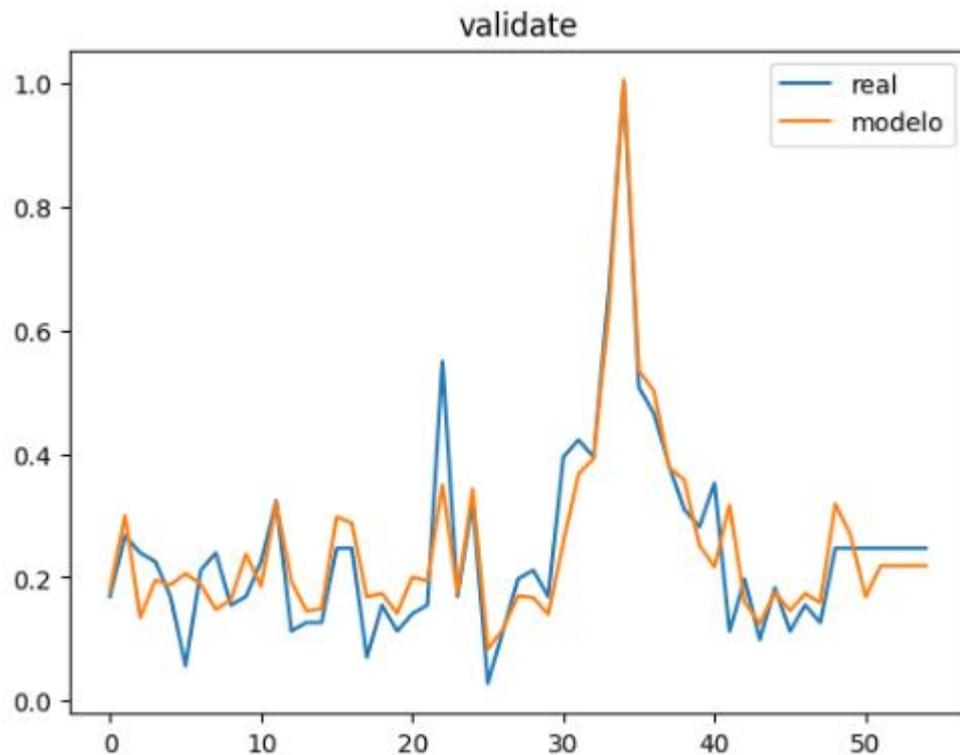
Venta Real	Ventas Desnormalizada	Pronóstico de la Red Neuronal	Error
14	0.169014	0.179826	0.010812
21	0.267606	0.300123	0.032518
19	0.239437	0.134965	0.104472
18	0.225352	0.194998	0.030355
14	0.169014	0.187776	0.018762
6	0.056338	0.205836	0.149498
17	0.211268	0.188918	0.02235
19	0.239437	0.147564	0.091873
13	0.15493	0.16392	0.00899
14	0.169014	0.237083	0.068069

18	0.225352	0.185985	0.039367
25	0.323944	0.320352	0.003592
10	0.112676	0.193637	0.080961
11	0.126761	0.145283	0.018522
11	0.126761	0.14903	0.02227
19.6	0.247197	0.297876	0.050679
19.6	0.247197	0.287309	0.040112
7	0.070423	0.168057	0.097634
13	0.15493	0.173904	0.018974
10	0.112676	0.141567	0.028891
12	0.140845	0.200225	0.05938
13	0.15493	0.194186	0.039256
41	0.549296	0.3487	0.200595
14	0.169014	0.173811	0.004797
25	0.323944	0.343077	0.019134
4	0.028169	0.083613	0.055444
10	0.112676	0.115746	0.00307
16	0.197183	0.169966	0.027218
17	0.211268	0.167041	0.044227
14	0.169014	0.140157	0.028857
30	0.394366	0.256302	0.138065
32	0.422535	0.366885	0.055651
30	0.394366	0.392456	0.00191
49	0.661972	0.627467	0.034505
73	1	1.004536	0.004536
38	0.507042	0.534423	0.027381
35	0.464789	0.501309	0.036521
29	0.380282	0.377192	0.003089
24	0.309859	0.358061	0.048202
22	0.28169	0.252147	0.029543
27	0.352113	0.216978	0.135135
10	0.112676	0.316443	0.203767
16	0.197183	0.159718	0.037465
9	0.098592	0.125036	0.026445
15	0.183099	0.175647	0.007452
10	0.112676	0.145944	0.033268
13	0.15493	0.173321	0.018391
11	0.126761	0.15914	0.032379
19.6	0.247197	0.318682	0.071484
19.6	0.247197	0.268592	0.021395
19.6	0.247197	0.168057	0.079141
19.6	0.247197	0.218969	0.028228
19.6	0.247197	0.218969	0.028228
19.6	0.247197	0.218969	0.028228
19.6	0.247197	0.218969	0.028228

Se visualiza en la Tabla 4 la cantidad de venta de cartera (real y des normalizada), se ve muy próximo a la cantidad que pronostica la red neuronal y consecuentemente el error disminuye significativamente

Figura 35

Gráfico de Entrenamiento de la Red Neuronal (Cartera)



Nota: En la figura 35 se visualiza la proximidad muy cercana de la red real, al modelo de la red neuronal

Validación con los datos de Prueba

Para realizar la validación, separamos nuestros datos en dos, los datos de entrada en `x_input` y `y_target`, para validar la capacidad de la red neuronal los datos deben de estar normalizados.

Figura 36

Resultados con los Datos de Validación

```
0s ✓ ▶ score = model.evaluate(x_input,y_target)
print("\n %s: %.2f%%" % (model.metrics_names[0],score[0]*100))

2/2 [=====] - 0s 8ms/step - loss: 0.0042 - mae: 0.0469
loss: 0.42%
```

Como se visualiza en la Figura 36, nombrando `score` emplearemos el método "evaluate", Este método requiere dos parámetros de entrada: los datos de entrada almacenados en la variable "`x_input`" y los datos de

salida en la variable "y_target" de nuestro conjunto de datos de prueba el error que tenemos para pronosticar con los datos de prueba es del 0.42%.

Tabla 5

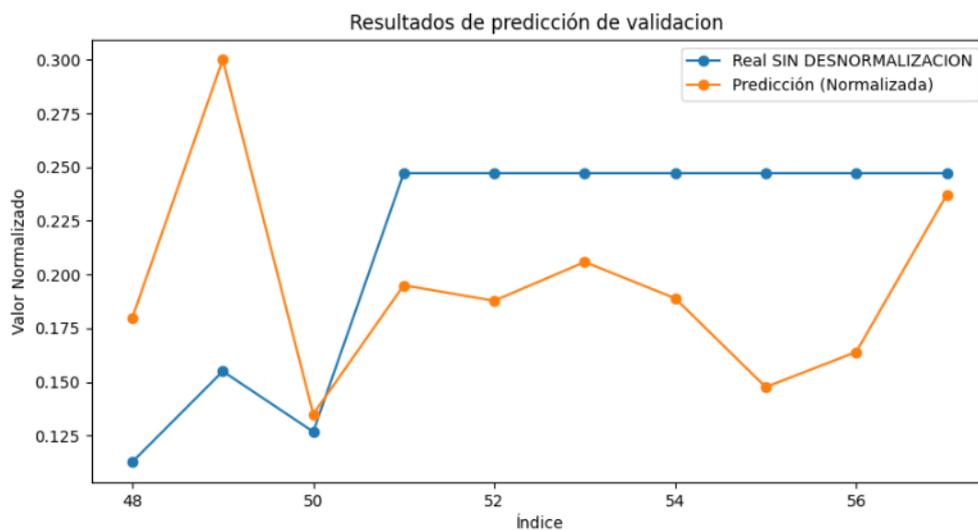
Datos de Validación (cartera)

Venta Real	Ventas Desnormalizada	Pronóstico de la Red Neuronal	Error
10	0.112676	0.145944	0.033268
13	0.15493	0.173321	0.018391
11	0.126761	0.15914	0.032379
19.6	0.247197	0.318682	0.071484
19.6	0.247197	0.268592	0.021395
19.6	0.247197	0.168057	0.079141
19.6	0.247197	0.218969	0.028228
19.6	0.247197	0.218969	0.028228
19.6	0.247197	0.218969	0.028228
19.6	0.247197	0.218969	0.028228

En la Tabla 5, realizamos una partición de los 10 últimos meses de venta de cartera (real y des normalizada), se ve muy próximo a la cantidad que pronostica la red neuronal y consecuentemente el error disminuye significativamente

Figura 37

Gráfico de los Datos del Entrenamiento de la Red Neuronal (cartera)



Nota: En la figura 37 se visualiza la proximidad muy cercana de la red real, al modelo de la red neuronal de los datos de validación.

Pronostico

Figura 38

Pronóstico de la prenda Cartera

```
✓ 0s ▶ respuesta = model.predict([[19.6, 19.6, 19.6]])
    print(respuesta.round())

1/1 [=====] - 0s 80ms/step
[[5.]]
```

(Ver Figura 38) Al pronosticar utilizaremos el método llamado “**model.predict**”, se utiliza para obtener las predicciones del modelo, en el cual, como datos de entrada, enviaremos como parámetros los tres últimos meses, es decir, de abril, junio y julio del 2023, se está pasando una secuencia de tres valores, que parece coincidir con el patrón de entrada que el modelo espera, para poder predecir las ventas que podrían generar en el mes de agosto. El cual nos da como resultado redondeado “**respuesta.round()**” ya des estandarización, 5 carteras vendidas para el mes de agosto 2023.

3.2.8.3. Fase de entrenamiento (Casaca)

Figura 39

Realizar el proceso de entrenamiento de la Red Neuronal

```
✓ 54s [23] # Entrenar el modelo (CASACA)
        history= model.fit(x_input, y_target_normalized, epochs=1500)
```

El modelo se entrena con los datos de entrada **x_input** y las etiquetas **y_target** durante 1000 épocas utilizando **model.fit ()**

Figura 40

Resultado de Entrenamiento

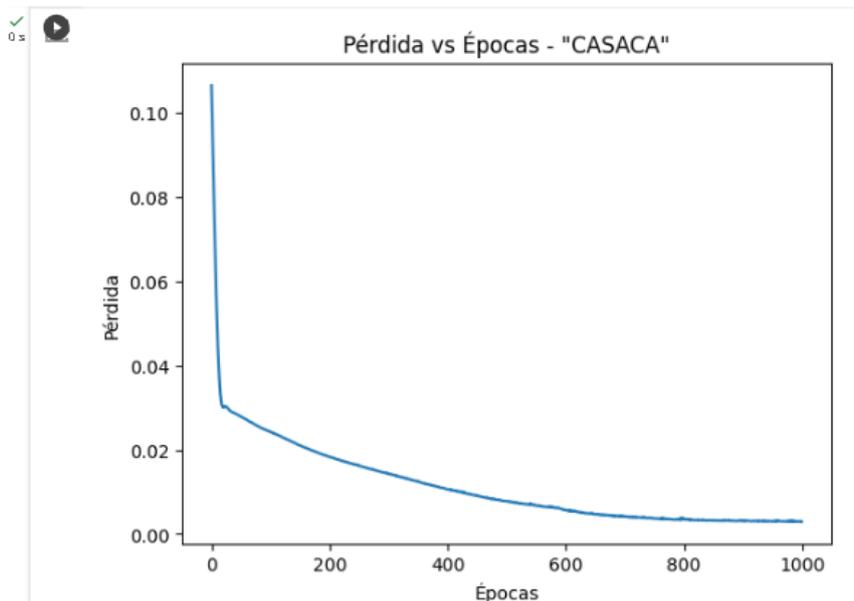
```
✓ [33] Epoch 994/1000  
2/2 [=====] - 0s 8ms/step - loss: 0.0014 - mae: 0.0185  
Epoch 995/1000  
2/2 [=====] - 0s 10ms/step - loss: 0.0014 - mae: 0.0165  
Epoch 996/1000  
2/2 [=====] - 0s 8ms/step - loss: 0.0014 - mae: 0.0162  
Epoch 997/1000  
2/2 [=====] - 0s 8ms/step - loss: 0.0014 - mae: 0.0182  
Epoch 998/1000  
2/2 [=====] - 0s 8ms/step - loss: 0.0014 - mae: 0.0199  
Epoch 999/1000  
2/2 [=====] - 0s 9ms/step - loss: 0.0014 - mae: 0.0149  
Epoch 1000/1000  
2/2 [=====] - 0s 8ms/step - loss: 0.0015 - mae: 0.0167
```

Como se visualiza en la Figura 40, observamos una reducción en nuestro error a lo largo del tiempo, alcanzando un error del 0.15%, debido a que se alcanzó el límite máximo de 1000 épocas.

El valor de pérdida (Loss) de 0.0015 indica el error entre las predicciones del modelo y los valores reales en el conjunto de evaluación, siendo más bajo deseable. Además, el Mean Absolute Error (MAE) de 0.0167 representa la media de las diferencias absolutas entre las predicciones y los valores reales, siendo un indicador bajo que sugiere que las predicciones son cercanas a los valores reales en el conjunto de evaluación.

Figura 41

Gráfico de Pérdida vs Épocas



En la figura 41, se espera ver una disminución en la pérdida a medida que avanzan las épocas. Esto nos indica que el modelo está aprendiendo y mejorando su capacidad para hacer predicciones precisas. Después de cierto punto, se observa que la pérdida deja de disminuir significativamente. Esto significa que el modelo ha alcanzado un punto donde ya no mejora de manera significativa.

Tabla 6

Datos del entrenamiento de la Red Neuronal (casaca)

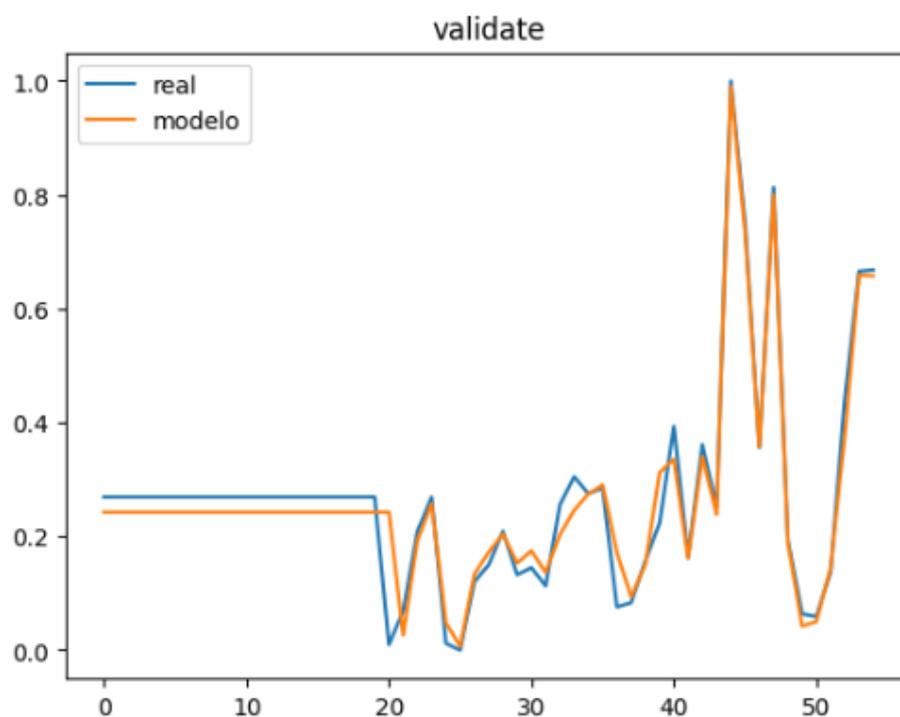
Venta Real	Ventas Desnormalizada	Pronóstico de la Red Neuronal	Error
133.529412	0.269114	0.251447	0.017667
133.529412	0.269114	0.251447	0.017667
133.529412	0.269114	0.251447	0.017667
133.529412	0.269114	0.251447	0.017667
133.529412	0.269114	0.251447	0.017667
133.529412	0.269114	0.251447	0.017667
133.529412	0.269114	0.251447	0.017667
133.529412	0.269114	0.251447	0.017667
133.529412	0.269114	0.251447	0.017667
133.529412	0.269114	0.251447	0.017667
133.529412	0.269114	0.251447	0.017667
133.529412	0.269114	0.251447	0.017667
133.529412	0.269114	0.251447	0.017667
133.529412	0.269114	0.251447	0.017667
133.529412	0.269114	0.251447	0.017667
133.529412	0.269114	0.251447	0.017667
133.529412	0.269114	0.251447	0.017667
133.529412	0.269114	0.251447	0.017667
28	0.009828	0.251447	0.241619
52	0.068796	0.055078	0.013718
109	0.208845	0.207971	0.000874
133.529412	0.269114	0.266755	0.002359
29	0.012285	0.011559	0.000726
24	0	-0.002598	0.002598
73	0.120393	0.128676	0.008283
85	0.149877	0.156968	0.007091
109	0.208845	0.212806	0.003961
78	0.132678	0.143499	0.010821
83	0.144963	0.15083	0.005867
70	0.113022	0.118703	0.005681
128	0.255528	0.235266	0.020263
148	0.304668	0.283928	0.02074

136	0.275184	0.28202	0.006835
140	0.285012	0.28753	0.002518
55	0.076167	0.115662	0.039495
58	0.083538	0.090489	0.006951
88	0.157248	0.154545	0.002703
115	0.223587	0.302639	0.079052
184	0.39312	0.387285	0.005835
91	0.164619	0.167617	0.002998
171	0.361179	0.355381	0.005798
126	0.250614	0.245133	0.005481
431	1	1.006389	0.006389
329	0.749386	0.749565	0.00018
169	0.356265	0.358738	0.002472
355	0.813268	0.812811	0.000457
103	0.194103	0.193621	0.000482
50	0.063882	0.059857	0.004025
48	0.058968	0.058615	0.000353
80	0.137592	0.130468	0.007124
204	0.44226	0.401218	0.041042
295	0.665848	0.666169	0.000321
296	0.668305	0.670275	0.00197

Se visualiza en la Tabla 6 la cantidad de venta de casaca (real y des normalizada), se ve muy próximo a la cantidad que pronostica la red neuronal y consecuentemente el error disminuye significativamente

Figura 42

Gráfico de Entrenamiento de la Red Neuronal (casaca)



Nota: En la figura 42 se visualiza la proximidad muy cercana de la red real, al modelo de la red neuronal

Validación con los datos de Prueba

Para realizar la validación, separamos nuestros datos en dos, los datos de entrada en `x_input` y `y_target`, para validar la capacidad de la red neuronal los datos deben de estar normalizados.

Figura 43

Resultados con los Datos de Validación

```

0s ✓ ▶ score = model.evaluate(x_input,y_target)
print("\n %s: %.2f%%" % (model.metrics_names[0],score[0]*100))

2/2 [=====] - 0s 9ms/step - loss: 0.0014 - mae: 0.0167

loss: 0.14%

```

Como se visualiza en la Figura 43, nombrando `score` emplearemos el método "evaluate", Este método requiere dos parámetros de entrada: los datos de entrada almacenados en la variable "`x_input`" y los datos de salida en la variable "`y_target`" de nuestro conjunto de datos de prueba el error que tenemos para pronosticar con los datos de prueba es del 0.14%.

Tabla 7

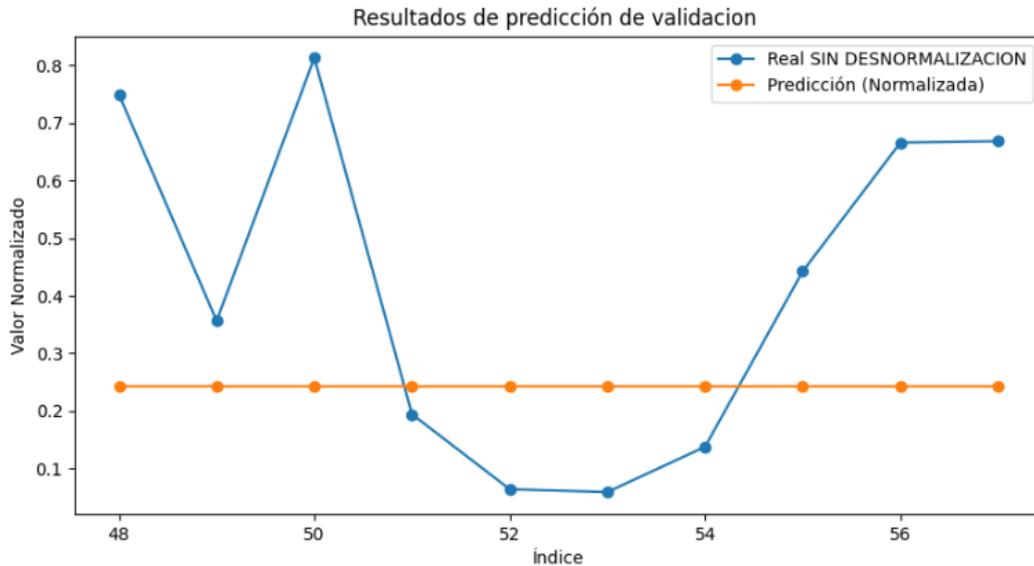
Datos de Validación (casaca)

Venta Real	Ventas Desnormalizada	Pronóstico de la Red Neuronal	Error
329	0.749386	0.749565	0.00018
169	0.356265	0.358738	0.002472
355	0.813268	0.812811	0.000457
103	0.194103	0.193621	0.000482
50	0.063882	0.059857	0.004025
48	0.058968	0.058615	0.000353
80	0.137592	0.130468	0.007124
204	0.44226	0.401218	0.041042
295	0.665848	0.666169	0.000321
296	0.668305	0.670275	0.00197

En la Tabla 7, realizamos una partición de los 10 últimos meses de venta de casaca (real y des normalizada), se ve muy próximo a la cantidad que pronostica la red neuronal y consecuentemente el error disminuye significativamente

Figura 44

Gráfico de los Datos del Entrenamiento de la Red Neuronal (Casaca)



Nota: En la figura 44 se visualiza la proximidad muy cercana de la red real, al modelo de la red neuronal de los datos de validación.

Pronóstico

Figura 45

Pronóstico de la prenda casaca

```
✓ [32] respuesta = model.predict([[204, 295, 296]])  
      print(respuesta.round())  
  
1/1 [=====] - 0s 38ms/step  
[[129.]]
```

(Ver Figura 45) Al pronosticar utilizaremos el método llamado “**model.predict**”, se utiliza para obtener las predicciones del modelo, en el cual, como datos de entrada, enviaremos como parámetros los tres últimos meses, es decir, de abril, junio y julio del 2023, se está pasando una secuencia de tres valores, que parece coincidir con el patrón de entrada que el modelo espera, para poder predecir las ventas que podrían generar en el mes de agosto. El cual nos da como resultado redondeado “**respuesta.round()**” ya des estandarización, 129 casacas vendidas para el mes de agosto 2023.

3.2.8.4. Fase de entrenamiento (Chompas)

Figura 46

Realizar el proceso de entrenamiento de la Red Neuronal

```
30 s ✓ # Entrenar el modelo (CHOMPAS)
      history= model.fit(x_input, y_target_normalized, epochs=1000)
```

El modelo se entrena con los datos de entrada **x_input** y las etiquetas **y_target** durante 1000 épocas utilizando **model.fit ()**

Figura 47

Resultado de Entrenamiento

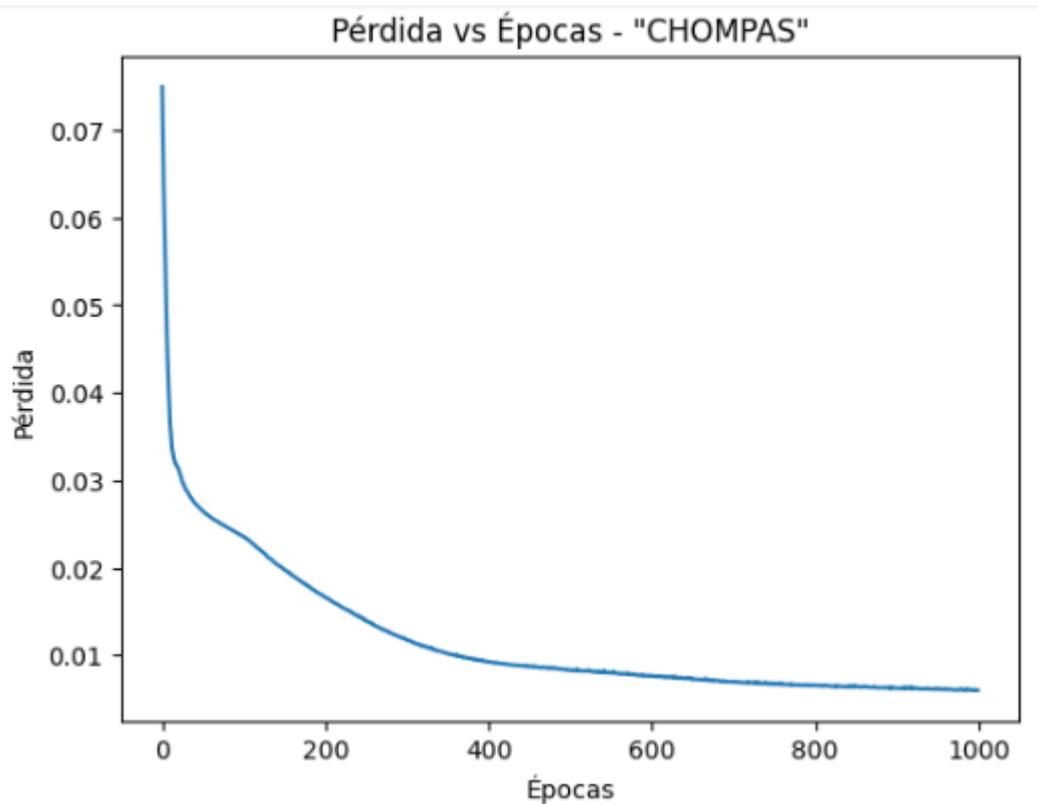
```
30 s ✓ 2/2 [=====] - 0s 8ms/step - loss: 0.0060 - mae: 0.0485
      Epoch 996/1000
      2/2 [=====] - 0s 7ms/step - loss: 0.0061 - mae: 0.0474
      Epoch 997/1000
      2/2 [=====] - 0s 8ms/step - loss: 0.0060 - mae: 0.0470
      Epoch 998/1000
      2/2 [=====] - 0s 7ms/step - loss: 0.0060 - mae: 0.0481
      Epoch 999/1000
      2/2 [=====] - 0s 7ms/step - loss: 0.0060 - mae: 0.0488
      Epoch 1000/1000
      2/2 [=====] - 0s 7ms/step - loss: 0.0060 - mae: 0.0487
```

Como se evidencia en la Figura 47, el valor de pérdida (Loss) de 0.0060 indica el error entre las predicciones del modelo y los valores reales en el conjunto de evaluación, siendo más bajo deseable. Además, el Mean Absolute Error (MAE) de 0.0487 representa la media de las diferencias absolutas entre las predicciones y los valores reales, siendo un indicador bajo que sugiere que las predicciones son cercanas a los valores reales en el conjunto de evaluación.

El modelo ha aprendido bien a hacer predicciones en el conjunto de evaluación, ya que tanto la pérdida como el MAE son bajos. Sin embargo, estos resultados son específicos al conjunto de evaluación y no necesariamente indican cómo el modelo se desempeñará en datos que no ha visto antes (conjunto de prueba).

Figura 48

Gráfico de Pérdida vs Épocas



En la figura 48, se espera ver una disminución en la pérdida a medida que avanzan las épocas. Esto nos indica que el modelo está aprendiendo y mejorando su capacidad para hacer predicciones precisas. Después de cierto punto, se observa que la pérdida deja de disminuir significativamente. Esto significa que el modelo ha alcanzado un punto donde ya no mejora de manera significativa.

Tabla 8

Datos del entrenamiento de la Red Neuronal (chompas)

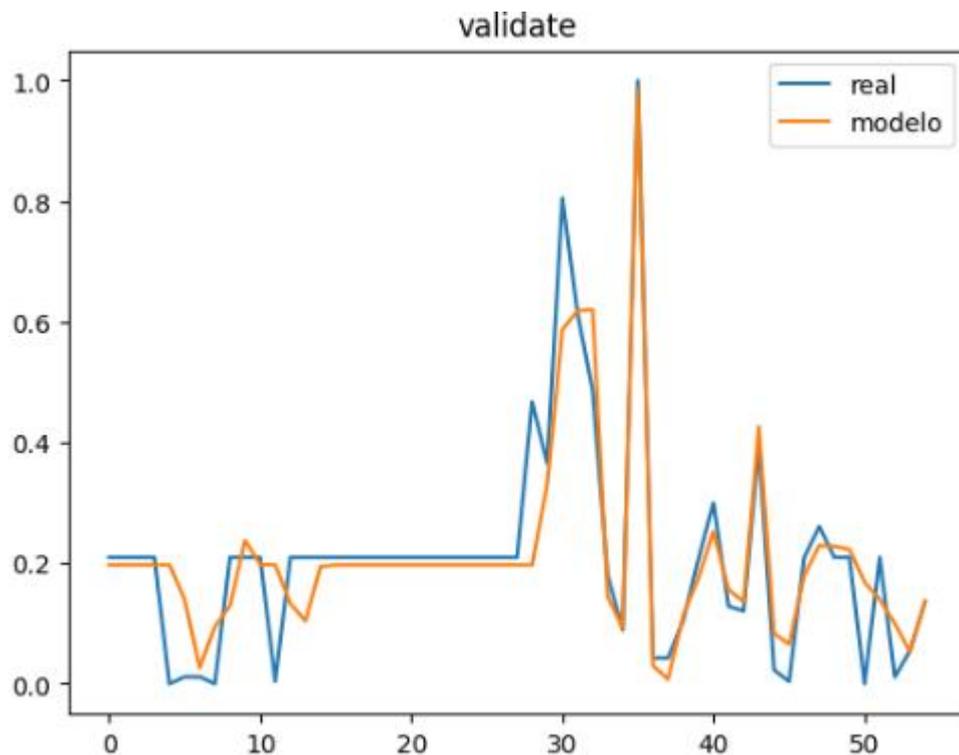
Venta Real	Ventas Desnormalizada	Pronóstico de la Red Neuronal	Error
54.9	0.209561	0.197253	0.012308
54.9	0.209561	0.197253	0.012308
54.9	0.209561	0.197253	0.012308
54.9	0.209561	0.197253	0.012308
1.0	0	0.197253	0.197253
4.0	0.011673	0.140054	0.128381
4.0	0.011673	0.026354	0.014681
1.0	0	0.094684	0.094684

54.9	0.209561	0.129543	0.080018
54.9	0.209561	0.237523	0.027962
54.9	0.209561	0.198052	0.011509
2.0	0.003891	0.197253	0.193362
54.9	0.209561	0.133177	0.076384
54.9	0.209561	0.104149	0.105412
54.9	0.209561	0.194371	0.01519
54.9	0.209561	0.197253	0.012308
54.9	0.209561	0.197253	0.012308
54.9	0.209561	0.197253	0.012308
54.9	0.209561	0.197253	0.012308
54.9	0.209561	0.197253	0.012308
54.9	0.209561	0.197253	0.012308
54.9	0.209561	0.197253	0.012308
54.9	0.209561	0.197253	0.012308
54.9	0.209561	0.197253	0.012308
54.9	0.209561	0.197253	0.012308
54.9	0.209561	0.197253	0.012308
54.9	0.209561	0.197253	0.012308
54.9	0.209561	0.197253	0.012308
54.9	0.209561	0.197253	0.012308
121.0	0.466926	0.197253	0.269673
95.0	0.365759	0.330922	0.034837
208.0	0.805447	0.586912	0.218536
159.0	0.614786	0.6186	0.003814
126.0	0.486381	0.62027	0.133888
47.0	0.178988	0.143891	0.035097
24.0	0.089494	0.092382	0.002888
258.0	1	0.984655	0.015345
12.0	0.042802	0.03016	0.012642
12.0	0.042802	0.007549	0.035252
28.0	0.105058	0.113871	0.008812
54.0	0.206226	0.175215	0.031011
78.0	0.299611	0.252179	0.047432
34.0	0.128405	0.155794	0.027389
32.0	0.120623	0.136021	0.015399
103.0	0.396887	0.426509	0.029622
7.0	0.023346	0.083944	0.060598
2.0	0.003891	0.065809	0.061918
54.9	0.209561	0.179418	0.030143
68.0	0.2607	0.229373	0.031327
54.9	0.209561	0.228063	0.018502
54.9	0.209561	0.222922	0.013361
1.0	0	0.168501	0.168501
54.9	0.209561	0.140054	0.069507
4.0	0.011673	0.099643	0.087969
15.0	0.054475	0.053643	0.000832
36.0	0.136187	0.137235	0.001048

Se visualiza en la Tabla 8 la cantidad de venta de casaca (real y des normalizada), se ve muy próximo a la cantidad que pronostica la red neuronal y consecuentemente el error disminuye significativamente

Figura 49

Gráfico de Entrenamiento de la Red Neuronal (chompas)



Nota: En la figura 49 se visualiza la proximidad muy cercana de la red real, al modelo de la red neuronal

Validación con los datos de Prueba

Para realizar la validación, separamos nuestros datos en dos, los datos de entrada en `x_input` y `y_target`, para validar la capacidad de la red neuronal los datos deben de estar normalizados.

Figura 50

Resultados con los Datos de Validación

```
score = model.evaluate(x_input,y_target)
print("\n %s: %.2f%%" % (model.metrics_names[0],score[0]*100))

2/2 [=====] - 0s 7ms/step - loss: 0.0060 - mae: 0.0476
loss: 0.60%
```

Como se visualiza en la Figura 50, nombrando score emplearemos el método "evaluate", Este método requiere dos parámetros de entrada: los datos de entrada almacenados en la variable "x_input" y los datos de salida en la variable "y_target" de nuestro conjunto de datos de prueba el error que tenemos para pronosticar con los datos de prueba es del 0.60%.

Tabla 9

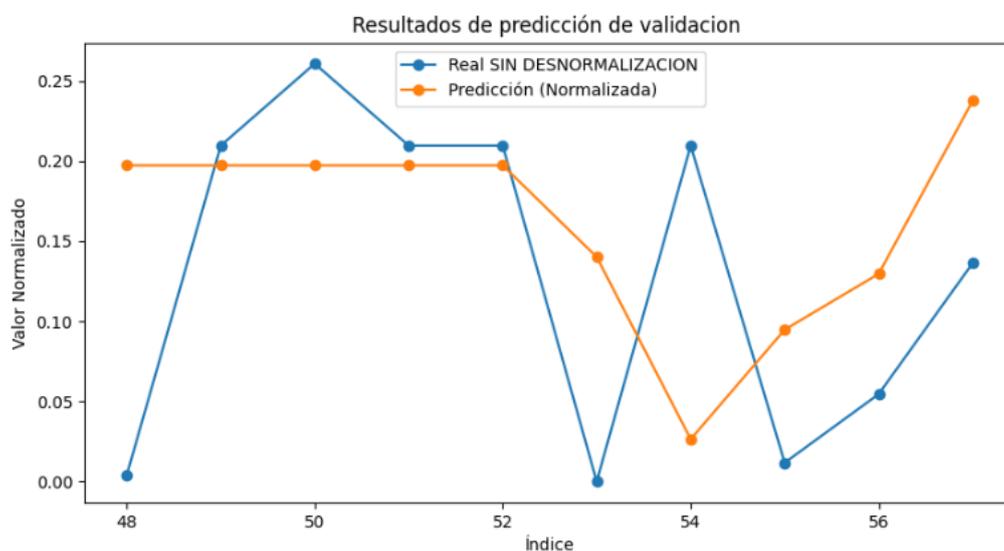
Datos de Validación (chompas)

Venta Real	Ventas Desnormalizada	Pronóstico de la Red Neuronal	Error
2.0	0.003891	0.065809	0.061918
54.9	0.209561	0.179418	0.030143
68.0	0.2607	0.229373	0.031327
54.9	0.209561	0.228063	0.018502
54.9	0.209561	0.222922	0.013361
1.0	0	0.168501	0.168501
54.9	0.209561	0.140054	0.069507
4.0	0.011673	0.099643	0.087969
15.0	0.054475	0.053643	0.000832
36.0	0.136187	0.137235	0.001048

En la Tabla 9, realizamos una partición de los 10 últimos meses de venta de chompa (real y des normalizada), se ve muy próximo a la cantidad que pronostica la red neuronal y consecuentemente el error disminuye significativamente

Figura 51

Gráfico de los Datos del Entrenamiento de la Red Neuronal (chompas)

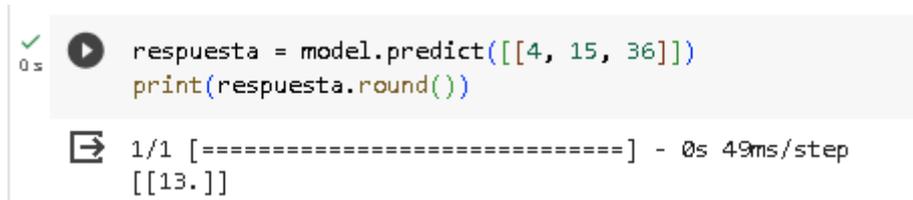


Nota: En la figura 51 se visualiza la proximidad muy cercana de la red real, al modelo de la red neuronal de los datos de validación.

Pronóstico

Figura 52

Pronóstico de la prenda Chompa



```
✓ 0s ▶ respuesta = model.predict([[4, 15, 36]])
    print(respuesta.round())

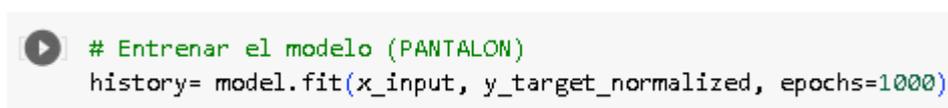
👉 1/1 [=====] - 0s 49ms/step
    [[13.]]
```

(Ver Figura 52) Al pronosticar utilizaremos el método llamado “**model.predict**”, se utiliza para obtener las predicciones del modelo, en el cual, como datos de entrada, enviaremos como parámetros los tres últimos meses, es decir, de abril, junio y julio del 2023, se está pasando una secuencia de tres valores, que parece coincidir con el patrón de entrada que el modelo espera, para poder predecir las ventas que podrían generar en el mes de agosto. El cual nos da como resultado redondeado “**respuesta.round()**” ya des estandarización, 13 chompas vendidas para el mes de agosto 2023.

3.2.8.6. Fase de entrenamiento (pantalón)

Figura 53

Realizar el proceso de entrenamiento de la Red Neuronal



```
▶ # Entrenar el modelo (PANTALON)
  history= model.fit(x_input, y_target_normalized, epochs=1000)
```

El modelo se entrena con los datos de entrada **x_input** y las etiquetas **y_target** durante 1000 épocas utilizando **model.fit ()**.

Figura 54

Resultado de Entrenamiento

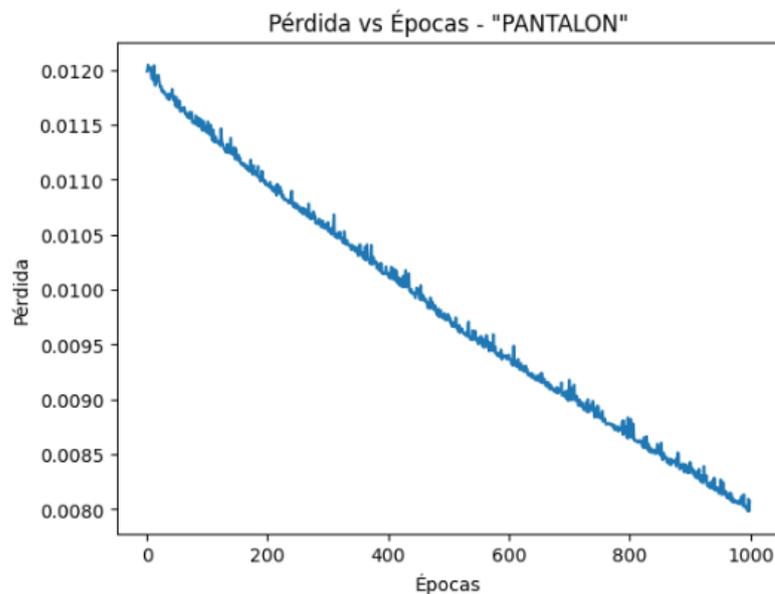
```
2/2 [=====] - 0s 8ms/step - loss: 0.0070 - mae: 0.0546
Epoch 995/1000
2/2 [=====] - 0s 5ms/step - loss: 0.0072 - mae: 0.0572
Epoch 996/1000
2/2 [=====] - 0s 8ms/step - loss: 0.0071 - mae: 0.0568
Epoch 997/1000
2/2 [=====] - 0s 7ms/step - loss: 0.0069 - mae: 0.0543
Epoch 998/1000
2/2 [=====] - 0s 6ms/step - loss: 0.0070 - mae: 0.0545
Epoch 999/1000
2/2 [=====] - 0s 6ms/step - loss: 0.0071 - mae: 0.0547
Epoch 1000/1000
2/2 [=====] - 0s 6ms/step - loss: 0.0069 - mae: 0.0545
```

Como se evidencia en la (Figura 54), el valor de pérdida (Loss) de 0.0069 indica el error entre las predicciones del modelo y los valores reales en el conjunto de evaluación, siendo más bajo deseable. Además, el Mean Absolute Error (MAE) de 0.0545 representa la media de las diferencias absolutas entre las predicciones y los valores reales, siendo un indicador bajo que sugiere que las predicciones son cercanas a los valores reales en el conjunto de evaluación.

El modelo ha aprendido bien a hacer predicciones en el conjunto de evaluación, ya que tanto la pérdida como el MAE son bajos. Sin embargo, estos resultados son específicos al conjunto de evaluación y no necesariamente indican cómo el modelo se desempeñará en datos que no ha visto antes (conjunto de prueba).

Figura 55

Gráfico de Pérdida vs Épocas



En la figura 55, se espera ver una disminución en la pérdida a medida que avanzan las épocas. Esto nos indica que el modelo está aprendiendo y mejorando su capacidad para hacer predicciones precisas. Después de cierto punto, se observa que la pérdida deja de disminuir significativamente. Esto significa que el modelo ha alcanzado un punto donde ya no mejora de manera significativa.

Tabla 10

Datos del Entrenamiento de la Red Neuronal (pantalón)

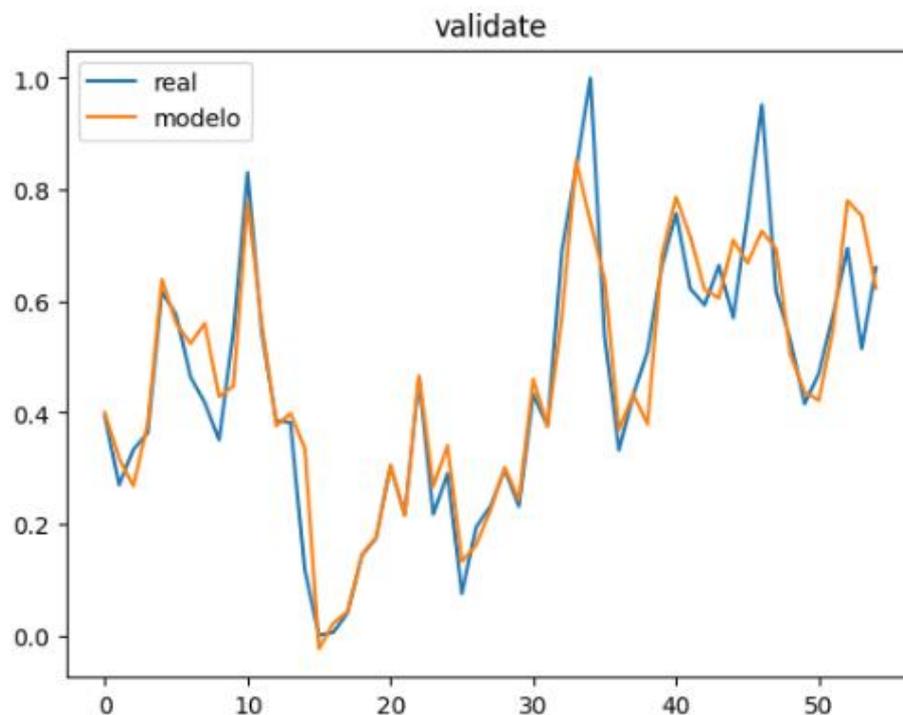
Venta Real	Ventas Desnormalizada	Pronóstico de la Red Neuronal	Error
401	0.393562	0.399909	0.006348
282	0.26999	0.317407	0.047418
343	0.333333	0.268154	0.065179
372	0.363448	0.379154	0.015706
616	0.616822	0.638698	0.021876
576	0.575286	0.559696	0.01559
469	0.464174	0.523362	0.059188
424	0.417445	0.560248	0.142803
360	0.350987	0.428627	0.07764
545	0.543094	0.447141	0.095953
822	0.830737	0.777943	0.052794
545	0.543094	0.558894	0.015799
393	0.385254	0.375924	0.00933
390	0.382139	0.398543	0.016404
135	0.117342	0.33586	0.218518
22	0	-0.024012	0.024012
27	0.005192	0.021712	0.01652
61	0.040498	0.042885	0.002386
160	0.143302	0.144807	0.001505
190	0.174455	0.176545	0.00209
314	0.303219	0.305705	0.002486
231	0.21703	0.215113	0.001917
466	0.461059	0.465647	0.004587
231	0.21703	0.267514	0.050484
302	0.290758	0.340581	0.049823
94	0.074766	0.132668	0.057901
209	0.194185	0.161946	0.032239
245	0.231568	0.224387	0.007181
308	0.296989	0.300984	0.003995
245	0.231568	0.243751	0.012183
439	0.433022	0.459524	0.026503
383	0.37487	0.374932	0.000062
686	0.689512	0.573078	0.116434

829	0.838006	0.85121	0.013204
985	1	0.742767	0.257233
539	0.536864	0.638265	0.101401
342	0.332295	0.368723	0.036428
437	0.430945	0.433326	0.002381
513	0.509865	0.379	0.130865
660	0.662513	0.682463	0.01995
751	0.757009	0.786115	0.029106
621	0.622015	0.715827	0.093813
593	0.592939	0.62033	0.027391
662	0.66459	0.605414	0.059175
571	0.570093	0.709779	0.139685
747	0.752856	0.668174	0.084681
939	0.952233	0.725504	0.226729
617	0.617861	0.693646	0.075785
532	0.529595	0.50407	0.025525
422	0.415369	0.436704	0.021336
475	0.470405	0.421759	0.048646
577	0.576324	0.550573	0.025751
691	0.694704	0.780573	0.085869
518	0.515057	0.753048	0.237991
658	0.660436	0.623185	0.037251

Se visualiza en la Tabla 10 la cantidad de venta de pantalón (real y des normalizada), se ve muy próximo a la cantidad que pronostica la red neuronal y consecuentemente el error disminuye significativamente

Figura 56

Gráfico de Entrenamiento de la Red Neuronal (pantalón)



Validación con los datos de Prueba

Para realizar la validación, separamos nuestros datos en dos, los datos de entrada en `x_input` y `y_target`, para validar la capacidad de la red neuronal los datos deben de estar normalizados.

Figura 57

Resultados con los Datos de Validación

```
✓ [63] score = model.evaluate(x_input,y_target)
0s print("\n %s: %.2f%%" % (model.metrics_names[0],score[0]*100))

2/2 [=====] - 0s 5ms/step - loss: 0.0070 - mae: 0.0555

loss: 0.70%
```

Como se visualiza en la Figura 53, nombrando `score` emplearemos el método "evaluate", Este método requiere dos parámetros de entrada: los datos de entrada almacenados en la variable "`x_input`" y los datos de salida en la variable "`y_target`" de nuestro conjunto de datos de prueba el error que tenemos para pronosticar con los datos de prueba es del 0.70%.

Tabla 11

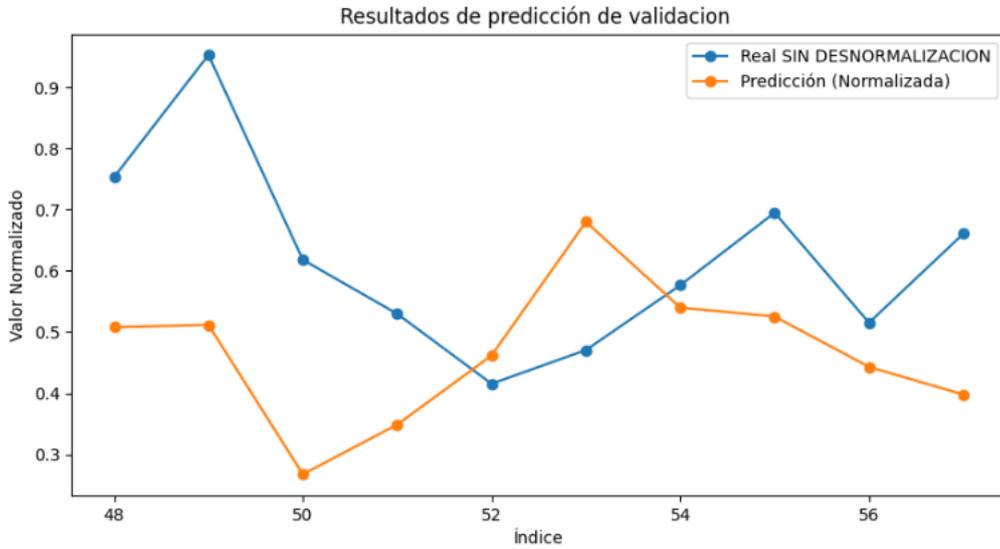
Datos de Validación (pantalón)

Venta Real	Ventas Desnormalizada	Pronóstico de la Red Neuronal	Error
747	0.752856	0.668174	0.084681
939	0.952233	0.725504	0.226729
617	0.617861	0.693646	0.075785
532	0.529595	0.50407	0.025525
422	0.415369	0.436704	0.021336
475	0.470405	0.421759	0.048646
577	0.576324	0.550573	0.025751
691	0.694704	0.780573	0.085869
518	0.515057	0.753048	0.237991
658	0.660436	0.623185	0.037251

En la Tabla 11, realizamos una partición de los 10 últimos meses de venta de chompa (real y des normalizada), se ve muy próximo a la cantidad que pronostica la red neuronal y consecuentemente el error disminuye significativamente

Figura 58

Gráfico de los Datos del Entrenamiento de la Red Neuronal (pantalón)



Nota: En la figura 58 se visualiza la proximidad muy cercana de la red real, al modelo de la red neuronal de los datos de validación.

Pronóstico

Figura 59

Pronóstico de la prenda pantalón

```
0s [▶] respuesta = model.predict([[691, 518, 658]])
print(respuesta.round())

1/1 [=====] - 0s 47ms/step
[[313.]]
```

(Ver Figura 59) Al pronosticar utilizaremos el método llamado “**model.predict**”, se utiliza para obtener las predicciones del modelo, en el cual, como datos de entrada, enviaremos como parámetros los tres últimos meses, es decir, de abril, junio y julio del 2023, se está pasando una secuencia de tres valores, que parece coincidir con el patrón de entrada que el modelo espera, para poder predecir las ventas que podrían generar en el mes de agosto. El cual nos da como resultado redondeado “**respuesta.round()**” ya des estandarización, 313 pantalones vendidas para el mes de agosto 2023.

3.2.8.7. Fase de entrenamiento (Sacos)

Figura 60

Realizar el proceso de entrenamiento de la Red Neuronal

```
✓ [21] # Entrenar el modelo (SACOS)
23 s history= model.fit(x_input, y_target_normalized, epochs=1000)
```

El modelo se entrena con los datos de entrada **x_input** y las etiquetas **y_target** durante 1000 épocas utilizando **model.fit ()**

Figura 61

Resultado de Entrenamiento

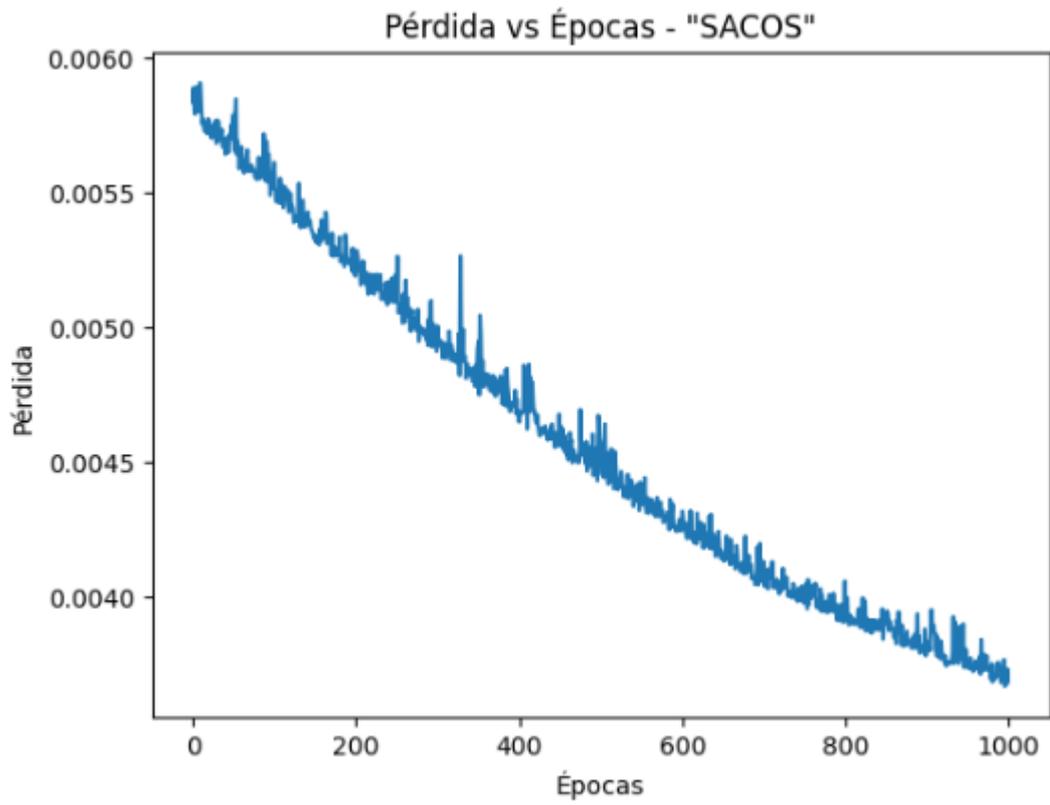
```
2/2 [=====] - 0s 5ms/step - loss: 0.0037 - mae: 0.0367
Epoch 995/1000
2/2 [=====] - 0s 5ms/step - loss: 0.0038 - mae: 0.0371
Epoch 996/1000
2/2 [=====] - 0s 5ms/step - loss: 0.0037 - mae: 0.0362
Epoch 997/1000
2/2 [=====] - 0s 5ms/step - loss: 0.0037 - mae: 0.0377
Epoch 998/1000
2/2 [=====] - 0s 5ms/step - loss: 0.0037 - mae: 0.0367
Epoch 999/1000
2/2 [=====] - 0s 5ms/step - loss: 0.0037 - mae: 0.0376
Epoch 1000/1000
2/2 [=====] - 0s 5ms/step - loss: 0.0037 - mae: 0.0368
```

Como se evidencia en la (Figura 61), el valor de pérdida (Loss) de 0.0037 indica el error entre las predicciones del modelo y los valores reales en el conjunto de evaluación, siendo más bajo deseable. Además, el Mean Absolute Error (MAE) de 0.0368 representa la media de las diferencias absolutas entre las predicciones y los valores reales, siendo un indicador bajo que sugiere que las predicciones son cercanas a los valores reales en el conjunto de evaluación.

El modelo ha aprendido bien a hacer predicciones en el conjunto de evaluación, ya que tanto la pérdida como el MAE son bajos. Sin embargo, estos resultados son específicos al conjunto de evaluación y no necesariamente indican cómo el modelo se desempeñará en datos que no ha visto antes (conjunto de prueba).

Figura 62

Gráfico de Pérdida vs Épocas



En la figura 62, se espera ver una disminución en la pérdida a medida que avanzan las épocas. Esto nos indica que el modelo está aprendiendo y mejorando su capacidad para hacer predicciones precisas. Después de cierto punto, se observa que la pérdida deja de disminuir significativamente. Esto significa que el modelo ha alcanzado un punto donde ya no mejora de manera significativa.

Tabla 12

Datos del Entrenamiento de la Red Neuronal (sacos)

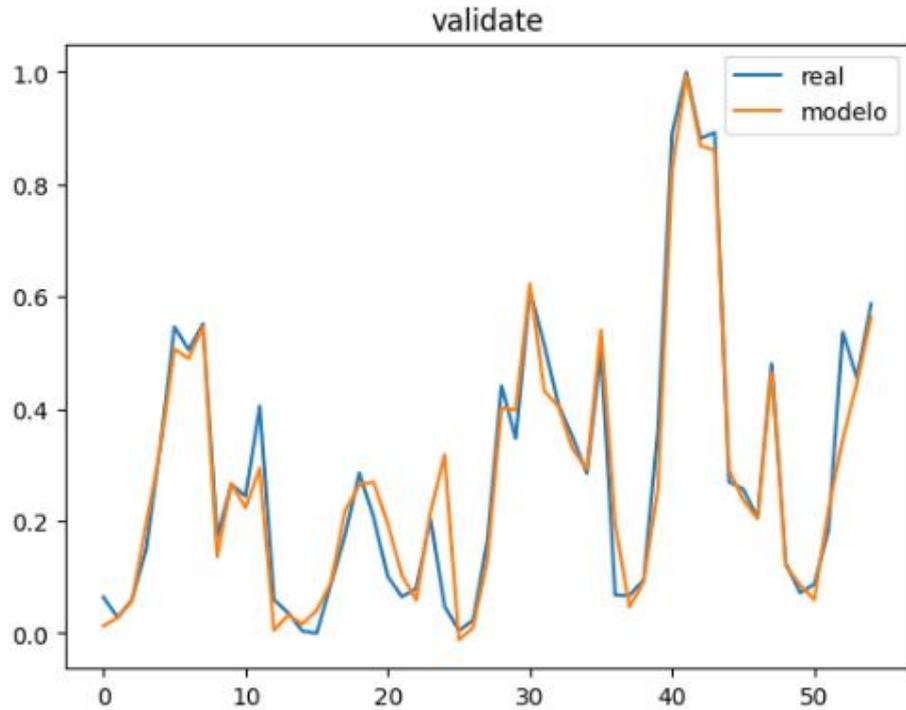
Venta Real	Ventas Desnormalizada	Pronóstico de la Red Neuronal	Error
104	0.064368	0.013013	0.051355
57	0.028352	0.027397	0.000956
97	0.059004	0.057084	0.00192
214	0.148659	0.192792	0.044133
454	0.332567	0.328452	0.004115
733	0.54636	0.506415	0.039945

679	0.504981	0.48973	0.015251
740	0.551724	0.547188	0.004537
244	0.171648	0.136573	0.035074
366	0.265134	0.266439	0.001305
339	0.244444	0.223886	0.020558
548	0.404598	0.293902	0.110696
98	0.05977	0.004917	0.054853
67	0.036015	0.0325	0.003516
25	0.003831	0.016468	0.012636
20	0	0.040735	0.040735
134	0.087356	0.08983	0.002474
248	0.174713	0.21784	0.043128
393	0.285824	0.264276	0.021548
290	0.206897	0.269643	0.062746
152	0.101149	0.195046	0.093897
105	0.065134	0.10332	0.038186
124	0.079693	0.059342	0.020351
287	0.204598	0.2145	0.009903
83	0.048276	0.318377	0.270101
26	0.004598	-0.012124	0.016722
50	0.022989	0.010207	0.012782
233	0.163218	0.129566	0.033653
595	0.440613	0.401743	0.03887
473	0.347126	0.398115	0.050988
817	0.610728	0.623377	0.012649
692	0.514943	0.432353	0.08259
552	0.407663	0.403821	0.003842
476	0.349425	0.32807	0.021355
392	0.285057	0.292755	0.007698
675	0.501916	0.540674	0.038758
109	0.068199	0.194649	0.126449
108	0.067433	0.046478	0.020955
144	0.095019	0.091856	0.003163
492	0.361686	0.257985	0.103701
1184	0.891954	0.826387	0.065568
1325	1	0.994637	0.005363
1171	0.881992	0.869289	0.012704
1185	0.89272	0.860653	0.032068
372	0.269732	0.291942	0.02221
355	0.256705	0.237217	0.019488
289	0.20613	0.20413	0.002001
647	0.48046	0.464495	0.015965
182	0.124138	0.119557	0.004581
114	0.072031	0.086336	0.014306
134	0.087356	0.059427	0.027929
261	0.184674	0.2213	0.036626
720	0.536398	0.345419	0.19098
615	0.455939	0.445996	0.009943
786	0.586973	0.564225	0.022748

Se visualiza en la Tabla 12 la cantidad de venta de casaca (real y des normalizada), se ve muy próximo a la cantidad que pronostica la red neuronal y consecuentemente el error disminuye significativamente

Figura 63

Gráfico de Entrenamiento de la Red Neuronal (sacos)



Nota: En la figura 63 se visualiza la proximidad muy cercana de la red real, al modelo de la red neuronal

Validación con los datos de Prueba

Para realizar la validación, separamos nuestros datos en dos, los datos de entrada en `x_input` y `y_target`, para validar la capacidad de la red neuronal los datos deben de estar normalizados.

Figura 64

Resultados con los Datos de Validación

```
0.5 ✓ ▶ score = model.evaluate(x_input,y_target)
print("\n %s: %.2f%%" % (model.metrics_names[0],score[0]*100))

2/2 [=====] - 0s 5ms/step - loss: 0.0037 - mae: 0.0375

loss: 0.37%
```

Como se visualiza en la Figura 64, nombrando score emplearemos el método "evaluate", Este método requiere dos parámetros de entrada: los datos de entrada almacenados en la variable "x_input" y los datos de salida en la variable "y_target" de nuestro conjunto de datos de prueba el error que tenemos para pronosticar con los datos de prueba es del 0.37%.

Tabla 13

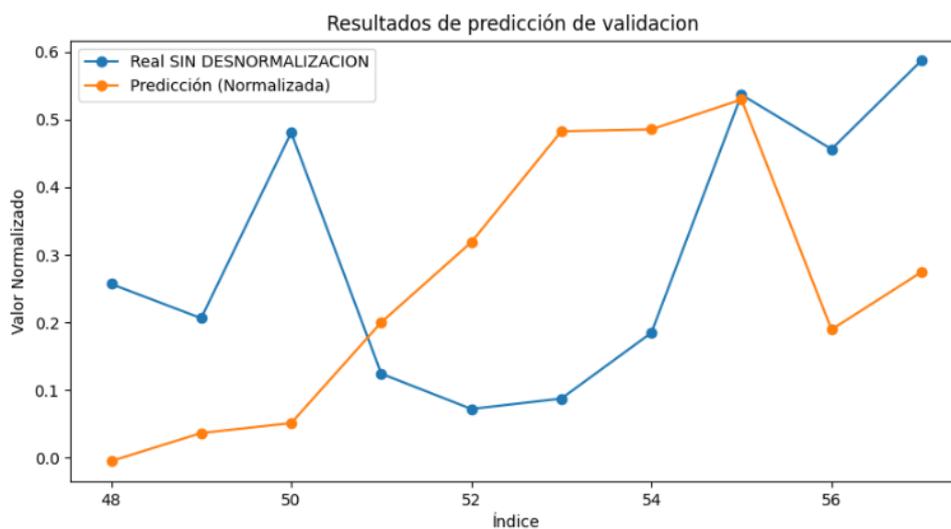
Datos de Validación (sacos)

Venta Real	Ventas Desnormalizada	Pronóstico de la Red Neuronal	Error
355	0.256705	0.237217	0.019488
289	0.20613	0.20413	0.002001
647	0.48046	0.464495	0.015965
182	0.124138	0.119557	0.004581
114	0.072031	0.086336	0.014306
134	0.087356	0.059427	0.027929
261	0.184674	0.2213	0.036626
720	0.536398	0.345419	0.19098
615	0.455939	0.445996	0.009943
786	0.586973	0.564225	0.022748

En la Tabla 13, realizamos una partición de los 10 últimos meses de venta de chompa (real y des normalizada), se ve muy próximo a la cantidad que pronostica la red neuronal y consecuentemente el error disminuye significativamente

Figura 65

Gráfico de los Datos del Entrenamiento de la Red Neuronal (Sacos)



Nota: En la figura 65 se visualiza la proximidad muy cercana de la red real, al modelo de la red neuronal de los datos de validación.

Pronostico

Figura 66

Pronóstico de la prenda Saco

```
✓ 0s ▶ respuesta = model.predict([[720, 615, 786]])  
print(respuesta.round())  
  
1/1 [=====] - 0s 27ms/step  
[[668.]]
```

(Ver Figura 66) Al pronosticar utilizaremos el método llamado “**model.predict**”, se utiliza para obtener las predicciones del modelo, en el cual, como datos de entrada, enviaremos como parámetros los tres últimos meses, es decir, de abril, junio y julio del 2023, se está pasando una secuencia de tres valores, que parece coincidir con el patrón de entrada que el modelo espera, para poder predecir las ventas que podrían generar en el mes de agosto. El cual nos da como resultado redondeado “**respuesta.round()**” ya des estandarización, 668 sacos vendida para el mes de agosto 2023.

3.2.8.9. Fase de entrenamiento (short)

Figura 67

Realizar el proceso de entrenamiento de la Red Neuronal

```
▶ # Entrenar el modelo (SHORT)  
history= model.fit(x_input, y_target_normalized, epochs=1000)
```

El modelo se entrena con los datos de entrada **x_input** y las etiquetas **y_target** durante 1000 épocas utilizando **model.fit ()**

Figura 68

Resultado de Entrenamiento

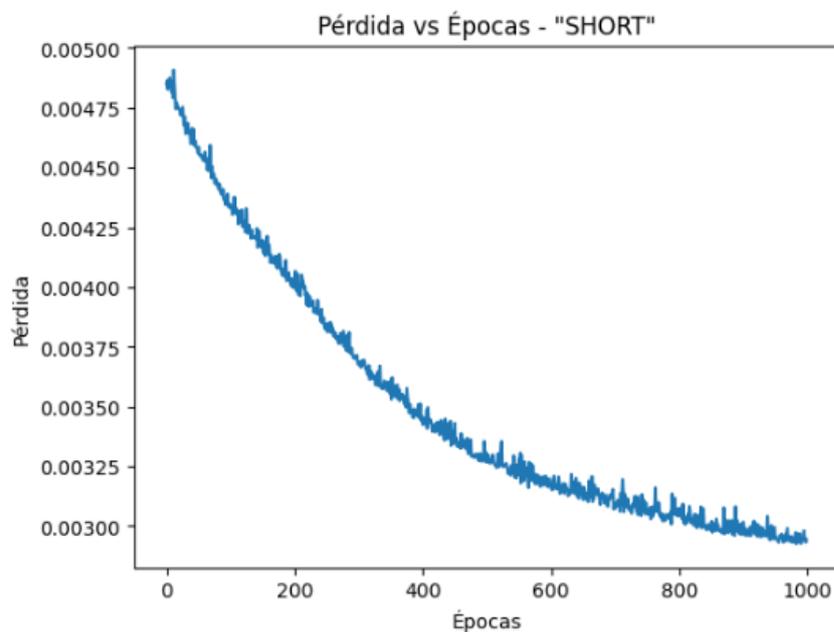
```
Epoch 996/1000
2/2 [=====] - 0s 8ms/step - loss: 0.0030 - mae: 0.0278
Epoch 997/1000
2/2 [=====] - 0s 8ms/step - loss: 0.0029 - mae: 0.0278
Epoch 998/1000
2/2 [=====] - 0s 8ms/step - loss: 0.0030 - mae: 0.0270
Epoch 999/1000
2/2 [=====] - 0s 8ms/step - loss: 0.0029 - mae: 0.0269
Epoch 1000/1000
2/2 [=====] - 0s 10ms/step - loss: 0.0029 - mae: 0.0276
```

Como se evidencia en la (Figura 68), el valor de pérdida (Loss) de 0.0029 indica el error entre las predicciones del modelo y los valores reales en el conjunto de evaluación, siendo más bajo deseable. Además, el Mean Absolute Error (MAE) de 0.0276 representa la media de las diferencias absolutas entre las predicciones y los valores reales, siendo un indicador bajo que sugiere que las predicciones son cercanas a los valores reales en el conjunto de evaluación.

El modelo ha aprendido bien a hacer predicciones en el conjunto de evaluación, ya que tanto la pérdida como el MAE son bajos. Sin embargo, estos resultados son específicos al conjunto de evaluación y no necesariamente indican cómo el modelo se desempeñará en datos que no ha visto antes (conjunto de prueba).

Figura 69

Gráfico de Pérdida vs Épocas



En la figura 69, se espera ver una disminución en la pérdida a medida que avanzan las épocas. Esto nos indica que el modelo está aprendiendo y mejorando su capacidad para hacer predicciones precisas. Después de cierto punto, se observa que la pérdida deja de disminuir significativamente. Esto significa que el modelo ha alcanzado un punto donde ya no mejora de manera significativa.

Tabla 14

Datos del Entrenamiento de la Red Neuronal (short)

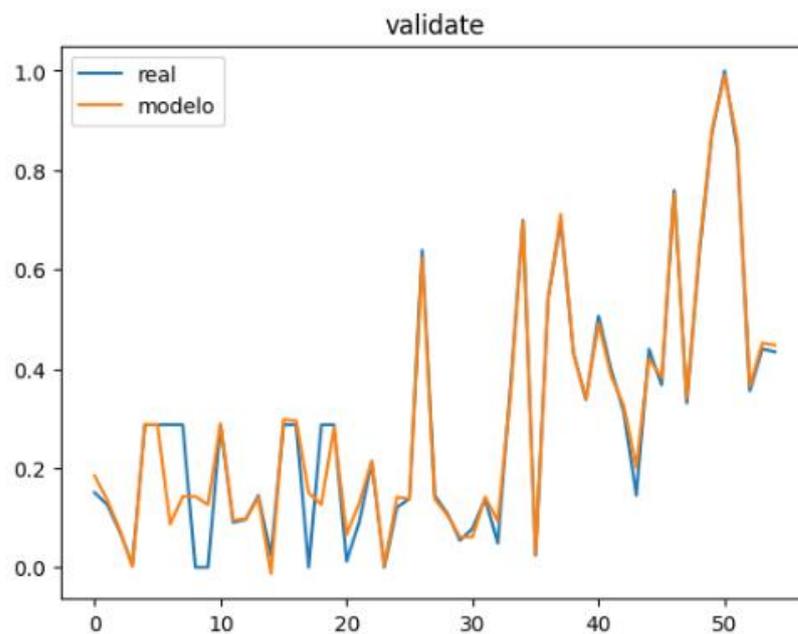
Venta Real	Ventas Desnormalizada	Pronóstico de la Red Neuronal	Error
26	0.150602	0.184631	0.034029
22	0.126506	0.136135	0.009629
13	0.072289	0.075124	0.002835
2	0.006024	0.00129	0.004734
49	0.286978	0.287813	0.000836
49	0.286978	0.287084	0.000106
49	0.286978	0.086629	0.200349
49	0.286978	0.142552	0.144426
1	0	0.142552	0.142552
1	0	0.126188	0.126188
49	0.286978	0.289139	0.002162
16	0.090361	0.094687	0.004326
17	0.096386	0.097889	0.001504
25	0.144578	0.140379	0.004199
5	0.024096	-0.012869	0.036966
49	0.286978	0.297712	0.010734
49	0.286978	0.295114	0.008136
1	0	0.149408	0.149408
49	0.286978	0.126188	0.16079
49	0.286978	0.281547	0.005431
3	0.012048	0.065702	0.053654
16	0.090361	0.128459	0.038097
36	0.210843	0.214511	0.003667
1	0	0.003631	0.003631
21	0.120482	0.14128	0.020798
24	0.138554	0.13779	0.000764
107	0.638554	0.62325	0.015304
25	0.144578	0.137227	0.007351
19	0.108434	0.105384	0.00305
10	0.054217	0.060939	0.006722
14	0.078313	0.061311	0.017002
24	0.138554	0.141765	0.003211
9	0.048193	0.092941	0.044748

61	0.361446	0.345564	0.015882
117	0.698795	0.696568	0.002227
5	0.024096	0.026194	0.002098
91	0.542169	0.540163	0.002006
117	0.698795	0.711073	0.012278
73	0.433735	0.431027	0.002708
57	0.337349	0.339069	0.00172
85	0.506024	0.49087	0.015154
67	0.39759	0.385723	0.011867
53	0.313253	0.325432	0.012179
25	0.144578	0.201338	0.056759
74	0.439759	0.420072	0.019687
62	0.36747	0.38218	0.014711
127	0.759036	0.75267	0.006366
56	0.331325	0.340339	0.009014
106	0.63253	0.641074	0.008544
146	0.873494	0.882424	0.00893
167	1	0.991031	0.008969
141	0.843373	0.860285	0.016911
60	0.355422	0.365153	0.009731
74	0.439759	0.451881	0.012122
73	0.433735	0.447478	0.013743

Se visualiza en la Tabla 14 la cantidad de venta de short (real y des normalizada), se ve muy próximo a la cantidad que pronostica la red neuronal y consecuentemente el error disminuye significativamente

Figura 70

Gráfico de entrenamiento de la Red Neuronal (short)



Nota: En la figura 70 se visualiza la proximidad muy cercana de la red real, al modelo de la red neuronal

Validación con los datos de prueba

Para realizar la validación, separamos nuestros datos en dos, los datos de entrada en `x_input` y `y_target`, para validar la capacidad de la red neuronal los datos deben de estar normalizados.

Figura 71

Resultados con los Datos de Validación

```

0s ✓ ▶ score = model.evaluate(x_input,y_target)
    print("\n %s: %.2f%%" % (model.metrics_names[0],score[0]*100))

2/2 [=====] - 0s 8ms/step - loss: 0.0029 - mae: 0.0278

    loss: 0.29%
    
```

Como se visualiza en la Figura 71, nombrando `score` emplearemos el método "evaluate", Este método requiere dos parámetros de entrada: los datos de entrada almacenados en la variable "`x_input`" y los datos de salida en la variable "`y_target`" de nuestro conjunto de datos de prueba el error que tenemos para pronosticar con los datos de prueba es del 0.29%.

Tabla 15

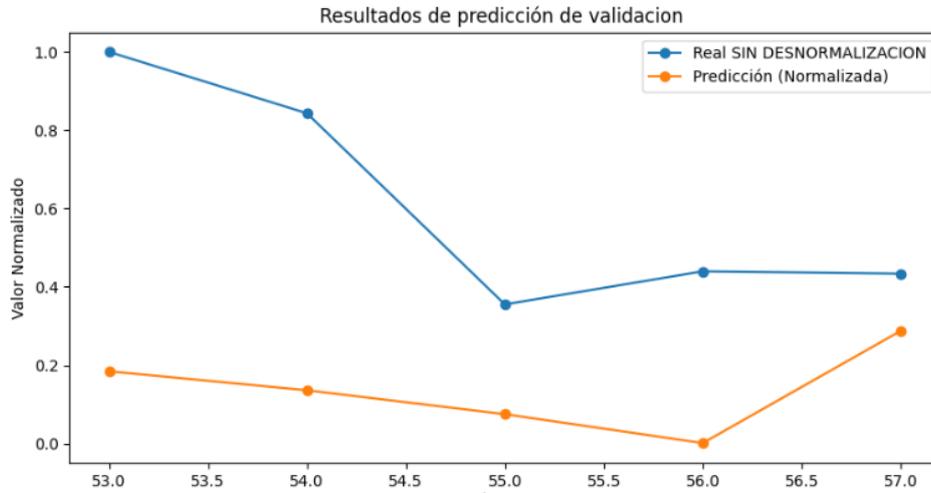
Datos de Validación (short)

Venta Real	Ventas Desnormalizada	Pronóstico de la Red Neuronal	Error
56	0.331325	0.340339	0.009014
106	0.63253	0.641074	0.008544
146	0.873494	0.882424	0.00893
167	1	0.991031	0.008969
141	0.843373	0.860285	0.016911
60	0.355422	0.365153	0.009731
74	0.439759	0.451881	0.012122
73	0.433735	0.447478	0.013743

En la Tabla 15, realizamos una partición de los 10 últimos meses de venta de chompa (real y des normalizada), se ve muy próximo a la cantidad que pronostica la red neuronal y consecuentemente el error disminuye significativamente

Figura 72

Gráfico de los datos del entrenamiento de la Red Neuronal (short)



Nota: En la figura 72 se visualiza la proximidad muy cercana de la red real, al modelo de la red neuronal de los datos de validación.

Pronóstico

Figura 73

Pronóstico de la prenda vestido

```
✓ [42] respuesta = model.predict([[60, 74, 73]])  
0s print(respuesta.round())  
  
1/1 [=====] - 0s 41ms/step  
[[48.]]
```

(Ver Figura 73) Al pronosticar utilizaremos el método llamado “**model.predict**”, se utiliza para obtener las predicciones del modelo, en el cual, como datos de entrada, enviaremos como parámetros los tres últimos meses, es decir, de abril, junio y julio del 2023, se está pasando una secuencia de tres valores, que parece coincidir con el patrón de entrada que el modelo espera, para poder predecir las ventas que podrían generar en el mes de agosto. El cual nos da como resultado redondeado “**respuesta.round()**” ya des estandarización, 48 shorts vendidas para el mes de agosto 2023.

3.2.8.11. Fase de entrenamiento (vestidos)

Figura 74

Realizar el proceso de entrenamiento de la Red Neuronal

```
[3] # Entrenar el modelo (VESTIDOS)
    history= model.fit(x_input, y_target_normalized, epochs=1000)
```

El modelo se entrena con los datos de entrada **x_input** y las etiquetas **y_target** durante 1000 épocas utilizando **model.fit ()**

Figura 75

Resultado de Entrenamiento

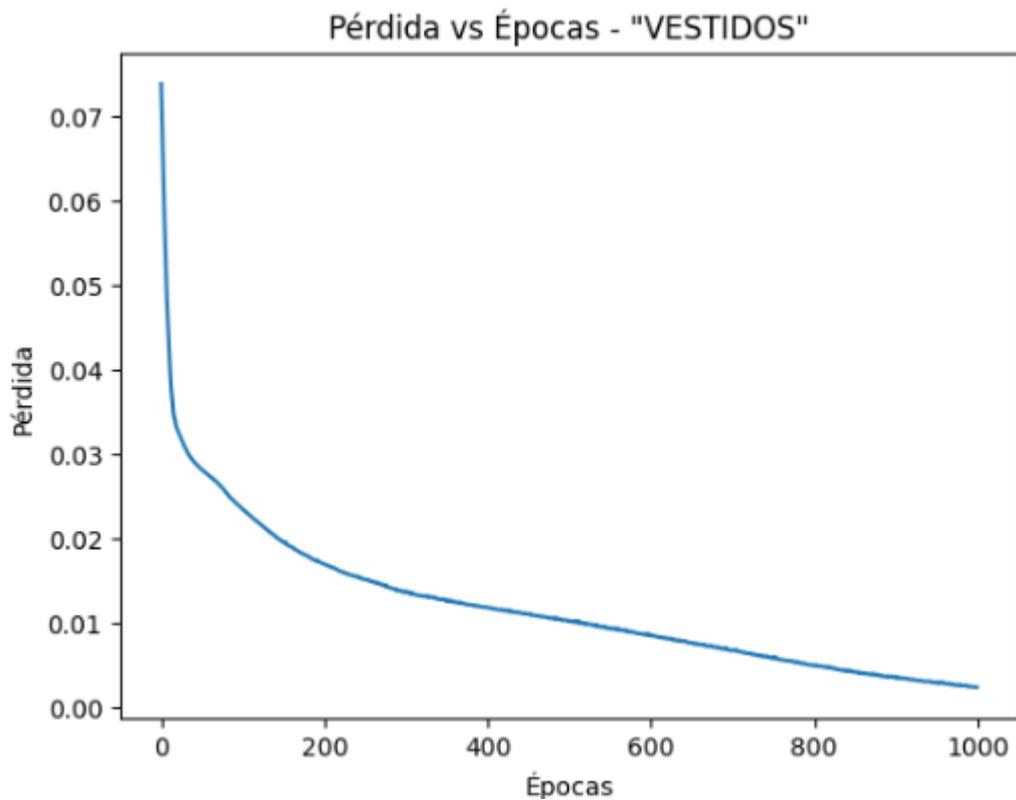
```
Epoch 996/1000
2/2 [=====] - 0s 12ms/step - loss: 0.0024 - mae: 0.0305
Epoch 997/1000
2/2 [=====] - 0s 8ms/step - loss: 0.0024 - mae: 0.0306
Epoch 998/1000
2/2 [=====] - 0s 8ms/step - loss: 0.0024 - mae: 0.0307
Epoch 999/1000
2/2 [=====] - 0s 8ms/step - loss: 0.0024 - mae: 0.0308
Epoch 1000/1000
2/2 [=====] - 0s 10ms/step - loss: 0.0023 - mae: 0.0303
Epoch 1000/1000
2/2 [=====] - 0s 10ms/step - loss: 0.0024 - mae: 0.0303
```

Como se evidencia en la (Figura 75), el valor de pérdida (Loss) de 0.0024 indica el error entre las predicciones del modelo y los valores reales en el conjunto de evaluación, siendo más bajo deseable. Además, el Mean Absolute Error (MAE) de 0.0303 representa la media de las diferencias absolutas entre las predicciones y los valores reales, siendo un indicador bajo que sugiere que las predicciones son cercanas a los valores reales en el conjunto de evaluación.

El modelo ha aprendido bien a hacer predicciones en el conjunto de evaluación, ya que tanto la pérdida como el MAE son bajos. Sin embargo, estos resultados son específicos al conjunto de evaluación y no necesariamente indican cómo el modelo se desempeñará en datos que no ha visto antes (conjunto de prueba).

Figura 76

Gráfico de Pérdida vs Épocas



En la figura 76, se espera ver una disminución en la pérdida a medida que avanzan las épocas. Esto nos indica que el modelo está aprendiendo y mejorando su capacidad para hacer predicciones precisas. Después de cierto punto, se observa que la pérdida deja de disminuir significativamente. Esto significa que el modelo ha alcanzado un punto donde ya no mejora de manera significativa.

Tabla 16

Datos del Entrenamiento de la Red Neuronal (vestidos)

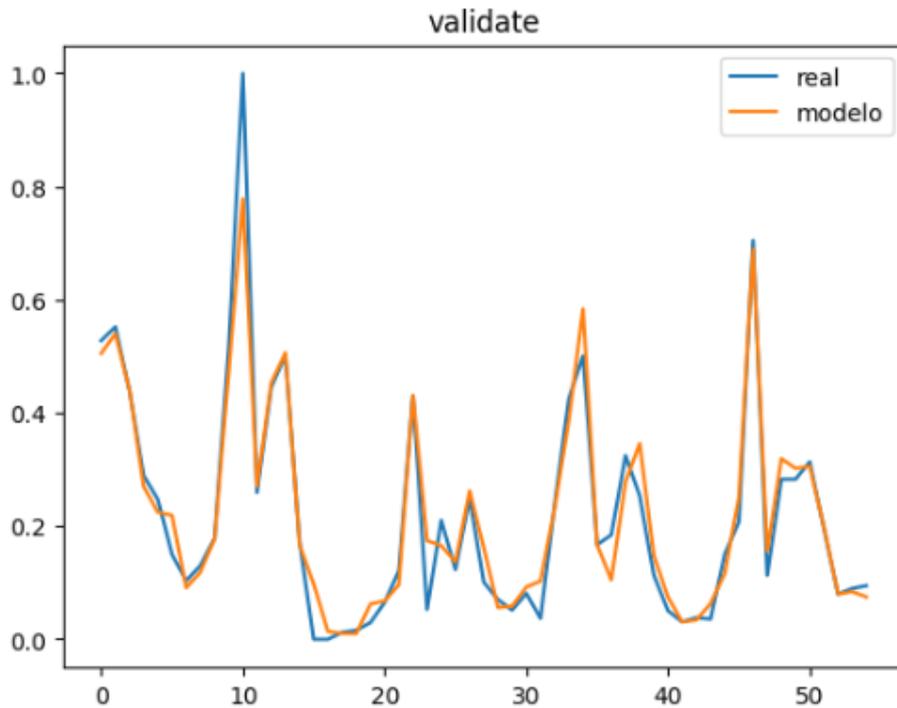
Venta Real	Ventas Desnormalizada	Pronóstico de la Red Neuronal	Error
432	0.527675	0.505095	0.02258
452	0.552276	0.540078	0.012197
360	0.439114	0.440496	0.001382
238	0.289053	0.269701	0.019352
204	0.247232	0.224573	0.02266
125	0.150062	0.21909	0.069028
87	0.103321	0.090982	0.012339

109	0.130381	0.118784	0.011597
148	0.178352	0.177602	0.000749
421	0.514145	0.47193	0.042215
816	1	0.778713	0.221287
214	0.259533	0.270376	0.010844
365	0.445264	0.452817	0.007553
411	0.501845	0.506507	0.004662
143	0.172202	0.166528	0.005674
3	0	0.097364	0.097364
3	0	0.014084	0.014084
13	0.0123	0.010311	0.001989
16	0.01599	0.009966	0.006025
27	0.02952	0.062529	0.033009
56	0.065191	0.067625	0.002434
101	0.120541	0.096336	0.024205
353	0.430504	0.430837	0.000332
46	0.052891	0.174459	0.121568
174	0.210332	0.165496	0.044836
103	0.123001	0.13669	0.013689
206	0.249692	0.26236	0.012667
85	0.100861	0.161556	0.060695
60	0.070111	0.056235	0.013876
45	0.051661	0.058649	0.006989
69	0.081181	0.092313	0.011132
33	0.0369	0.102946	0.066046
197	0.238622	0.232127	0.006495
349	0.425584	0.387317	0.038267
410	0.500615	0.583848	0.083233
139	0.167282	0.166455	0.000827
153	0.184502	0.105008	0.079494
267	0.324723	0.27792	0.046803
209	0.253383	0.346114	0.092731
95	0.113161	0.148072	0.034911
44	0.050431	0.074262	0.023831
28	0.03075	0.030877	0.000127
34	0.03813	0.034007	0.004124
32	0.03567	0.06303	0.02736
125	0.150062	0.116044	0.034017
172	0.207872	0.251541	0.043669
576	0.704797	0.689478	0.015319
95	0.113161	0.155902	0.042741
233	0.282903	0.319208	0.036305
233	0.282903	0.302142	0.019239
258	0.313653	0.305656	0.007997
162	0.195572	0.199032	0.00346
68	0.079951	0.080077	0.000126
76	0.089791	0.084257	0.005534
80	0.094711	0.074848	0.019863

Se visualiza en la Tabla 16 la cantidad de venta de casaca (real y des normalizada), se ve muy próximo a la cantidad que pronostica la red neuronal y consecuentemente el error disminuye significativamente

Figura 77

Gráfico de entrenamiento de la Red Neuronal (vestidos)



Nota: En la figura 77 se visualiza la proximidad muy cercana de la red real, al modelo de la red neuronal

Validación con los datos de Prueba

Para realizar la validación, separamos nuestros datos en dos, los datos de entrada en `x_input` y `y_target`, para validar la capacidad de la red neuronal los datos deben de estar normalizados.

Figura 78

Resultados con los Datos de Validación

```
0s ✓ ▶ score = model.evaluate(x_input,y_target)
print("\n %s: %.2f%%" % (model.metrics_names[0],score[0]*100))

2/2 [=====] - 0s 7ms/step - loss: 0.0024 - mae: 0.0302
loss: 0.24%
```

Como se visualiza en la Figura 78, nombrando score emplearemos el método "evaluate", Este método requiere dos parámetros de entrada: los datos de entrada almacenados en la variable "**x_input**" y los datos de salida en la variable "**y_target**" de nuestro conjunto de datos de prueba el error que tenemos para pronosticar con los datos de prueba es del 0.24%.

Tabla 17

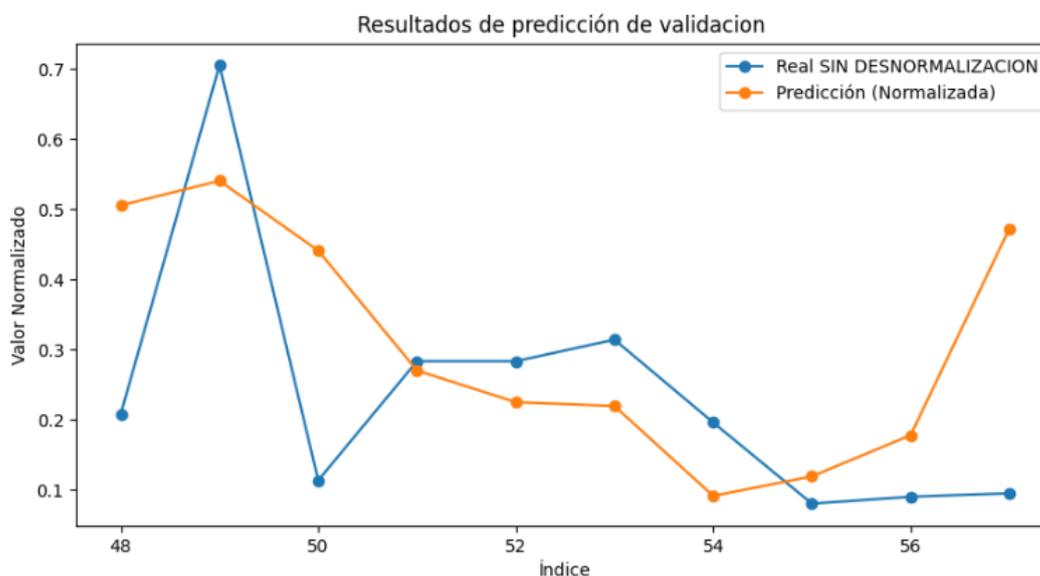
Datos de Validación (vestidos)

Venta Real	Ventas Desnormalizada	Pronóstico de la Red Neuronal	Error
172	0.207872	0.251541	0.043669
576	0.704797	0.689478	0.015319
95	0.113161	0.155902	0.042741
233	0.282903	0.319208	0.036305
233	0.282903	0.302142	0.019239
258	0.313653	0.305656	0.007997
162	0.195572	0.199032	0.00346
68	0.079951	0.080077	0.000126
76	0.089791	0.084257	0.005534
80	0.094711	0.074848	0.019863

En la Tabla 17, realizamos una partición de los 10 últimos meses de venta de chompa (real y des normalizada), se ve muy próximo a la cantidad que pronostica la red neuronal y consecuentemente el error disminuye significativamente.

Figura 79

Gráfico de los datos del entrenamiento de la Red Neuronal (vestidos)



Nota: En la figura 79 se visualiza la proximidad muy cercana de la red real, al modelo de la red neuronal de los datos de validación.

Pronóstico

Figura 80

Pronóstico de la prenda vestido

```

✓ [22] respuesta = model.predict([[68, 76, 80]])
0s print(respuesta.round())

1/1 [=====] - 0s 68ms/step
[[53.]]

```

(Ver Figura 80) Al pronosticar utilizaremos el método llamado “**model.predict**”, se utiliza para obtener las predicciones del modelo, en el cual, como datos de entrada, enviaremos como parámetros los tres últimos meses, es decir, de abril, junio y julio del 2023, se está pasando una secuencia de tres valores, que parece coincidir con el patrón de entrada que el modelo espera, para poder predecir las ventas que podrían generar en el mes de agosto. El cual nos da como resultado redondeado “**respuesta.round()**” ya des estandarización, 53 vestidos vendida para el mes de agosto 2023.

Gráficos de análisis

Tabla 18

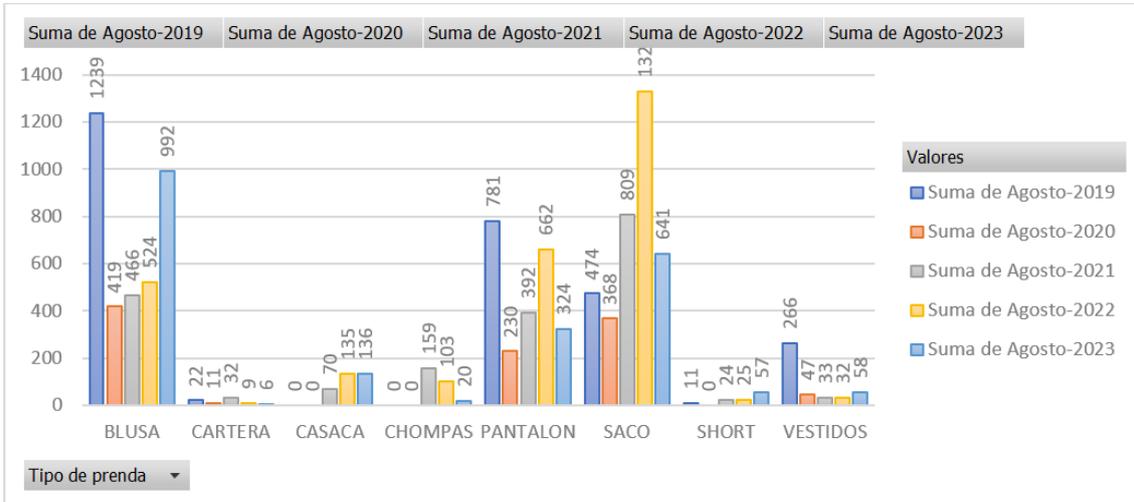
Histórico de ventas de los meses de agosto

No	Tipo de prenda	Agosto-2019	Agosto-2020	Agosto-2021	Agosto-2022	Agosto-2023
1	BLUSA	1,239	419	466	524	992
2	CARTERA	22	11	32	9	6
3	CASACA	0	0	70	135	136
4	PANTALÓN	781	230	392	662	324
5	SACO	474	368	809	1,329	641
6	SHORT	11	0	24	25	57
7	VESTIDOS	266	47	33	32	58
8	CHOMPAS	0	0	159	103	20
Total		2,793	1,075	1,985	2,819	2,234

Nota. En la tabla muestra las cantidades vendidas de los últimos años de los meses de agosto.

Figura 80

Histórico de ventas de los meses de agosto



Nota. En la figura muestra el análisis de las ventas por prenda de los meses de agosto desde el 2018 hasta la actualidad.

Tabla 19

Resumen de la predicción de venta para el mes de agosto

No	Tipo de prenda	Predicción para agosto 2023
1	BLUSA	1,000
2	CARTERA	5
3	CASACA	129
4	PANTALÓN	313
5	SACO	668
6	SHORT	48
7	VESTIDOS	53
8	CHOMPAS	13
Total		2,229

Nota: En la tabla muestra los resultados de las predicciones que se realizaron para el mes de agosto

Presupuesto.

El costo para la implementación del proyecto, respecto al software es S/ 00.00, debido que se realizó en la plataforma de colab google que es Open Source y tiene una capacidad mayor a 80 GB de almacenamiento y también se implementó librerías Open Source. De igual manera se detalla el costo de pago de personal y equipos de hardware.

Tabla 20

Costos para la implementación del proyecto

Concepto	Descripción	costos
Colab	Open Source	S/ 0.00
Librerías	Open Source	S/ 0.00
Pago al personal	Desarrollador del proyecto	S/ 2800.00
Equipo pc	Recurso para la implementación de la herramienta a desarrollar	S/ 2600.00
Electricidad	Consumo durante el desarrollo del proyecto	S/ 50.00
total		S/ 5450.00

Dato: Presupuesto

3.3 Resultados

Los resultados obtenidos fueron lo siguiente

- Se ha logrado diseñar un modelo de redes neuronales adaptado a las necesidades específicas de la predicción de tendencias de moda, específicamente para las ventas de prendas en la empresa R&D Corporación. Este modelo incorpora características clave del conjunto de datos recopilado, estructurado con 3 longitudes de entrada, 3 capas ocultas con 16 neuronas con función de activación Relu y una capa de salida con 1 sola neurona con la función de activación Lineal óptimo para problemas de regresión, garantizando una representación efectiva de los patrones y relaciones presentes
- El modelo de redes neuronales fue entrenado con éxito utilizando datos relevantes de venta de la empresa R&D Corporación. Durante este proceso, las redes neuronales artificiales para el análisis de cada prenda, ha aprendido de manera eficaz las complejidades y estructuras subyacentes en los datos, demostrando su capacidad para generalizar y prever tendencias futuras con precisión, obteniendo como porcentaje de error menor a 0.70%
- La aplicación del modelo entrenado a conjuntos de datos específicos ha permitido evaluar de manera integral su impacto en

las decisiones comerciales y la rentabilidad de R&D Corporaciones. Los resultados de esta evaluación destacan la utilidad práctica del modelo en la toma de decisiones estratégicas. Se realizó el pronóstico para el mes de agosto 2023, donde se pudo visualizar que se vendara 1000 blusas, 5 carteras, 129 carteras, 313 pantalones, 668 sacos, 48 shorts, 53 vestidos y 13 chompas.

CONCLUSIONES

- El diseño de un modelo de redes neuronales adaptado a la predicción de tendencias de moda establece una base sólida para la aplicación de técnicas avanzadas en el ámbito de la moda. La incorporación de características clave en una estructura con 3 longitudes de entrada y 3 capas ocultas demuestra una cuidadosa adaptación a la complejidad de los datos. El diseño, con activaciones Relu y Lineal, garantiza una representación efectiva de los patrones y relaciones, sentando las bases para pronósticos precisos en el ámbito de la moda.

- La fase de entrenamiento del modelo, evidenciada por un bajísimo porcentaje de error menor a 0.70%, destaca la capacidad de la red neuronal para aprender y generalizar complejidades en los datos de tendencias de moda. La precisión alcanzada en la predicción de futuras tendencias subraya la eficacia del modelo en anticipar de manera confiable los patrones emergentes en la empresa R&D Corporación, brindando una herramienta valiosa para la toma de decisiones informadas.

- La evaluación del impacto en las decisiones comerciales y la rentabilidad de R&D Corporaciones, revela una utilidad práctica tangible. La capacidad del modelo para realizar pronósticos, como se evidenció en la predicción de los 8 tipos de prendas vendidas para el mes de agosto, ayuda en el momento de lanzar una nueva colección y analizar que prendas se venderá más al momento de producir dicha prenda con más demanda, destaca su contribución significativa a la toma de decisiones estratégicas.

RECOMENDACIONES

- Respecto al diseño del modelo de redes neuronales, se recomienda mantener mejoras continuas, realizar ajustes adicionales en la arquitectura de la red, probar diferentes funciones de activación o incluso considerar la implementación de técnicas más avanzadas podría potenciar aún más la capacidad del modelo para capturar patrones emergentes en la industria de la moda.

- La fase de entrenamiento exitosa y la baja tasa de error destacan la eficacia del modelo, pero es esencial establecer un sistema de monitoreo continuo. La industria de la moda es dinámica, y patrones y preferencias pueden cambiar con el tiempo. Mantener el modelo actualizado con nuevos datos garantizará que siga siendo relevante y preciso a medida que evolucionen las tendencias del mercado.

- Como se visualiza en los resultados del entrenamiento de la red, la prenda que más se vendería son las blusas, con un promedio de 1000 unidades, seguidamente del saco, que tendrá un promedio de venta en 668 unidades y la que menos se vendería son las chompas y carteras con 13 y 5 unidades, respectivamente, en ese caso se recomienda que para el mes de agosto se diseñe y produzca al mercado más blusas y sacos, para cubrir la demandas y evitar tener pérdidas de producción. De igual manera, se sugiere utilizar los resultados del modelo no solo para la planificación operativa, sino también para análisis estratégicos más profundos que informen decisiones clave y maximicen el impacto positivo en la rentabilidad de la empresa.

REFERENCIAS BIBLIOGRÁFICAS

Osnayo, C., & Jesús, C. (2018). Aplicación de redes neuronales artificiales para la predicción de la recuperación de planta concentradora en Minsur S.A. Tesis de titulación, Facultad de Ingeniería de Procesos. Universidad Nacional de San Agustín de Arequipa, <http://repositorio.unsa.edu.pe/handle/UNSA/6629>

Jiménez, A., & Soraya, A. (2010). Análisis del proceso de adopción de tecnologías de información y comunicaciones en actividades de aprovisionamiento empresarial en pequeñas y medianas empresas manufactureras. Tesis doctoral, Facultad de ingeniería informática, Universidad Politécnica de Valencia, <http://hdl.handle.net/10251/8935>.

Montaño moreno (2000). Redes Neuronales Artificiales aplicadas al Análisis de Datos. Tesis Doctoral, Facultad de Psicología, Universidad de las illes balears, <https://disi.unal.edu.co/~lctorress/RedNeu/LiRna001.pdf>

Wilmer Rivas, Bertha Mazon (2017). Redes neuronales artificiales aplicadas al reconocimiento de patrones. Generalidades de las redes neuronales artificiales, editorial Utmach. <https://pdfcoffee.com/redesneuronalesartificialesaplicadasalreconomiendodepatr>

Vidaurre Siadén, Yasmín Andrea (2012). Aplicación de las redes neuronales artificiales para el pronóstico de la demanda de agua potable en la Empresa Epsel S.A. de la ciudad de Lambayeque. Tesis de Titulación, Facultad de Ingeniería, Universidad Católica Santo Toribio de Mogrovejo, <http://hdl.handle.net/20.500.12423/525>

Montemayor Gallegos, José Enrique (2014). Métodos de pronósticos para negocios, Editorial General, <https://repositorio.tec.mx/handle/11285/621230>

Catunta, Saulo y Roncal, Dalyuska (2020). Redes neuronales recurrentes para la predicción de casos por Covid-19 usando datos abiertos en el Perú. Tesis para Titulación, Facultad de Ingeniería y Arquitectura, Universidad Cesar Vallejo. <https://hdl.handle.net/20.500.12692/63041>

Pavel, Cotacallapa (2022). Redes neuronales convolucionales para datos composicionales: Una aplicación a la industria textil de la moda. Tesis para grado

Magíster, Escuela Posgrado, Pontificia Universidad Católica del Perú.
<http://hdl.handle.net/20.500.12404/22095>

Delgado Cervantes, Deyli (2020). Estudio de las emisiones de Nox mediante redes neuronales recurrentes. Tesis para grado Máster, Departamento de Ingeniería de Organización, Administración de Empresas y Estadística, Universidad Politécnica de Madrid. <https://oa.upm.es/65419/>

Alejandro Gamarra, Luis Enrique (2020). Modelo De Red Neuronal Artificial Para El Pronóstico Del Crecimiento De Contagios Por Covid-19 en el Distrito De San Juan De Miraflores. Tesis de Titulación, Facultad de Ingeniería y Gestión, Universidad Nacional Tecnológica de Lima Sur. <https://repositorio.untels.edu.pe/jspui/handle/123456789/680>

Wes McKinney (02/2023). Manipulación de datos con pandas, NyumPy y Jupyter, Editorial Anaya Multimedia. <https://is.gd/k66hH8>

Travis E. Oliphant. (2006). Guide to NumPy https://www.researchgate.net/publication/213877900_Guide_to_NumPy

ANEXO

Ventas diarias de las diferentes prendas de la empresa R&D Corporación.

Año	Fecha	BLUSAS	CARTER	CASACA	CHOMPÁ	ENTERIZ	FALDA	PANTALC	POLO	SACOS	SHORT	VESTIDO	
2018	octubre	97	2				6	74		80		43	
2018	noviembre	529	21				82	348		323		326	
2018	diciembre	1169	36				76	535		201	11	673	
2019	enero	750	14				72	401		104	26	432	
2019	febrero	486	21				41	282		57	22	452	
2019	marzo	635	19				41	343		97	13	360	
2019	abril	502	18				75	372		214	2	238	
2019	mayo	974	14			1	84	616		454		204	
2019	junio	586	6			4	71	576		733		125	
2019	julio	547	17			4	81	469		679		87	
2019	agosto	512	19			1	40	424		740		109	
2019	octubre	742	13				33	360		244	1	148	
2019	noviembre	1050	14				70	545		366	1	421	
2019	diciembre	1804	18				101	822		339		816	
2019	septiembre	817	25			2	117	545		548	16	214	
2020	enero	1091	10				84	393		98	17	365	
2020	febrero	1117	11				43	390		67	25	411	
2020	marzo	336	11				28	135		25	5	143	
2020	abril	38					4	22		20		3	
2020	mayo	119					8	27		134		3	
2020	junio	168	7				11	61		248	1	13	
2020	julio	281	13				21	160		393		16	
2020	agosto	282	10				26	190	100	290		27	
2020	octubre	534	12	28			47	314	161	152	3	56	
2020	noviembre	710	13	52			38	231	230	105	16	101	
2020	diciembre	1375	41	109			54	466	563	124	36	353	
2020	septiembre	520	14				37	231	33	287	1	46	
2021	enero	1042	25	29			40	302	392	83	21	174	
2021	febrero	294	4	24			49	94	163	26	24	103	
2021	marzo	958	10	73			71	209	349	50	107	206	
2021	abril	619	16	85			39	245	189	233	25	85	
2021	mayo	481	17	109	121		8	65	308	198	595	19	60
2021	junio	305	14	78	95		5	71	245	221	473	10	45
2021	julio	386	30	83	208	3	114	439	362	817	14	69	
2021	agosto	376	32	70	159	2	103	383	289	692	24	33	
2021	octubre	758	30	128	126	2	216	686	394	552	9	197	
2021	noviembre	1272	49	148	47		8	217	829	373	476	61	349
2021	diciembre	1881	73	136	24	31	258	985	438	392	117	410	
2021	septiembre	566	38	140	258	3	153	539	359	675	5	139	
2022	enero	953	35	55	12	33	116	342	256	109	91	153	
2022	febrero	1250	29	58	12	49	117	437	354	108	117	267	
2022	marzo	1325	24	88	28	35	140	513	380	144	73	209	
2022	abril	980	22	115	54	9	100	660	543	492	57	95	
2022	mayo	1281	27	184	78	2	208	751	364	1184	85	44	
2022	junio	819	10	91	34	2	145	621	228	1325	67	28	
2022	julio	653	16	171	32	1	159	593	180	1171	53	34	
2022	agosto	479	9	126	103	2	125	662	205	1185	25	32	
2022	octubre	1295	15	431	7		136	571	285	372	74	125	
2022	noviembre	1427	10	329	2	41	179	747	642	355	62	172	
2022	diciembre	1562	13	169			80	212	939	850	289	127	576
2022	septiembre	1083	11	355	68	5	153	617	293	647	56	95	
2023	enero	1120		103			119	532	384	182	106	233	
2023	febrero	1013		50			150	422	250	114	146	233	
2023	marzo	1302		48	1		151	475	444	134	167	258	
2023	abril	1235		80			185	577	313	261	141	162	
2023	mayo	1240		204	4		187	691	257	720	60	68	
2023	junio	910		295	15		144	518	317	615	74	76	
2023	julio	1072		296	36		137	658	309	786	73	80	

Nota: ventas del 2018 a 2023

Entrevistas:

Entrevista	
Datos personales	
Apellidos y Nombres: Ortega Alex	
Fecha: 25/07/2023	Cargo: Jefe de Almacén
Área: Acabados	
1. ¿Cómo analizas las tendencias de ventas y la demanda de productos?	
<input checked="" type="checkbox"/> Histórico de ventas	
<input type="checkbox"/> Herramientas de análisis de datos	
<input type="checkbox"/> Retroalimentación de clientes	
<input type="checkbox"/> Otro (especificar)	
2. ¿Cuál es el sistema actual de gestión de inventario en tu empresa?	
<input type="checkbox"/> Manual	
<input checked="" type="checkbox"/> Software de ventas y gestión de inventario	
3. ¿Cuáles crees que son los principales motivos para la producción en exceso?	
<input checked="" type="checkbox"/> Pronósticos incorrectos de demanda	
<input type="checkbox"/> Falta de coordinación entre departamentos	
<input type="checkbox"/> Estrategias de marketing ineficaces	
<input type="checkbox"/> Otro (especificar)	
4. ¿Cómo se manejan los productos obsoletos o con baja rotación?	
Se aplican descuentos del 20 % o 30 % para reducir el excedente y evitar tener descuadres de inventario en el almacén	
5. ¿Cuál es el proceso para determinar las cantidades a producir?	
Se evalúa las ventas de cada tienda y dependiendo a la demanda refregada, se calcula la cantidad a producir para la siguiente temporada	
6. ¿Cómo reajustan la producción en respuesta a cambios en la demanda?	
A través de coordinación entre el área de producción y área de venta	