CIMAT

Centro de Investigación en Matemáticas, A.C.

EVALUACIÓN Y COMPARACIÓN DE MODELOS DE APRENDIZAJE DE MÁQUINA MEJORADOS POR METAHEURÍSTICAS PARA LA ELABORACIÓN DE PRONÓSTICOS DE VENTA MEDIANTE SERIES TEMPORALES CORTAS

T E S I S

Que para obtener el grado de

Maestro en Modelación y Optimización de Procesos

Presenta

Fernando Vital Perea

Director de la tesis:

Dr. Héctor de la Torre Gutiérrez

Codirector de la tesis:

Dr. Luis Eduardo Urbán Rivero

Autorización de la versión final

Aguascalientes, Ags., 24 de septiembre de 2021

Resumen

El problema de los pronósticos de ventas, es general para todas las organizaciones. Su importancia se ve reflejada en todas las áreas de las empresas, y especialmente, en los costos operacionales. Para empresas con poco tiempo en el mercado, con nuevos productos o aquellas que no tienen técnicas automatizadas de recolección de información, se vuelve aún más crítico contar con un pronóstico acertado de sus ventas. La principal ventaja de dichos pronósticos es que se puede administrar de mejor manera toda la cadena de suministros, logrando ventajas competitivas que benefician directamente a la empresa y sus clientes. A partir de estos pronósticos se puede generar un plan maestro de producción que dictamine la dinámica laboral y operacional de la organización. Cuando las organizaciones no disponen de grandes volúmenes de datos para realizar sus pronósticos, se puede analizar la información existente con la intención de obtener información valiosa para su toma de decisiones. En esta investigación se resolvió el problema de pronósticos de ventas para una empresa de manufactura, debido a que la cantidad de información disponible es limitada, en este caso series temporales cortas, se decidió implementar modelos de aprendizaje de máquina que pudieran construir dichos pronósticos. De igual manera, se buscó mejorar la precisión de dichos modelos mediante algoritmos metaheurísticos aplicados al ajuste de hiperparámetros. Los resultados obtenidos muestran que éstos algoritmos tienen efectos significativos para el comportamiento y desempeño de los modelos de ML, con respecto a una selección aleatoria o con base en experiencia, de los mencionados hiperparámetros.

Dedicatoria

A los brazos que me han sostenido y acompañado toda mi vida y, que con amor y paciencia, me han hecho ser quien soy.

Agradecimientos

Agradezco a todos aquellos que hicieron posible la realización de esta tesina, de manera particular:

- Al Consejo Nacional de Ciencia y Tecnología (CONACyT), por la beca de manutención que me permitió estudiar este posgrado.
- A la Unidad Aguascalientes del Centro de Investigación en Matemáticas, por todo el apoyo que recibí durante mis estudios y por todo el conocimiento adquirido por parte de sus investigadores.
- Al Dr. Héctor de la Torre Gutiérrez, por su valiosa amistad, dirigir esta tesina y contribuir de manera significativa en mi crecimiento personal y profesional.
- Al Dr. Luis Eduardo Urbán Rivero, por co-dirigir este trabajo, por todas las vivencias compartidas durante mi estancia en Aguascalientes y por su valiosa amistad.
- A mis padres y familia, quienes siempre me han apoyado y estado conmigo en las buenas y las no tan buenas.
- A la Dra. Merari del Socorro Prado Anaya, por su amor, cariño e inagotable paciencia. Nada de esto sería posible sin tu compañía y motivación constante.
- A mis amigos de siempre, por ser el ejemplo de personas que son y por caminar conmigo esta vida.
- A mis compañeros de la Unidad Aguascalientes, este viaje no hubiera sido igual sin ustedes.
- A todas las nuevas amistades forjadas en Aguascalientes, en especial al Lic. Francisco Martínez: ¡qué va!.

Índice general

Resumen I

Dedicatoria II

Agradecimientos III

1. Introducción 1

1.1. Objetivo de la tesina . . . 2

1.1.1. Objetivo general . . . 2

1.1.2. Objetivos específicos . . . 2

1.2. Justificación del estudio . . . 3

1.3. Hipótesis . . . 3

2. Revisión de literatura 4

2.1. El problema de los pronósticos . . . 4

2.2. Análisis de series temporales y algoritmos de Machine Learning . 6

2.3. Uso combinado de algoritmos de Machine Learning y algoritmos metaheurísticos . . . 7

2.4. Algoritmos de Machine Learning y algoritmos metaheurísticos aplicados a series temporales cortas . . . 8

3. Marco teórico 10

3.1. Inteligencia de Negocios . . . 10

3.1.1. El rol de los pronósticos en el control de la demanda . . . 12

3.2. Análisis de series de tiempo . . . 14

3.2.1. Series de tiempo estacionarias . . . 16

3.2.2. Modelo Autorregresivo Integrado de Medias Móviles (ARIMA) . . . 17

3.2.3. Descomposición STL 19
3.3. Aprendizaje de Máquina (ML) 20
3.3.1. Aprendizaje estadístico 21
3.3.2. Transformación de series temporales a aprendizaje supervisado 21
3.3.3. Regresión por Vectores de Soporte (SVR) 23
3.3.4. Redes neuronales 25
3.3.5. Redes LSTM 31
3.3.6. Algoritmo XGBoost 33
3.3.7. Redes LSTM Convolucionales (ConvLSTM) 34
3.3.8. Redes LSTM combinadas con Redes Neuronales Convolucionales (CNN-LSTM) 35
3.4. Algoritmos metaheurísticos 35
3.4.1. Recocido simulado 37
3.4.2. Búsqueda aleatoria 39
3.4.3. Estrategias evolutivas 40
3.5. Métricas para la evaluación del desempeño 41

4. Resultados **44**
4.1. Análisis exploratorio de los datos 44
4.1.1. Producto 1 45
4.1.2. Producto 2 46
4.1.3. Producto 3 47
4.1.4. Producto 4 49
4.1.5. Producto 5 49
4.2. Implementación de la función auto_arima 51
4.3. Aplicación de algoritmos de ML 58
4.3.1. Soporte de regresión vectorial 58
4.3.2. Red neuronal de una capa oculta con una neurona 62
4.3.3. Red neuronal multicapa 67
4.3.4. Red neuronal LSTM con 50 neuronas 70

4.3.5. Red neuronal LSTM apilado 73
4.3.6. Red neuronal LSTM bidireccional 76
4.3.7. Algoritmo XGBoost 79
4.3.8. CNN LSTM 82
4.3.9. Conv LSTM 86
4.4. Aplicación de algoritmos de ML y metaheurísticos 89
4.5. Análisis estadístico de los resultados 94
4.5.1. Distribución probabilística de los resultados 95

5. Conclusiones y trabajos futuros **106**
5.1. Trabajo futuro 111

Bibliografía **113**

A. Análisis de varianza multivariado y univariado **A-1**
A.1. Validación de supuestos MANOVA A-1
A.2. Validación de supuestos ANOVA A-4

Índice de figuras

3.1. Diagrama del flujo de información en un MRP. Fuente: elaboración propia. 13
3.2. Ejemplo de serie temporal. 15
3.3. Ejemplo de descomposición STL. 19
3.4. SVR lineal unidimensional. 24
3.5. Efecto de la aplicación de la función kernel. 26
3.6. Neurona como unidad computacional. 27
3.7. Diagrama de una red neuronal multicapa. 30
3.8. Diagrama de una célula LSTM. 32

4.1. Venta de productos 45
4.2. Venta del producto 1 45
4.3. Análisis del producto 1 46
4.4. Venta del producto 2 47
4.5. Análisis del producto 2 47
4.6. Venta del producto 3 48
4.7. Análisis del producto 3 48
4.8. Venta del producto 4 49
4.9. Análisis del producto 4 50
4.10. Venta del producto 5 50
4.11. Análisis del producto 5 51
4.12. Predicciones de la función auto_arima contra los valores reales observados de ventas en los últimos cinco periodos de tiempo para el producto 1. 52
4.13. Predicciones de la función auto_arima contra los valores reales observados de ventas en los últimos cinco periodos de tiempo para el producto 2. 53

4.14. Predicciones de la función auto_arima contra los valores reales observados de ventas en los últimos cinco periodos de tiempo para el producto 3. 54
4.15. Predicciones de la función auto_arima contra los valores reales observados de ventas en los últimos cinco periodos de tiempo para el producto 4. 55
4.16. Predicciones de la función auto_arima contra los valores reales observados de ventas en los últimos cinco periodos de tiempo para el producto 5. 56
4.17. Predicciones del SVR contra los valores reales de venta observados en los últimos cinco periodos temporales. 61
4.18. Predicciones del NN contra los valores reales observados de ventas en los últimos cinco periodos de tiempo para el Producto 3. 64
4.19. Predicciones del NN contra los valores reales observados de ventas en los últimos cinco periodos de tiempo para el Producto 4. 65
4.20. Predicciones del NN contra los valores reales observados de ventas en los últimos cinco periodos de tiempo para el Producto 5. 67
4.21. Predicciones del NN Multicapa contra los valores reales de venta observados en los últimos cinco periodos temporales. 70
4.22. Predicciones del NN LSTM con 50 neuronas contra los valores reales de venta observados en los últimos cinco periodos temporales. 73
4.23. Predicciones de la NN LSTM apilado contra los valores reales de venta observados en los últimos cinco periodos temporales. 76
4.24. Predicciones la NN LSTM Bidireccional contra los valores reales de venta observados en los últimos cinco periodos temporales. . . 79
4.25. Predicciones del algoritmo XGBoost contra los valores reales de venta observados en los últimos cinco periodos temporales. 82
4.26. Predicciones del CNN LSTM contra los valores reales de venta observados en los últimos cinco periodos temporales. 85
4.27. Predicciones del Conv LSTM contra los valores reales de venta observados en los últimos cinco periodos temporales. 88
4.28. Histograma de frecuencias para la métrica 1. 96
4.29. Histograma de frecuencias para la métrica 2. 97

5.1. Comparación entre las funciones y modelos de predicción para la métrica 1. 108
5.2. Comparación entre las funciones y modelos de predicción para la métrica 2. 109
5.3. Comparación entre las funciones y modelos de predicción para la métrica 3. 110
5.4. Comparación entre las funciones y modelos de predicción para la métrica 3. 111

Índice de tablas

3.1. Ejemplo del conjunto de datos de una serie temporal. 22
3.2. Ejemplo de la serie temporal modificada para aprendizaje supervisado. 23

4.1. Métricas Producto 1 auto_arima. 52
4.2. Métricas Producto 2 auto_arima. 53
4.3. Métricas Producto 3 auto_arima. 54
4.4. Métricas Producto 4 auto_arima. 55
4.5. Métricas Producto 5 auto_arima. 56
4.6. Resumen de resultados para la función auto_arima en los cinco productos. 57
4.7. Resultados para Soporte de Regresión Vectorial en los cinco productos. 60
4.8. Métricas Producto 1 NN. 62
4.9. Comparación pronosticado contra real: Producto 1 NN. 62
4.10. Métricas Producto 2 NN. 63
4.11. Comparación pronosticado contra real: Producto 2 NN. 63
4.12. Métricas Producto 3 NN. 64
4.13. Métricas Producto 4 NN. 64
4.14. Métricas Producto 5 NN. 65
4.15. Resumen de resultados para Red Neuronal de una capa oculta con una neurona en los cinco productos. 66
4.16. Resumen de resultados para Red Neuronal Multicapa en los cinco productos. 69
4.17. Resumen de resultados para Red Neuronal LSTM con 50 neuronas en los cinco productos. 72
4.18. Resumen de resultados para Red Neuronal LSTM apilado en los cinco productos. 75

4.19. Resumen de resultados para Red Neuronal LSTM bidireccional en los cinco productos. 78
4.20. Resumen de resultados para XGBoost en los cinco productos. . . 81
4.21. Resumen de resultados para CNN LSTM en los cinco productos. . 84
4.22. Resumen de resultados para Conv LSTM en los cinco productos. . 87
4.23. Resultados de la mejora del modelo Soporte de Regresión Vectorial. 90
4.24. Resultados de la mejora del modelo Red Neuronal de una capa y una neurona. 90
4.25. Resultados de la mejora del modelo Red Neuronal multicapa. . . . 91
4.26. Resultados de la mejora del modelo Red Neuronal LSTM con 50 neuronas 91
4.27. Resultados de la mejora del modelo Red Neuronal LSTM apilado. 92
4.28. Resultados de la mejora del modelo Red Neuronal LSTM bidireccional. 92
4.29. Resultados de la mejora del modelo XGBoost. 93
4.30. Resultados de la mejora del modelo CNN-LSTM. 93
4.31. Resultados de la mejora del modelo Conv-LSTM. 94
4.32. Resultados de la obtención de parámetros y prueba Kolmogorov-Smirnov para las métricas del producto 1. 97
4.33. Resultados de la regresión Beta para el producto 1. 98
4.34. Resultados de la obtención de parámetros y prueba Kolmogorov-Smirnov para las métricas del producto 2. 99
4.35. Resultados de la regresión beta para el producto 2. 100
4.36. Resultados de la obtención de parámetros y prueba Kolmogorov-Smirnov para las métricas del producto 3. 101
4.37. Resultados de la regresión beta para el producto 3. 101
4.38. Resultados de la obtención de parámetros y prueba Kolmogorov-Smirnov para las métricas del producto 4. 102
4.39. Resultados de la regresión beta para el producto 4. 103
4.40. Resultados de la obtención de parámetros y prueba Kolmogorov-Smirnov para las métricas del producto 5. 104
4.41. Resultados de la regresión beta para el producto 5. 104

5.1. Mejor combinación de algoritmos de ML y Metaheurísticos para cada producto. 107

A.1. Resultados de la prueba de normalidad multivariada mediante la prueba Doornik-Hansen. A-3

A.2. Resultados de la prueba de normalidad multivariada tras la transformación de las métricas, mediante la prueba Doornik-Hansen. . A-3

A.3. Resultados de la prueba de normalidad Anderson-Darling. A-4

A.4. Resultados de la prueba de igualdad de varianzas de Levene. . . . A-5

Capítulo 1

Introducción

La capacidad de predecir es una habilidad que el ser humano ha buscado obtener desde siempre. Aunque se tenía la creencia de que ésta era un don, era común al menos en la Grecia clásica encontrar los santuarios oraculares como Delfos, Dodona, Delos, Dídima, entre otros; en los cuales además de ser un punto neurálgico de la política griega, se solían hacer consultas sobre el futuro y principalmente: anticipar el resultado de alguna guerra o batalla. Situaciones similares se presentaron en todo el mundo, incluso en América con el ejemplo particular de la cultura inca en donde estos oráculos representaban la principal forma de adivinación.

Si bien la práctica de la predicción o adivinación del futuro en el ámbito social ha evolucionado y todavía se cuestionan sus resultados al estar apartados del método científico, el avance acelerado del desarrollo tecnológico ha permitido desarrollar avanzados algoritmos que permiten con cierto grado de precisión realizar predicciones. Éstas hoy en día son cruciales en el mundo de los negocios y de acuerdo con Christou en [1] se pueden clasificar en: (a) pronósticos económicos, que se focalizan en la predicción de valores y cantidades macro-económicas como tipos de cambio, tasas de inflación, entre otros; (b) pronósticos tecnológicos, cuyo propósito es predecir puntos de inflexión en términos de innovación; y por último (c) pronósticos de la demanda en los que se busca predecir la venta de productos de los cuales se tiene un registro previo y preciso.

Es en este último apartado de pronóstico de demanda en donde tiene cobijo esta tesina, justamente aprovechando los últimos desarrollos en materia de Aprendizaje de Máquina, Inteligencia Artificial y algoritmos metaheurísticos para optimización buscando resolver el problema de pronósticos de demanda de una empresa de manufactura. La compañía centra sus esfuerzos principalmente en la fabricación, distribución y venta de herramientas para uso doméstico e industrial y se busca con este trabajo crear una ventaja competitiva en términos de costos. Afortunadamente se cuentan con registros históricos de algunas líneas de productos y se emplea dicha información pasada para realizar predicciones en un futuro

a corto plazo. En este trabajo de investigación se emplean métodos como análisis de series de tiempo mediante modelos ARIMA, soporte de regresión vectorial adaptado de máquinas de soporte vectorial, y distintos tipos de redes neuronales; todos los anteriores mejorados en sus parámetros por algortimos metaheurísticos que permiten acelerar el tiempo de refinamiento de los modelos predictivos e integrarse todos a un sistema inteligente que evalúa y selecciona la mejor combinación para una serie de tiempo dada.

1.1. Objetivo de la tesina

Debido a la naturaleza del proyecto a realizar, los objetivos están directamente relacionados con la obtención de buenos resultados evaluándose en sus respectivas métricas.

1.1.1. Objetivo general

Evaluar diversos métodos de aprendizaje de máquina y los mismos mejorados por algoritmos metaheurísticos para elaborar pronósticos de ventas de manera precisa y eficiente de acuerdo al tipo de información histórica disponible y proporcionada por la empresa, específicamente de series temporales univariables cortas.

1.1.2. Objetivos específicos

- Analizar los registros históricos de ventas por línea de producto para determinar y describir estadísticamente su comportamiento.
- Describir y construir los modelos de aprendizaje de máquina a usar para caracterizar sus propiedades funcionales.
- Describir e implementar los algoritmos metaheurísticos en los modelos de aprendizaje de máquina y obtener los mejores parámetros de funcionamiento para mejorar su rendimiento.
- Evaluar la precisión de los pronósticos mediante las métricas comúnmente empleadas en la industria para aprendizaje de máquina.
- Seleccionar un modelo estadístico adecuado para evaluar la significancia de los algoritmos metaheurísticos en el ajuste de hiperparámetros en modelos de aprendizaje de máquina aplicados a series temporales cortas.

- Seleccionar tras el análisis de los resultados, el modelo que mejor desempeño tenga de acuerdo a las métricas empleadas.
- Encontrar evidencia científica que soporte el uso de modelos de aprendizaje de máquina y sus mejoras mediante metaheurísticas, en series temporales cortas.

1.2. Justificación del estudio

Actualmente la empresa dueña de los datos a utilizar en este trabajo realiza pronósticos basados en experiencias previas y hace uso del módulo de pronósticos de su sistema de planeación de recursos (ERP) corporativo. Se sabe que dicho módulo no tiene ninguna implementación de aprendizaje de máquina para pronósticos. El desarrollar una herramienta ajustada al tipo de información disponible supone como consecuencia, una mejora sustancial para la elaboración de estrategias operativas y de mercado. En otras palabras, se espera tener reflejados ahorros en costos operacionales al establecer con anticipación planes de producción acordes a las predicciones de ventas realizadas por los modelos a evaluar en esta tesina, además de la creación de nuevas oportunidades de negocio y líneas de productos que permitan incrementar la rentabilidad de la organización.

1.3. Hipótesis

Existe un modelo de aprendizaje de máquina mejorado por algoritmos metaheurísticos, capaz de predecir con al menos una precisión de 90 % evaluada mediante la métrica adecuada, las ventas de una línea de producto para los siguientes cinco meses a partir de la última observación histórica empleando como información de entrenamiento el registro histórico de al menos dos años y seis meses anteriores.

Capítulo 2

Revisión de literatura

La elaboración de pronósticos ha sido ampliamente estudiada por la comunidad científica relacionada a la estadística, matemáticas, ciencias de la computación y actuaría [2]; además, se ha mostrado un especial interés en la industria gracias al enorme potencial de su aplicación en beneficio de ahorro de costos entre otros [3]. En este capítulo se exploran y documentan una parte de la gran diversidad de propuestas de solución a la elaboración de pronósticos de series de tiempo de ventas desde la comunidad científica con la intención de conocer lo realizado hasta el momento así como alternativas para trabajos futuros. El presente capítulo se divide en tres secciones; en la primera se plantea de manera general el problema de la realización de pronósticos, en la segunda sección se exploran algunas de las propuestas de la comunidad científica para resolver los retos de los pronósticos incluyendo algoritmos de aprendizaje de máquina, y en la última parte se exploran los trabajos en donde se han combinado algoritmos metaheurísticos junto con los de aprendizaje de máquina.

2.1. El problema de los pronósticos

En la economía de mercado actual donde el objetivo de todas las empresas y organizaciones es generar e incrementar las utilidades, existen varios aspectos a considerar para optimizar dicha rentabilidad del negocio. Por ejemplo se quiere minimizar costos de producción, almacén, recursos humanos y desperdicios generales al mismo tiempo que se quiere maximizar las ventas, la capacidad productiva, eficiencia de máquinas y personas así como darle continuidad a la generación de valor y economía. Es aquí donde entra en juego la generación de pronósticos precisos y confiables sobre ventas ya que el contar con las predicciones adecuadas permitirá a la dirección tomar las decisiones operativas y de negocio que le ayude a realizar una mejor planificación de su cadena de valor, recursos humanos, reducción de costos en almacén, logro de resultados, entre otros; al

reducir la incertidumbre de mercado en el flujo de dinero.

Un acercamiento temprano a la importancia de los pronósticos en las organizaciones lo proporcionan Florance M. y Sawicz M. en [4]: "El pronóstico de ventas no es una ciencia exacta, pero saber qué impulsa la demanda de ventas es crucial para el éxito de los pronósticos de ventas". Hoy en día, los pronósticos en los negocios se han convertido en una necesidad para la toma de decisiones, el uso eficaz de los recursos y el éxito del negocio [3].

Los pronósticos deben ser vistos como una herramienta para disminuir la incertidumbre sobre el comportamiento futuro del objeto a estudiar. Al tratar de predecir los patrones futuros de lo estudiado, se presentan varios retos para el científico como lo son: la precisión y las correctas métricas para la evaluación, además del correcto entendimiento del fenómeno o problema analizado [5]. En términos de precisión, se debe entender que al ser un pronóstico, éste se encuentra dentro de intervalos de aceptación que generalmente se obtienen después de un análisis estadístico, en [5] se ejemplifica a esto con un supuesto de precisión de 90 % para un pronóstico de venta que tiene como límite inferior de ventas 50 unidades y como superior 900 unidades. Esto evidentemente no es información útil para el tomador de decisiones ya que el intervalo es muy amplio a tal grado que resulta ineficiente pero ayuda a ejemplificar una parte de la problemática al generar predicciones en los negocios.

Sankaran, G. et al. en [6] define dos tipos básicos de pronósticos: *cualitativos* usualmente basados en factores descriptivos o de calidad y los *cuantitativos* basados en series de tiempo y métodos causales. Al emplear los métodos causales se busca responder a la pregunta *¿por qué?* al entender las interdependencias entre variables que impactan en aquello que se está estudiando o estimando. Por otro lado, en las series de tiempo se busca responder a la pregunta *¿qué?* al analizar su comportamiento y emplear algoritmos que interpretan la serie y la extrapolan en el futuro y se asume que la historia y lo observado en la serie de tiempo se repetirá en alguna otra escala.

En [7] se analiza y discuten las técnicas empleadas por algunas empresas estadounidenses entre las décadas de 1970 y 1980 para la realización de sus pronósticos. Se aprecia el amplio uso de los conocimientos del negocio empleados con algo de ayuda de herramientas estadísticas mostrando además, que los pronósticos combinados con la experiencia del negocio incrementaron la precisión para periodos atípicos. Se puede entender entonces que históricamente, ha sido importante acompañar los datos e información con experiencia y conocimiento previo del negocio. Otro estudio [8] documenta el uso de diversas técnicas de predicción empleadas en compañías estadounidenses a mediados de la década de 1980, en el cual se destacan sus aplicaciones en actividades de planeación, pro-

ducción, mercadeo, logística, entre otras; además se mencionan las principales técnicas de pronóstico empleadas destacando los modelos por computadora, regresión, simulación, econometría, medias móviles y análisis de ciclos de vida. Es importante destacar que gracias a las limitantes tecnológicas de aquel entonces, no era posible cuantificar ni analizar grandes cantidades de datos de soporte para la elaboración de los pronósticos.

2.2. Análisis de series temporales y algoritmos de Machine Learning

Para la elaboración de pronósticos se han desarrollado diferentes herramientas y técnicas estudiadas tales como análisis de series de tiempo en donde se obtienen estadísticas significativas además de otras propiedades de los datos y donde se busca predecir con base en información del pasado [9], pero que puede no resultar como una de las técnicas más adecuadas al considerar la relación entre entradas y salidas como lineal, esto no es algo común en la realidad [10]. Dentro del análisis de series de tiempo se encuentra un modelo particular llamado ARIMA, el cual trabaja buscando las relaciones entre valores y errores pasados [11], el nombre ARIMA se le es otorgado a este modelo gracias a su acrónimo en inglés ya que es autorregresivo (AR), integrado (I) y de promedios o medias móviles (MA del inglés: moving average). Existen algunos trabajos ya realizados sobre predicción o pronósticos con base en este modelo como el realizado por Chen en [12], en donde usa la información disponible de la Asociación China de Manufactureras de Automóviles para realizar y evaluar un pronóstico de demanda de automóviles. Otro ejemplo fue el aplicado en [13], en donde combina un modelo ARIMA con una Red Neuronal Recurrente para pronosticar las ganancias diarias de ventas.

Otro tipo de algoritmos empleados para el pronóstico en el análisis de series de tiempo son las máquinas de soporte vectorial, los cuales pertenecen a la familia de técnicas supervisadas de aprendizaje en el aprendizaje de máquina. Estos algoritmos tienen la particularidad de que muestran un buen desempeño en aquellos casos en donde la relación entre los datos de entrada y salida del algoritmo es desconocida [10]. En el trabajo realizado por Xiangsheng Xie en [14], para el pronóstico de ventas minoristas de la industria restaurantera en China, se empleó un algoritmo de máquinas de soporte vectorial como método de aprendizaje supervisado mostrando un desempeño superior al pronóstico realizado mediante un modelo ARIMA.

Las Redes Neuronales Artificiales son un método de aprendizaje de máquina, en estos algoritmos son usados ampliamente para la realización de pronósticos

de series de tiempo, clasificación, entre otras aplicaciones [13]. Estos algoritmos fueron desarrollados con la intención de simular la estructura cerebral humana de tal manera que las redes neuronales son capaces de aprender cualquier función matemática con ayuda de suficiente información de entrenamiento [15]. Dentro de las redes neuronales existen diferentes tipos de mejoras hechas a estos algoritmos como el empleado por Chen, F. y Ou, T. en [16]. En donde emplean una combinación del aprendizaje de máquina extremo gris y aprendizaje de máquina extremo junto con el método Taguchi para entonces, al resultado de esta combinación emplearlo como datos de entrada en una red neuronal de aprendizaje de máquina extremo para ayudar a la toma de decisiones. El resultado fue comparado contra otros modelos de aprendizaje de máquina de propagación hacia atrás mostrando un desempeño superior.

Otro algoritmo de aprendizaje de máquina es XGBoost, creado por Chen, T. y Guestrin, C. en [17], el cual es un sistema de refuerzo de árboles escalable de extremo a extremo. El algoritmo se propone como consciente de la escasez de datos y genera un esquema cuantitativo ponderado para el aprendizaje aproximado de refuerzo de árboles y está inspirado en la técnica de mejora del árbol de gradientes. Existen trabajos en donde se emplea este algoritmo con éxito en la elaboración de pronósticos de series de tiempo como en la investigación desarrollada por Wang, Y. y Guo, Y. en [18]. Ésta tenía como objetivo realizar pronósticos sobre el precio de apertura para el mercado de valores empleando información del periodo 2015 a 2018 a través de un proceso híbrido que emplea un modelo mejorado del algoritmo XGBoost y que demostró resultados sobresalientes. Otra aplicación de esta técnica es el estudio aplicado a series de tiempo para pronósticos de ventas realizado por Pavlyshenko, B. en [19] en la cual compara la combinación de diferentes modelos predictivos. De igual manera, otro ejemplo de la aplicación del algoritmo XGBoost es el trabajo realizado por Sarafrazi et al en [20], donde lo emplean junto con otras técnicas de aprendizaje de máquina aplicado a la detección temprana de tratamiento para sepsis a través del uso de series de tiempo multivariables con información proveniente de hospitales en Estados Unidos. El modelo XGBoost mostró un gran desempeño al predecir la sepsis 12 horas antes de su diagnóstico.

2.3. Uso combinado de algoritmos de Machine Learning y algoritmos metaheurísticos

Las herramientas descritas en la sección anterior son de las más empleadas hoy en día para la realización de pronósticos, en el caso particular del aprendizaje

de máquina se tienen variables que son ajustadas durante la ejecución de los algoritmos con la intención de mejorar los resultados obtenidos, además algunos algoritmos metaheurísticos han sido usados para optimizar dichas variables. Un ejemplo es el realizado por Zhou, Y., y Zheng, S. en [21], en donde desarrollaron una metodología de optimización al integrar aprendizaje de máquina supervisado junto con un algoritmo avanzado de optimización basado en la enseñanza y el aprendizaje. Liu et al. en [22], propone un nuevo modelo de predicción basado en aprendizaje extremo de máquina y un optimizador inteligente basado en averíos de pollos (chicken swarm) para acelerar la convergencia y los pesos del algoritmo de aprendizaje extremo de máquina y así predecir la potencia fotovoltaica bajo diferentes condiciones climáticas. Ayumi et al. en [23], propone un modelo para la optimización de una Red Neuronal Convolucional a través de un algoritmo de recocido micro canónico para el procesamiento de imágenes. En [24] Jiang, Y. et al. presenta un método basado en un algoritmo genético para optimizar los parámetros de entrenamiento del algoritmo XGBoost para la clasificación usado en la detección de peatones. A pesar de que se han realizado experimentos empleando diversas técnicas para la realización de pronósticos, el principal reto sigue siendo la calidad y tipo de la información disponible. Es en este sentido que deben realizarse análisis y comparaciones de diferentes métodos de predicción para encontrar aquellos que mejores resultados y desempeño muestren de acuerdo a la información y datos históricos obtenidos.

2.4. Algoritmos de Machine Learning y algoritmos metaheurísticos aplicados a series temporales cortas

El problema de pronósticos y análisis de series de tiempo cortas es un tema poco estudiado. En este trabajo se empleará la definición provista en [25] respecto a que una serie temporal corta se entiende como el registro de datos de frecuencia mensual, comprendido en un periodo de menos de tres o cuatro años. En [26] ser realizó un estudio comparativo de la capacidad predictiva entre modelos de suavización exponencial en series de tiempo cortas. Se utilizaron datos sobre ventas de cierto producto y la métrica de comparación fue el error cuadrático (SE) de las predicciones.

En [27] se desarrolló una metodología para corregir el efecto del sesgo en la estimación del componente autorregresivo de una serie de tiempo corta. La metodología propuesta funcionó bien cuando la autocorrelación era menor a 0.5. En [28] se desarrolló un método para la predicción de series de tiempo cortas

basado en técnicas de agrupamiento de series de tiempo y modelos de regresión basados en distancias. Dicho método, aunque sí consideraba la longitud de las series de tiempo como un problema que influye en la precisión de las predicciones, el método demanda contar con una enorme cantidad de series de tiempo cortas parecidas en el mismo conjunto de datos para realizar las predicciones. La metodología propuesta fue compara con los métodos de suavización exponencial.

El único trabajo detectado en la literatura sobre series de tiempo cortas utilizando herramientas de inteligencia artificial es el realizado por [29]. En dicho trabajo se propone un modelo de Autocorrelación Bayesiana donde la información a-priori es obtenida mediante un algoritmo de inteligencia artificial. La metodología mostró resultados prometedores.

Capítulo 3

Marco teórico

3.1. Inteligencia de Negocios

En el mundo de los negocios, el tomador de decisiones usualmente busca una salida o resultado satisfactorio antes que encontrar la mejor solución posible e históricamente esto tiene un costo oculto que puede ocasionar el fracaso del negocio, pero el objetivo de una estrategia de negocio es y debe ser un crecimiento sustentable [30]. Para asegurar el máximo aprovechamiento de una estrategia de negocio, el intercambio de información entre todos los interesados es de vital importancia, es así como la dinámica de trabajo hoy en día en las organizaciones a nivel mundial se caracteriza por su gran y rápida adopción de sistemas informáticos para el control y administración de sus operaciones. Tal es el caso de los sistemas de planificación de recursos empresariales (ERP por sus siglas en inglés), que de acuerdo con la décimo quinta edición del diccionario APICS, un ERP se puede entender como una herramienta para organizar, definir y estandarizar los procesos de negocio de tal manera que se pueda lograr una mejor organización y control de la empresa con la intención de lograr mejoras competitivas [31]. La adopción de este tipo de sistemas ha permitido a las organizaciones recolectar, y en algunos casos procesar, la información necesaria para el funcionamiento del negocio. Es aquí en donde adquiere relevancia el análisis de los datos, ya que se debe buscar y extraer todo el valor agregado oculto para generar ventajas competitivas.

Los principales retos para la correcta elaboración de pronósticos y la toma de decisiones son la recolección, extracción, transformación y carga de los datos a los sistemas inteligentes o de negocios con las habilidades de procesamiento de los datos. Para reducir los riesgos inherentes al manejo de los datos, las organizaciones de todo el mundo están empleando, en la medida de sus posibilidades y en ocasiones junto con su sistema ERP, sistemas computacionales llamados *sistemas de inteligencia de negocios* comúnmente llamados BI gracias a sus siglas en inglés (Business Intelligence) los cuales permiten después de cierto tiempo,

analizar y procesar los datos recolectados con el objetivo de mejorar la calidad de las decisiones a tomar así como el desempeño general del negocio [32], es decir, mejorar la efectividad de los procesos estratégicos de la organización.

En [30] se mencionan cuatro aspectos principales a entender al implementar una solución de BI, los cuales, junto con otros factores externos e internos de la organización, determinan las estrategias a elegir: clientes, costos, competidores y competencias. Éstos son los determinantes que permiten a los directivos entender e interpretar la información, por lo que cualquier solución de BI debe tomarlos en consideración siempre teniendo en cuenta que las decisiones estratégicas de las empresas usualmente se basan en alguno de los dos escenarios:

- Mejorar la situación del negocio actual para tener un máximo crecimiento a un menor riesgo.
- Cambiar la situación el negocio actual ya que representa crecimiento negativo, riesgos elevados o desastres financieros.

El directivo debe entonces buscar una estrategia que le genere ventajas competitivas tomando en cuenta principalmente las necesidades de sus clientes, así como las de todos los interesados y enfocarse entonces en la solución de alguno de los dos puntos anteriores. Las implementaciones de BI deben ser más que sólo presentar gráficas, deben proveer un contexto que haga sentido con los números y que complemente con información una mejor toma de decisiones. Se tienen ejemplos claros de la implementación de de estrategias de BI en operaciones empresariales, éstos son tan variados que pueden ir desde administración automática de inventarios la cual permite que los consumos y suministros del mismo sean automáticas basándose en algoritmos que toman en consideración el uso actual, predicciones y ritmos de producción; hasta pronósticos de la demanda que emplean técnicas de análisis histórico de datos y lo combinan con datos de otras fuentes permitiendo así una mejor planeación de manera estructurada [30].

Con la mejora y disponibilidad de recursos tecnológicos que permiten a las organizaciones recolectar y analizar sus datos para transformarlos en información, los pronósticos de venta y demanda han tomado un papel fundamental para impulsar el valor real dentro de la cadena de suministro [33]. El análisis predictivo hoy en día se está empleando para encontrar patrones en el comportamiento de consumidores, medir la efectividad de la estrategia de mercadotecnia y optimizar el desempeño financiero con el objetivo de prepararse y poder responder a las necesidades y demandas de los clientes. Algunas compañías como Nestlé, DOW, Cisco y Procter & Gamble están ajustando sus procesos hacia un enfoque pragmático en el uso de análisis de datos para mejorar la precisión de sus pronósticos de

demanda e impactar de manera positiva en el servicio al cliente y en la reducción de inventarios [33].

3.1.1. El rol de los pronósticos en el control de la demanda

Gran parte del éxito de las empresas radica en su capacidad de administrar el flujo de materiales, información y efectivo dentro de la misma y con todos los integrantes del proceso productivo, a esto se le suele llamar: *cadena de suministros*. Un mal manejo o administración de dicha cadena suele tener como consecuencia problemas como altos costos de inventario, bajo nivel de satisfacción del cliente, ventas perdidas, entre otros [34]. Al analizar dicha cadena, se pueden tomar decisiones para mejorar e incrementar las utilidades de la organización, por ejemplo, se puede determinar si algún proceso se puede tercerizar, si es que algún almacén ofrece un costo más bajo, e incluso qué medios de transporte permiten reducir tiempos y costos de entrega.

Si bien los sistemas ERP permiten tener información en tiempo real de la operación en la organización, su enfoque está pensado para las áreas financieras. Muchos de estos sistemas implementan diversos módulos para el resto de los departamentos como los MRP (Material Resource Planning) o planeación de los requerimientos de materias. Dicho módulo descompone los productos finales en sus componentes, rutas de producción, materia prima y lista de materiales como se muestra en la figura 3.1 y está pensado originalmente para administrar el flujo de materiales, componentes y otros recursos [35]. Se puede destacar la importancia del pronóstico de ventas como información de entrada directa para la administración de la demanda, esto sucede porque al analizar la información histórica y presente de las ventas se puede obtener información relevante como la estacionalidad, tendencias e incluso datos atípicos que periten a la organización adaptarse y flexibilizar su capacidad productiva de acuerdo a las necesidades y comportamientos de los clientes [30].

Es a partir de estos pronósticos que se crea un plan maestro de producción, mismo que determina el ritmo y tipo de trabajo de toda la organización. Tales tipos de trabajo se clasifican comúnmente en dos tipos: *pull* y *push*, el primero se puede entender como la metodología Kanban mientras que el segundo se le conoce como metodología MRP. Aunque el objetivo es compartido, la forma de desarrollar el trabajo es diferente en términos de tiempos, trabajo en proceso y materiales en piso. Una estrategia Kanban es se basa principalmente en el reaprovisionamiento, diseñado para la producción de productos con alta demanda y ambientes productivos de múltiples productos, requiere además una cantidad mínima de inventario para cada producto al final de cada estación de trabajo.

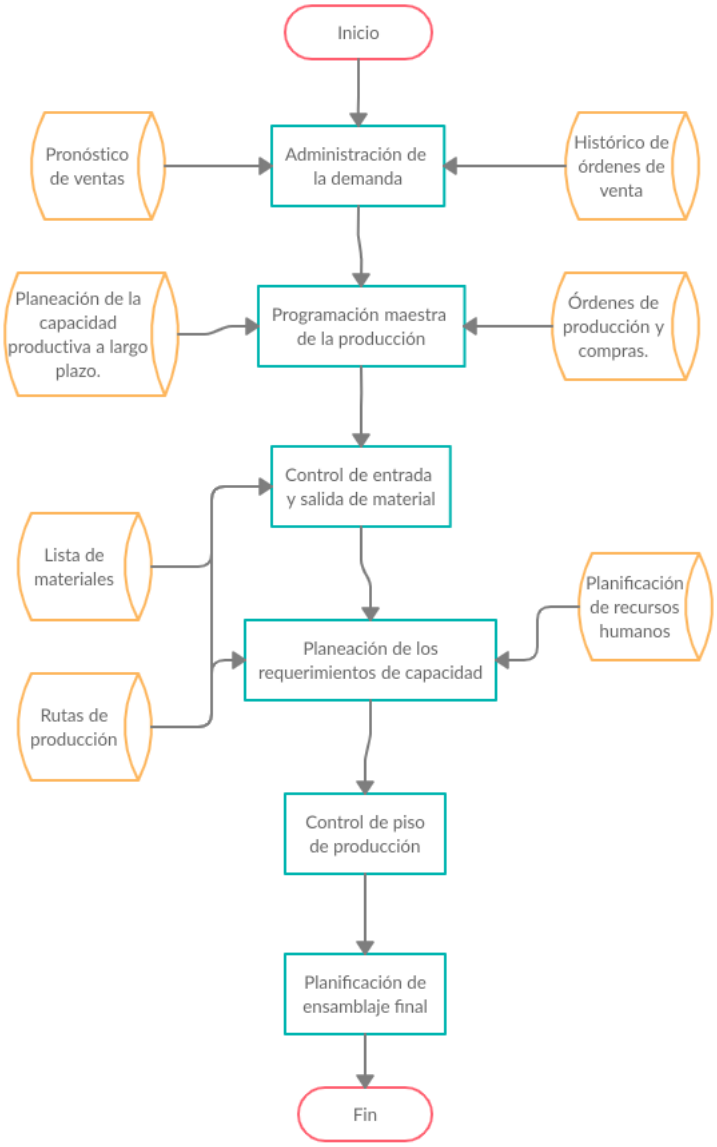


Figura 3.1: Diagrama del flujo de información en un MRP. Fuente: elaboración propia.

Cuando una unidad de material de inventario es consumida por una estación de trabajo, se envía una señal a la estación anterior para volver a surtir el material consumido, esto puede generar grandes tiempos de trabajo en proceso y estaciones detenidas cuando el producto final depende de múltiples procesos [36].

Por otro lado, en la estrategia MRP tiene un funcionamiento basado en tiempos de liberación de trabajos en cada estación. Dichos tiempos se obtienen de

la programación hacia atrás de las fechas de vencimiento de los pedidos en las estaciones de producción utilizando los plazos planificados en cada una de estas. Para controlar la diferencia entre el tiempo de entrega planificado y el flujo real de una orden, esta estrategia incorpora con frecuencia inventario de seguridad o plazos de entrega seguros con la intención de no tener retrasos en su producción. El riesgo en este esquema de trabajo es que se tendrán tiempos altos para la terminación de un producto además de costos elevados en inventario dado el esquema de seguridad [36].

Como conclusión, el contar con un adecuado pronóstico de ventas permitirá a la organización reducir costos en áreas críticas como almacén, producción, recursos humanos, entre otros. Además, la empresa será capaz entonces de realizar una mejor planificación de sus recursos con la intención de satisfacer las necesidades de sus clientes y el cumplimiento de los objetivos en términos de costos fijados por la dirección.

3.2. Análisis de series de tiempo

Una serie de tiempo es comúnmente, un conjunto ordenado de observaciones sobre una misma variable almacenado en un específico periodo de tiempo, en intervalos igualmente espaciados (también conocido como un *proceso estocástico*) y se pueden encontrar aplicaciones en ingeniería, ciencias sociales, economía, entre otras. De acuerdo con Shumway, R. y Stoffer, D. en [37], el enfoque sistemático en el que se busca, mediante herramientas matemáticas y estadísticas, responder a las preguntas sobre las correlaciones existentes en los datos, es a lo que se le llama: análisis de series de tiempo o series temporales. Según los mismos autores, existen típicamente dos enfoques en el análisis de series temporales: por tiempo y frecuencia. En el primero se estudian las relaciones rezagadas o retardadas asumiendo que lo que suceda hoy afecta a lo que suceda mañana. En el segundo se estudian las variaciones o cambios en los datos de acuerdo a ciclos de tiempo por ejemplo, el comportamiento de los ciclos económicos en periodos de crecimiento o recesión.

Usualmente, una serie temporal se representa de manera gráfica al colocar los valores de las variables observadas de acuerdo al eje vertical mientras, que en el horizontal, se colocan los espacios de tiempo en los cuales se obtuvieron los valores mencionados. La figura 3.2 es un ejemplo de una serie de tiempo, en este caso se grafican cien valores $\{x_t\} \sim \text{iid } N(\mu = 0, \sigma = 1)$ es decir, que son aleatorios, independientes e idénticamente distribuidos de acuerdo a una distribución normal con media = 0 y desviación estándar = 1.

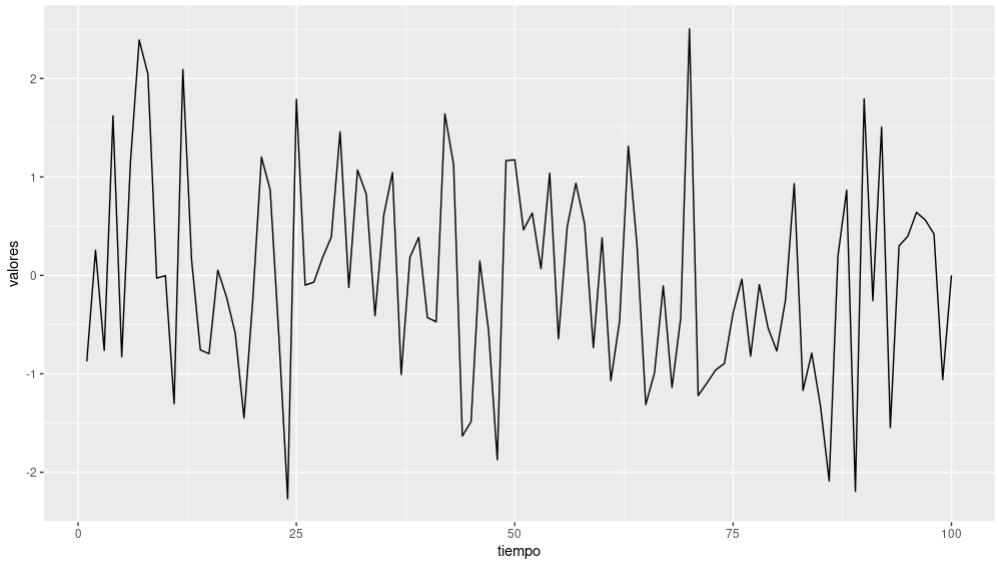


Figura 3.2: Ejemplo de serie temporal.

Shumway, R. y Stroffer, D en [37], describen algunas de las propiedades de las series temporales. Se dice que la función de distribución conjunta puede describir una serie de tiempo entendida como la colección de n variables aleatorias en puntos de tiempo $t_1, t_2, ..., t_n$, para cualquier entero positivo n evaluada como la probabilidad de que los valores de la serie son en conjunto, menores que las n constantes $c_1, c_2, ...c_n$:

$$F_{t_1,t_2,...t_n}(c_1, c_2, ..., c_n) = \mathrm{P}(x_{t_1} \leq c_1, x_{t_2} \leq c_2, ..., x_{t_n} \leq c_n) \tag{3.1}$$

A pesar de que dicha función de distribución conjunta puede describir toda la serie temporal en términos de sus datos, es una herramienta bastante difícil de emplear para analizar los datos de la serie ya que se necesitaría que todas las variables aleatorias sean conjuntamente normales además, la función 3.1 debe evaluarse en función de n argumentos. La función de distribución

$$F_t(x) = \mathrm{P}\{x_t \leq x\}$$

y su correspondiente función de densidad

$$f_t(x) = \frac{\partial F_t(x)}{\partial x},$$

son usualmente informativas cuando existen y describen de manera superficial el comportamiento de la serie.

La función media también es medianamente útil para tratar de entender el comportamiento de una serie de tiempo, ésta se define como

$$\mu_{xt} = \mathrm{E}(x_t) = \int_{-\infty}^{\infty} x f_t(x) dx \tag{3.2}$$

Para determinar si existe independencia entre dos valores adyacentes x_s y x_t, se pueden emplear las nociones de covarianza y correlación. La función de autocovarianza se define entonces como:

$$\gamma_x(s,t) = \mathrm{cov}(x_s, x_t) = \mathrm{E}[(x_s - \mu_s)(x_t - \mu_t)] \tag{3.3}$$

y mide o determina la dependencia lineal entre dos puntos en la misma serie observados en diferentes tiempos. Si $\gamma_x(s,t) = 0$, x_s y x_t entonces no están linealmente relacionados, pero aún puede existir cierta dependencia o relación no lineal entre ellos. Por otro lado, si x_s y x_t son normalmente bivariados, $\gamma_x(s,t) = 0$ asegura su independencia. Si $s = t$ la autocovarianza se reduce a varianza.

La función de autocorrelación mide la correlación entre dos variables separadas por $s - t$ periodos. Dicha función se define como:

$$\rho(s,t) = \frac{\gamma(s,t)}{\sqrt{\gamma(s,s)\gamma(t,t)}} \tag{3.4}$$

Si x_t es perfectamente predecible desde x_s a través de una relación lineal $x_t = \beta_0 + \beta_1 x_s$, entonces la correlación será positiva si $\beta_1 > 0$, y negativa en caso contrario.

3.2.1. Series de tiempo estacionarias

Se dice que una serie temporal estacionaria se caracteriza porque sus propiedades no dependen del tiempo en el que la misma serie es observada. En otras palabras, según Shumway, R. y Stroffer, D. en [37], una serie temporal **estrictamente estacionaria** se caracteriza porque el comportamiento probabilístico de cada conjunto de valores $\{x_{t1}, x_{t2}, ..., x_{tk}\}$ es idéntica a una serie en la cual el tiempo está desplazado h veces: $\{x_{t1+h}, x_{t2+h}, ..., x_{tk+h}\}$. Realizando formalmente la definición de una serie temporal estacionaria:

$$\mathrm{P}(x_{t1} \leq c_1, ..., x_{tk} \leq c_k) = \mathrm{P}(x_{t1+h} \leq c_1, ..., x_{tk+h} \leq c_k) \tag{3.5}$$

para todas los $k = 1, 2, ...$, todos los puntos temporales $t_1, t_2, ..., t_k$, todos los números $c_1, c_2, ..., c_k$ y todos los desplazamientos $h = 0, \pm 1, \pm 2, ...$

Dado que la función de media $E(x_t) = \mu_t$, de una serie temporal estacionaria es independiente del tiempo [37], se puede simplificar la notación de tal suerte que

$$\mu_t = \mu. \tag{3.6}$$

Debido a que la función de autocovarianza $\gamma(s,t)$, de una serie temporal x_t depende de la diferencia $|s-t|$, se define $s = t+h$, en donde h es el desplazamiento temporal o *lag*. Entonces, para definir la función de autocovarianza para las series temporales estacionarias, se recupera la definición de la ecuación 3.3, quedando entonces como:

$$\gamma(t+h,t) = \text{cov}(x_{t+h}, x_t) = \text{cov}(x_h, x_0) = \gamma(h,0)$$

Ya que la diferencia temporal entre los periodos $t+h$ y t es la misma que la diferencia entre los periodos h y 0, la función de autocovarianza de uan serie de tiempo estacionaria no depende del argumento temporal t [37]. La función de autocovarianza de una serie temporal estacionaria se define como:

$$\gamma(h) = \text{cov}(x_{t+h}, x_t) = \text{E}[(x_{t+h} - \mu)(x_t - \mu)]. \tag{3.7}$$

La definición de la función de autocorelación de una serie temporal estacionaria, tomando en consideración la definición establecida en la ecuación 3.4, es:

$$\rho(h) = \frac{\gamma(t+h,t)}{\sqrt{\gamma((t+h,t)+h)\gamma(t,t)}} = \frac{\gamma(h)}{\gamma(0)}. \tag{3.8}$$

3.2.2. Modelo Autorregresivo Integrado de Medias Móviles (ARIMA)

El pronóstico de series temporales llamado ARIMA (del inglés *Auto Regressive Integrated Moving Average*) es una clase de modelos que predicen el comportamiento de una serie de tiempo con base en patrones o comportamientos del pasado [37]. En [38] se establece que un modelo ARIMA contiene tres parámetros a ajustar:

- p: orden de la parte autorregresiva (AR).Es el número de términos desconocidos que multiplican la serie temporal en tiempos pasados en tantas veces como sea el valor de p.

- d: base o grado de diferenciación (I) de la serie temporal. Es la cantidad de veces que se debe diferenciar la serie de tiempo para convertirla en estacionaria.
- q: parámetro u orden de los modelos de medias móviles (MA). Es el número de términos desconocidos que multiplican los errores de pronósticos en tiempos pasados en tantas veces como sea q.

Una serie temporal estacionaria puede ser descrita como un modelo ARMA (autorregresivo de medias móviles) de orden (p, q), dicho modelo se representa como:

$$y_t = a_1 y_{t-1} + a_2 y_{t-2} + ... + a_p y_{t-p} + b_1 e_{t-1} + ... + b_q e_{t-q} + e_t \tag{3.9}$$

en donde y_t representa los datos a modelar, $y_{t-1}, y_{t-2}...$, son datos en instantes de tiempo pasado; $e_{t-1}, e_{t-2}...$, son errores en periodos de tiempo pasado y e_t es el error actual. Aquí es importante resaltar que los modelos ARMA asumen el error como normalmente distribuido [39].

Un modelo ARIMA es entonces, un modelo ARMA diferenciado en un orden d, aplicado en la serie temporal original con la intención de hacerla estacionaria. Dicho modelo ARIMA se compone de tres principales pasos [39]:

1. **Hacer la serie temporal estacionaria:** Dado que no todas las series de tiempo son estacionarias, un modelo ARMA no se puede ajustar. Para solucionar este inconveniente, se le aplica una operación de diferenciación d-veces.
2. **Identificar los valores adecuados para el orden del modelo:** Las funciones de autocorrelación y autocorrelación parcial son aplicadas a los datos diferenciados, esto con la intención de encontrar los valores (p, q). Si la función de autocorrelación tiene un decrecimiento sinusoidal al mismo tiempo que la función de autocorrelación parcial se hace 0 tras una ventana de tiempo p, entonces el modelo es Autorregresivo (AR) de orden p. Por otro lado, si la función de autocorrelación se hace 0 tras una ventana temporal q, y la función de autocorrelación parcial tiene una caída sinusoidal, entonces el modelo se llama de Medias Móviles (MA) de orden q. Finalmente, si las funciones de autocorrelación y autocorrelación parcial tienen un decrecimiento sinusoidal, entonces se trata de un modelo ARIMA de orden (p, d, q).
3. **Predecir de la serie temporal:** Una vez obtenidos los parámetros (p, d, q), los parámetros $a_1, a_2, ..., a_p, b_1, b_2, ..., b_q$ son encontrados. Al mismo tiempo, la varianza de e_t puede ser obtenida y la serie temporal puede ser pronosticada empleando métodos como mínimos cuadrados y máxima verosimilitud.

3.2.3. Descomposición STL

La descomposición STL (del inglés *Seasonal-Trend decomposition using LOESS*) es un procedimiento de descomposición de tendencia estacional basado en LOESS (del inglés *LOcally Estimated Scatterplot Smoothing*, que quiere decir: Suavizado de Gráficos de Dispersión Estimado Localmente). Es una técnica de descomposición de series temporales desarrollada por Cleveland, *et al.* en [40].

Esta técnica se caracteriza por descomponer una serie temporal en sus componentes de tendencia, estacionalidad y residuos. Su diseño y simplicidad permiten la ejecución de esta técnica a un bajo costo computacional en series temporales largas y cortas. Además, permite la descomposición sin afectación por presencia de datos atípicos [40], dichos datos se ven reflejados en el componente residual del STL. La figura 3.3 muestra un ejemplo de la descomposición STL, se observa en la parte superior la serie temporal original; en la segunda sección se encuentra la tendencia, seguida de la estacionalidad y los residuales en la parte inferior. Gracias a la descomposición STL se puede apreciar el comportamiento de los componentes de la serie temporal, permitiendo la realización de un rápido análisis visual.

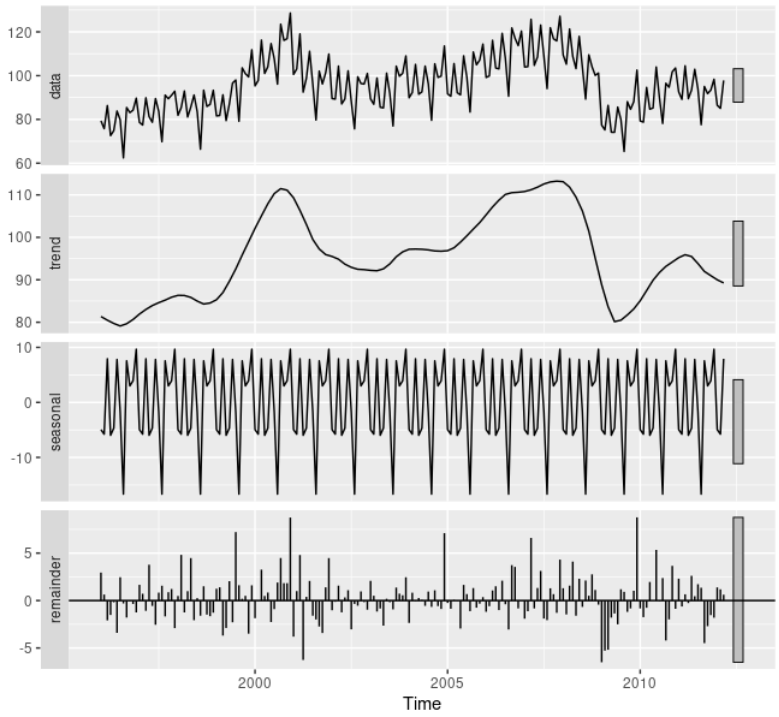


Figura 3.3: Ejemplo de descomposición STL.

3.3. Aprendizaje de Máquina (ML)

El aprendizaje de máquina es una rama de la inteligencia artificial que tiene como objetivo el desarrollo de técnicas y modelos que permiten a las computadoras aprender. En sus inicios a mediados de la década de 1950 los primeros esfuerzos de aprendizaje por computadoras buscaban generar modelos de auto organización, auto estabilización e identificación de formas [41]. A pesar de que el concepto de ML tiene más de medio siglo de existencia, es hasta años recientes que comienza a tener un mayor desarrollo y uso en gran medida gracias a las capacidades computacionales existentes hoy en día.

Al igual que como ocurre con los seres vivos principalmente humanos, los algoritmos de ML permiten a una computadora aprender mediante la experiencia. A través de ellos, los datos consumidos por la computadora se transforman en modelos o programas sin que estos últimos sean escritos por algún científico o desarrollador. Los algoritmos de ML permiten a las computadoras resolver problemas difíciles de programar como una secuencia de instrucciones [42]. De la mano con estos poderosos algoritmos se encuentran los datos, que son pieza angular para su funcionamiento y correcta implementación. Como en el caso de los seres vivos, que aprenden a través de los sentidos, las técnicas de ML deben ser alimentadas por los mejores datos disponibles, y dependen fundamentalmente del fenómeno a estudiar. Runkler, T. en [43] establece cuatro fases de los proyectos de análisis de datos:

1. Preparación: Involucra cuatro actividades esenciales en las que destacan la planeación del proyecto de recolección de datos, la correcta recolección, generación de características y selección de los datos.

2. Preprocesamiento: Engloba a la limpieza, filtrado, terminación, corrección, estandarización y transformación.

3. Análisis: Mediante la visualización, correlación, regresión, pronóstico, clasificación y agrupamiento.

4. Postprocesamiento: La etapa final que integra la interpretación, documentación y evaluación del proyecto.

Para iniciar con una correcta implementación de ML debe comenzar con la preparación de los datos, su preprocesamiento y tener su punto protagónico en el análisis y es justamente en este punto en donde se ubica la mayor parte de la literatura.

3.3.1. Aprendizaje estadístico

A principios del siglo XIX, Gauss y Legendre publicaron una serie de artículos sobre el método de mínimos cuadrados que sienta las bases de lo que hoy en día se conoce como Regresión Lineal. Ésta técnica fue aplicada inicial y exitosamente a problemas astronómicos. Hoy en día es usada ampliamente para predecir valores cuantitativos como el salario de una persona; en predicción de valores cualitativos como si un paciente sobrevive no, tendencia del precio de acciones en los mercados bursátiles, entre otros [44]. Fue hasta inicio de la década de 1970 cuando Nelder y Wedderbun definen el término *modelos lineales generalizados* para referirse a los métodos de aprendizaje estadístico que incluyen a la regresión lineal y logística.

Para la década de 1980, la tecnología computacional disponible permitió el desarrollo de nuevas técnicas de aprendizaje y se removió la barrera tecnológica que restringía la implementación de modelos únicamente lineales. Aparecieron entonces modelos como los árboles de regresión y clasificación, entre otros. Desde entonces, fueron creadas disciplinas como el ML y otras más que han posicionado al aprendizaje estadístico como un sub campo de la estadística y están principalmente enfocados en modelado y predicción supervisada y no supervisada.

James en [44] define al aprendizaje estadístico como un amplio conjunto de herramientas para el entendimiento de los datos. Estas herramientas suelen ser clasificadas como supervisadas o no supervisadas, y difieren entre sí en cuestión de que las herramientas supervisadas involucran la construcción de modelos estadísticos para predecir o estimar una salida basada en una o más entradas de datos, y el resultado se valida contra información previa de lo que se está estimando, es decir ya se sabe el resultado. Por otro lado, en el esquema no supervisado, existen entradas pero no salidas supervisadas o revisadas, pero es posible encontrar relaciones o estructuras en sus datos.

3.3.2. Transformación de series temporales a aprendizaje supervisado

En el uso de algoritmos de ML para el pronóstico de series temporales se debe considerar el manejo de los datos para poder usar dichos algoritmos. El propósito de la transformación de series temporales a un problema de aprendizaje supervisado permite el uso de algoritmos de ML que se pueden ajustar de manera lineal o no lineal. Se le llama aprendizaje supervisado al proceso de usar un algoritmo de ML para aprender la función de mapeo que relaciona las variables de entrada (X) y la variable de salida (y) de tal suerte que se puede expresar como:

$$y = f(X) \tag{3.10}$$

Una vez obtenida la función de mapeo o función que modela el comportamiento, se puden predecir las variables (y) de salida con base en la información de entrada (X). El proceso mediante el cual un algoritmo aprende de los datos de entrenamiento, se le llama *aprendizaje supervisado*. Al conocer las respuestas correctas, el algoritmo realiza predicciones de manera iterativa sobre el conjunto de entrenamiento y se corrige mediante la reducción o ampliación de sus métricas de evaluación. Los problemas de aplicación de los algoritmos de aprendizaje supervisado se dividen en:

- Clasificación: cuando la salida de un problema es una variable categórica, como un color o un grupo.
- Regresión: cuando la variable de salida es un valor real, como alguna unidad de medición.

Para el ajuste de series temporales a un problema de ML, una de las técnicas más comunes es la ventana corrediza, del inglés *Sliding Window*. Dada una secuencia de números o valores del conjunto de datos de la serie temporal (ver tabla 3.1), se debe reestructurar la información usando pasos previos en el tiempo, como variables de entrada y, emplear el siguiente paso temporal, como la variable de salida.

Tiempo	**Valor**
1	200
2	210
3	208
4	215
5	220

Tabla 3.1: Ejemplo del conjunto de datos de una serie temporal.

En la tabla 3.2 se muestra el resultado de aplicar la técnica de ventana corrediza a la serie temporal. Se observa que el paso de tiempo anterior es la variable de entrada (X), y el paso siguiente en el tiempo es la salida (y). Además, el orden entre las observaciones se mantiene y debe conservarse para entrenar el modelo de aprendizaje supervisado.

X	y
?	200
200	210
210	208
208	215
215	220
220	?

Tabla 3.2: Ejemplo de la serie temporal modificada para aprendizaje supervisado.

3.3.3. Regresión por Vectores de Soporte (SVR)

Se le llama Regresión por Vectores de Soporte (SVR del inglés *Support Vector Regression*) a una máquina de soporte vectorial (SVM del inglés *Support Vector Machine*) aplicado a problemas de regresión [45] y fue desarrollado por Vladimir Vaptnik en los laboratorios de AT&T en 1997 [46]. Los métodos de SVR han sido aplicados a resolver problemas y han mostrado ser un buen modelo predictivo, sin embargo, la elección de sus parámetros como en todos los algoritmos de ML han resultado ser una ardua tarea y deben ser cuidadosamente estudiadas. El SVR tiene fundamentalmente, los mismos principios que el SVM aplicado en clasificación. Debido a que la salida del SVR es un número real, resulta complicado predecir a partir de la información disponible ya que se tienen infinitas posibilidades. Sin embargo, la idea principal es la misma: minimizar el error, caracterizar el hiperplano que maximiza el margen tomando en cuenta la tolerancia del error.

El concepto del SVR, también descrito y definido en [45] es que de manera no lineal, el conjunto de datos original x_i es asignado a un espacio de alta dimensión. Dado el conjunto $\{(x_1, y_1), ..., (x_n, y_n)\}$, en donde x_i es el vector de entrada, y y_i es el valor de salida asociado a él, se puede definir la función de regresión del SVR como la ecuación 3.11:

$$f(x) = w \times \Phi(x) + b \tag{3.11}$$

en donde $\Phi(x)$ representa la función no lineal de asignación, w es el vector de pesos y b es el término de sesgo. El objetivo ahora es encontrar una función $f(x)$ tal que tenga la mayoría de las ε desviaciones de los objetivos y_i para todos los datos de entrenamiento y al mismo tiempo, sea lo más plano posible. Se introduce también la función de pérdida ε-insensible para asegurar la precisión de la solución, en otras palabras, el número de vectores de soporte incrementa más lentamente que de manera lineal cuando el tamaño del problema aumenta. Se define entonces la ecuación 3.12:

$$L_\varepsilon(y_i, f(x_i)) = \begin{cases} |y_i - f(x_i)| - \varepsilon, & |y_i - f(x_i)| \geq \varepsilon \\ 0 & \text{en otro caso} \end{cases} \tag{3.12}$$

en donde la pérdida es igual a cero si el error de los valores pronosticados es menor que ε, de otro modo el valor de la pérdida es igual a un valor mayor que ε.

Al igual que con el problema de clasificación, se pueden introducir variables de holgura no negativas, ξ_i y ξ^*, para representar la distancia desde los valores reales hasta los valores límite correspondientes del ε-tubo. Entonces, la forma restringida se puede formular como:

$$\text{mín}\, \frac{1}{2}\|w\|^2 + C(\xi_i + \xi_i^*) \tag{3.13}$$

sujeto a:

$$y_i - [w \times \Phi(x)] - b \leq \varepsilon + \xi_i$$
$$[w \times \Phi(x)] + b - y_i \leq \varepsilon + \xi_i^*$$
$$\xi_i, \xi_i^* \geq 0$$

en donde C es una función costo que mide el riesgo empírico. La figura 3.4 muestra el SVR aplicado a un conjunto de datos unidimensional en una regresión lineal.

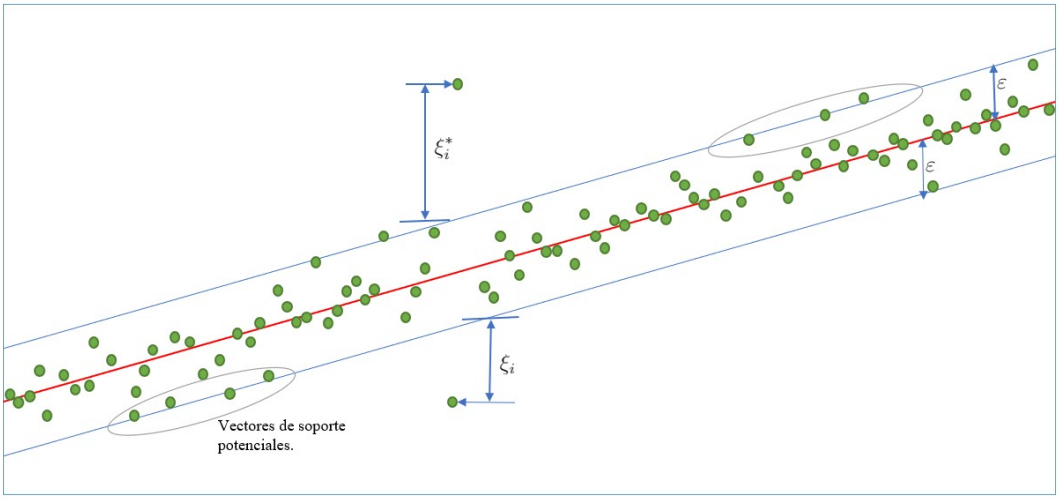


Figura 3.4: SVR lineal unidimensional.

Finalmente, el problema de optimización restringido se resuelve usando la

forma lagrangiana:

$$\sum_{i=1}^{l} y_i(\alpha_i - \alpha_i^*) - \varepsilon \sum_{i=1}^{l}(\alpha_i - \alpha_i^*) - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_{ij}^*)K(x_i, x_j) \quad (3.14)$$

sujeto a:

$$\sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0$$

$$\alpha_i, \alpha_i^* \in [0, C]$$

donde α_i y α_i^* son multiplicadores lagrangianos.

$K(x_i, x_j) = \varphi(x_i)\varphi(x_j)$ es una función kernel. Al usarla, es posible calcular el SVR sin necesidad de asignar de manera explícita en el espacio de características. La función kernel es útil para hallar un hiperplano en un espacio dimensional mayor sin incrementar el costo computacional (ver Fig. 3.5). Evidentemente, el costo computacional incrementa si la dimensionalidad de los datos también lo hace, sin embargo, este incremento dimensional es necesario cuando no es posible encontrar un hiperplano funcional en la dimensionalidad actual. Las condiciones de elegibilidad de un kernel deben estar dentro de la condición de Mercer, que permite que las sustituciones del kernel representen productos punto en algún espacio de Hilbert. Los SVM construídos por la función de base radial gaussiana (RBF): $K(x_i, x_j) = \exp(-\|x_i - x_j\|/2\sigma^2)$ tienen un excelente desempeño en pronósticos no lineales [45]. Por lo tanto, y empleando la función kernel RBF, La ecuación 3.11 puede entonces reescribirse como:

$$w = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)x_i, \text{ y por tanto:}$$

$$f(x) = \sum(\alpha_i - \alpha_i^*)K(x_i, x_j) + b \quad (3.15)$$

Queda entonces por definir los parámetros C, σ y ε los cuales influyen de manera significativa en el comportamiento del SVR y su precisión predictiva, es en este punto donde las técnicas metaheurísticas deberán entrar en juego para encontrar los mejores valores posibles.

3.3.4. Redes neuronales

Los orígenes de las redes neuronales (NN del inglés *Neural Networks*) datan de la década de 1950 [47]. Han tenido gran auge debido a la disponibilidad de

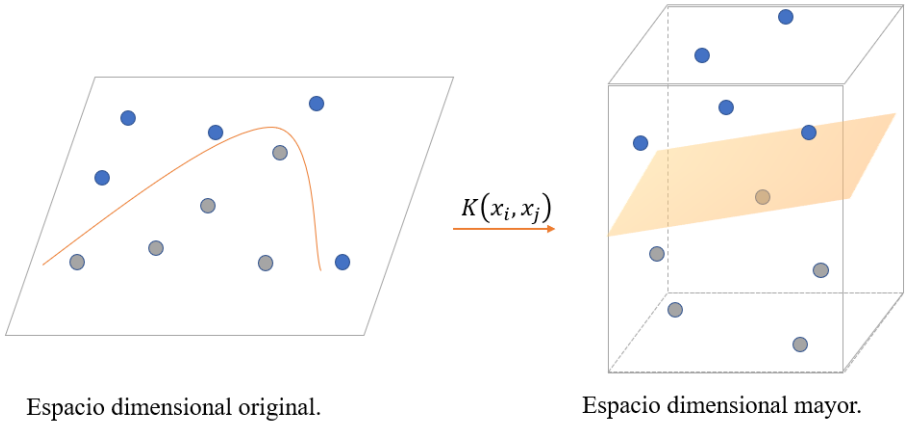


Figura 3.5: Efecto de la aplicación de la función kernel.

mayores y mejores recursos computacionales de la actualidad y, según Aggarwal en [15], son una técnica popular de ML que simula un mecanismo de aprendizaje como el de los seres vivos. Específicamente, las redes neuronales tienen como base el funcionamiento de las neuronas biológicas como las humanas, las cuales están conectadas entre sí y su función principal es recibir, procesar y transmitir información a través de señales químicas y eléctricas. Las neuronas perciben dichas señales a través de las dendritas, que son las encargadas de transmitirlas hacia el núcleo celular para su procesamiento y, de ser necesario, dicha neurona se puede conectar con otra mediante una prolongación llamada axón.

En el caso de las NN, cada neurona se considera una unidad computacional conectada a otra y sus valores son ajustados mediante pesos (w_n), éstos a su vez influyen en el cómputo matemático a realizar en la neurona destino. Es decir, cada neurona tiene como entradas los valores modificados por los pesos en cada una de sus dendritas. Posteriormente, dicha información es procesada en el núcleo neuronal para finalmente, mediante una función matemática, ser enviada a otra neurona mediante su salida en el axón (y) como se muestra en la figura 3.6.

El proceso de aprendizaje ocurre mediante los ajustes de los pesos durante la interacción de las neuronas, y es derivado de la información de entrenamiento como entrada de la NN en forma de entradas-salida. Dicha información es usada para inferir y hacer predicciones sobre las salidas ($\hat{y}$). Los datos de entrenamiento sirven también como parámetros de retroalimentación en el refinamiento de los pesos, dependiendo de qué tan precisa sea la predicción realizada. En otras palabras, y según lo establecido por da Silva et. al. [48], el proceso de entrenamiento de una red neuronal consiste en seguir los pasos correctos para el ajuste de los pesos y límites de las neuronas con la intención de generalizar las soluciones pro-

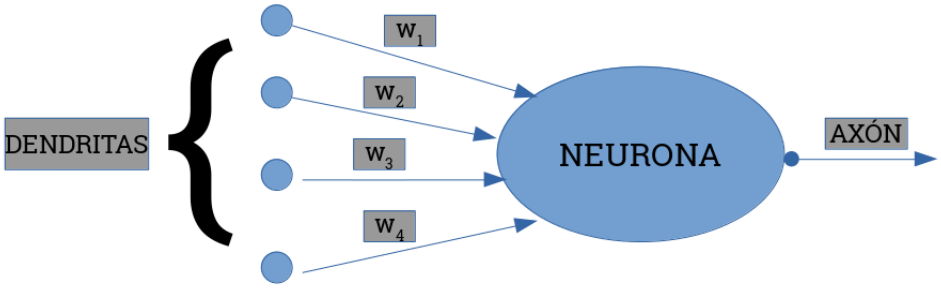


Figura 3.6: Neurona como unidad computacional.

ducidas en la salida de la NN. Similar al proceso cognitivo humano de prueba y error en el que por ejemplo se aprende a no tocar superficies calientes al haberlas tocado previamente, en las neuronas se ajustaron ciertas conexiones neuronales para que ese conocimiento sirva en la prevención de lesiones futuras.

La principal limitante de las NN es la disponibilidad y calidad de información de entrenamiento [15], ya que es necesario disponer de una cantidad suficiente de datos para que la NN pueda apreder y ajustar los pesos de mejor manera. La principal ventaja de las NN contra otros algoritmos de ML, radica en su capacidad de combinar y conectar las neuronas entre sí creando una red de unidades computacionales con la habilidad de aprender por sí mismas.

Red neuronal de una capa

El funcionamiento de una red neuronal de una capa SLP (del inglés: *single-layer perceptron*) es descrita por Y. Hu y J. Tsai en [49] como una red de una capa de alimentación hacia adelante. Se supone un conjunto de datos de la forma $\{(x_1, x_2, ..., x_n)(y_i)\}$, en donde y_i es la salida actual del valor correspondiente en el conjunto de x, mientras que n es la cantidad de características de entrada. En un ejercicio simple de clasificación, una NN puede decidir si las características de entrada pertenecen a una u otra clase, suponiendo que sólo existen dos de estas al realizar los cálculos necesarios en la función de activación $f_h(u)$, en donde u es igual a $\mathbf{wx} - \beta$. De tal manera que $y = f_h(u)$ y β es el término de sesgo o ajuste en la función de activación f_h. Por lo tanto, $\mathbf{wx} = w_1x_1 + w_2x_2 + ... + w_nx_n$ es la suma ponderada del vector de pesos $\mathbf{W}$ con la matriz de valores $\mathbf{X}$.

La función $f_h(u)$ usualmente se elige no lineal, pero esta decisión depende del objetivo de la NN. Algunos ejemplos de funciones de activación son:

- Sigmoide:
$$f_h(u) = \frac{1}{1+e^{-u}}$$
- Signo (lineal):
$$f_h(u) = \text{sign}(u)$$
- Tangente hiperbólica:
$$f_h(u) = \frac{e^{2u}-1}{e^{2u}+1}$$
- Unidad Lineal Rectificada (ReLU):
$$f_h(u) = \text{máx}\{u, 0\}$$
- Tangente hiperbólica dura:
$$f_h(u) = \text{máx}\{\text{mín}[u, 1], -1\}$$

En el ejemplo previo de clasificación, y aplicando una SLP, la capa de entrada estaría configurada de n nodos que transmiten los valores de la matriz $\mathbf{X}$ con sus pesos $\mathbf{W}$ hacia un nodo de salida. La capa de entrada no realiza ningún cálculo y es hasta el nodo de salida en donde, para este ejemplo particular, se calcula una función lineal $\mathbf{WX} = \sum_{i=1}^{n} w_i x_i$. Es así como el signo del valor real es usado para predecir o asignar una observación a un grupo de la siguiente manera:

$$\hat{y} = \text{sign}\{\mathbf{WX} - \beta\} = \text{sign}\left\{\left(\sum_{j=1}^{n} w_j x_j\right) - \beta\right\}$$

La función signo es útil ya que permite asociar un valor real en una de dos categorías $\{+1, -1\}$ dependiendo del signo del resultado en u. El error en la predicción se calcula entonces como:

$$E(\mathbf{X}) = y_j - \hat{y}_j \tag{3.16}$$

Siendo y_j el valor real en la j-ésima observación y $\hat{y_j}$ el valor predicho en la misma observación. En el caso de que el error de predicción medido mediante el valor esperado sea diferente de cero, los pesos de la NN deberán actualizarse mediante su gradiente descendiente para minimizar el error [15]. Dependiendo del tipo de función de activación empleada, será entonces el tipo de gradiente a usar; en el caso de la función signo, al tomar valores reales constantes sobre largas partes del dominio, el gradiente toma valores de cero en puntos diferenciables y

por tanto, el resultado es una superficie o plano del tipo escalera que no es recomendable en el comportamiento descendiente del gradiente [15]. El algoritmo del SLP implícitamente usa una aproximación suavizada del gradiente en su función objetivo respecto a cada ejemplo:

$$\nabla F_{\text{suavizada}} = \sum_{\mathbf{X},y} (y - \hat{y})\mathbf{X} \tag{3.17}$$

A pesar de que la función objetivo está definida sobre todo el conjunto de entrenamiento, el algoritmo de entrenamiento de las NN trabaja al alimentar cada entrada de datos $\mathbf{X}$ en la red de una a una o en pequeños lotes para crear la predicción $\hat{y}$. Los pesos son entonces actualizados con base en el valor esperado de la ecuación 3.16, el vector de pesos $\mathbf{W}$ se actualiza entonces como:

$$\mathbf{W} \Leftarrow \mathbf{W} + \alpha(y - \hat{y})\mathbf{X} \tag{3.18}$$

El parámetro α controla la tasa de aprendizaje de la NN. El SLP entonces se cicla en todas las instancias del conjunto de entrenamiento de manera aleatoria y va ajustando el vector de pesos hasta que se el algoritmo converge, es decir, se encuentran los pesos en donde a pesar de las iteraciones ya no hay mejoras en el comportamiento.

Red neuronal multicapa

Las NN multicapa contienen más de una capa intermedia entre la entrada y salida en su configuración, estás capas también se les conoce como *escondidas* u *ocultas* ya que los cálculos hechos en ellas no son visibles para el usuario [15]. Usualmente este tipo de NN trabajan en sus capas de manera similar a una NN de una sola capa, es decir existe un único flujo de información en una sola dirección y se asume que todos los nodos o neuronas de la NN se conectan e intercambian información entre todos. Es decir, las NN multicapa no tienen ningún tipo de retroalimentación producido por la capa de salida o las capas intermedias como se puede observar en la figura 3.7.

Para ejemplificar la configuración y funcionamiento de una NN multicapa, suponga que se quiere una NN con tres capas. La primer capa no es usualmente considerada en el conteo debido a que únicamente transmite los datos y no realiza ningún cálculo en sus unidades [15]. Si una NN contiene $p_1, ..., p_k$ unidades en cada una de sus k capas, entonces el vector de representaciones de las salidas $(\bar{h}_1, ..., \bar{h}_k)$ tiene dimensiones $p_1, ..., p_k$. Se dice entonces que el número de unidades en cada capa es su dimensionalidad. Los pesos de las conexiones entre la capa de entrada

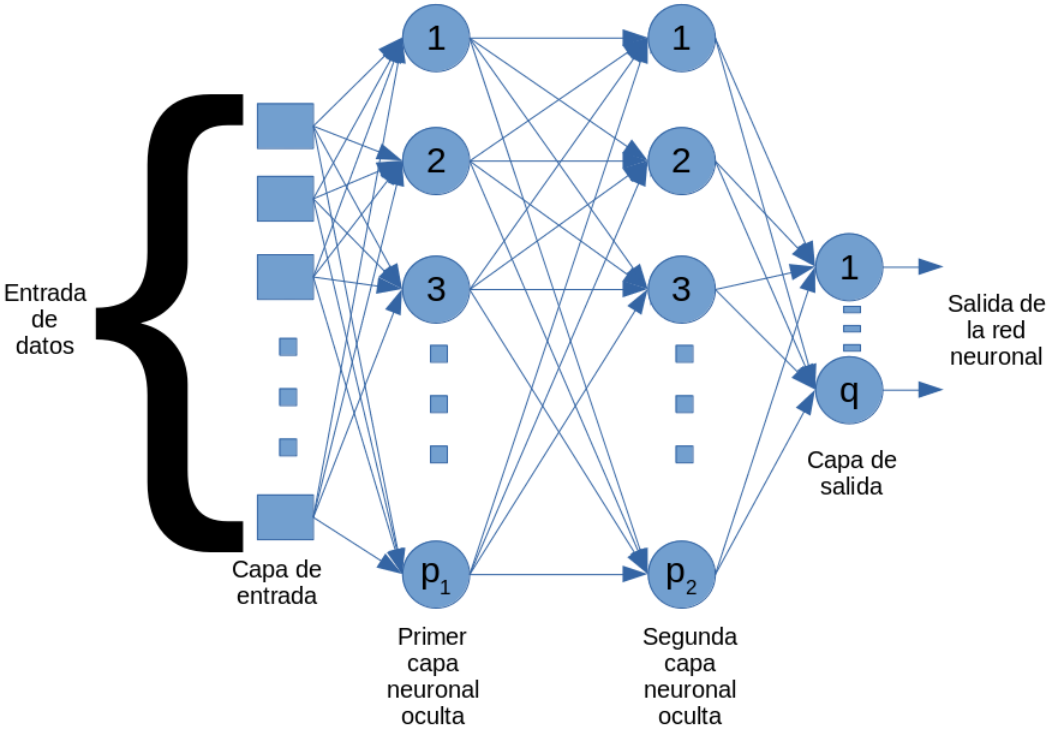


Figura 3.7: Diagrama de una red neuronal multicapa.

y la primer capa oculta son almacenados en una matriz W_1 de dimensión $d \times p_1$, donde los pesos entre la r-ésima capa y la $(r+1)$ capa son calculados por la matriz W_r de dimensión $(p_r \times p_{r+1})$. Si la capa de salida contiene q nodos, entonces la matriz final W_{k+1} es de tamaño $p_k \times q$. El vector d-dimensional $\bar{x}$ es entonces transformado en las salidas usando las siguientes ecuaciones recursivas:

- De la capa de entrada a las capas ocultas:

$$\bar{h}_1 = f_h(W_1^T \bar{x})$$

- Entre capas ocultas:

$$\bar{h}_{p+1} = f_h(W_{p+1}^T \bar{h}_p) \ \forall p \in \{1, ..., k-1\}$$

- De las capas ocultas a la capa de salida:

$$\bar{q} = f_h(W_{k+1}^T \bar{h}_k)$$

La configuración específica de cada NN incluyendo el número de neuronas y capas depende del tipo de problema a solucionar. El ajuste de los pesos, como ya se ha mencionado depende del proceso de entrenamiento y se asume que a cada entrada de datos debe corresponder una salida [48]. Es importante resaltar que estas ecuaciones y la arquitectura de la NN descrita únicamente aplican a redes de flujo continuo hacia adelante, es decir en aquellas donde el flujo de información va únicamente de un lado a otro y no se recomiendan para otras estructuras.

3.3.5. Redes LSTM

Similar a las NN multicapa, las Redes de Memoria a Largo y Corto Plazo (conocidas como LSTM gracias a su nombre en inglés *Long Short-Term Memory*) están compuestas por capas. El vector de entrada es propagado en la red con la intención de predecir un valor o un vector resultante. A diferencia de las NN multicapa que llama neuronas a los elementos de sus capas, en las redes LSTM se les conoce como células. Dichas células tienen la particularidad de que están compuestas por matrices de pesos y compuertas de entrada y salida [50]. A cada capa de las redes LSTM se les llama bloques de memoria y cada uno de estos bloques se compone de células de memoria conectadas de manera recurrente. La red es capaz de interactuar con las células únicamente mediante las compuertas. En la figura 3.8 se muestra un diagrama que ilustra la composición de la célula LSTM.

Este modelo de redes LSTM fue inicialmente propuesto por Hochreiter y Schmidhuber en 1997, su diseño busca solucionar los problemas de fuga (*vanishing*) y explosión (*exploding*) del gradiente [51] que se presenta, usualmente, en las NN entrenadas mediante propagación hacia atrás. Se le llama fuga del gradiente cuando los pesos cambian muy rápido y llegan a ser demasiado pequeños que no tienen efecto alguno en el funcionamiento de la red. Por otro lado, en la explosión del gradiente, los pesos crecen aceleradamente con el mismo resultado en la red.

Una célula de memoria tiene tres parámetros o valores en sus pesos que se ajustan en cada interacción temporal o paso en el tiempo, éstos son:

- Pesos de entrada $\mathbf{W}_i$: Usados para ponderar los datos de entrada en el momento temporal actual.
- Pesos de salida $\mathbf{W}_o$: Usados para ponderar la salida en el siguiente momento temporal.
- Estado interno C_t: Estado interno usado en el cálculo de la salida para el momento temporal actual.

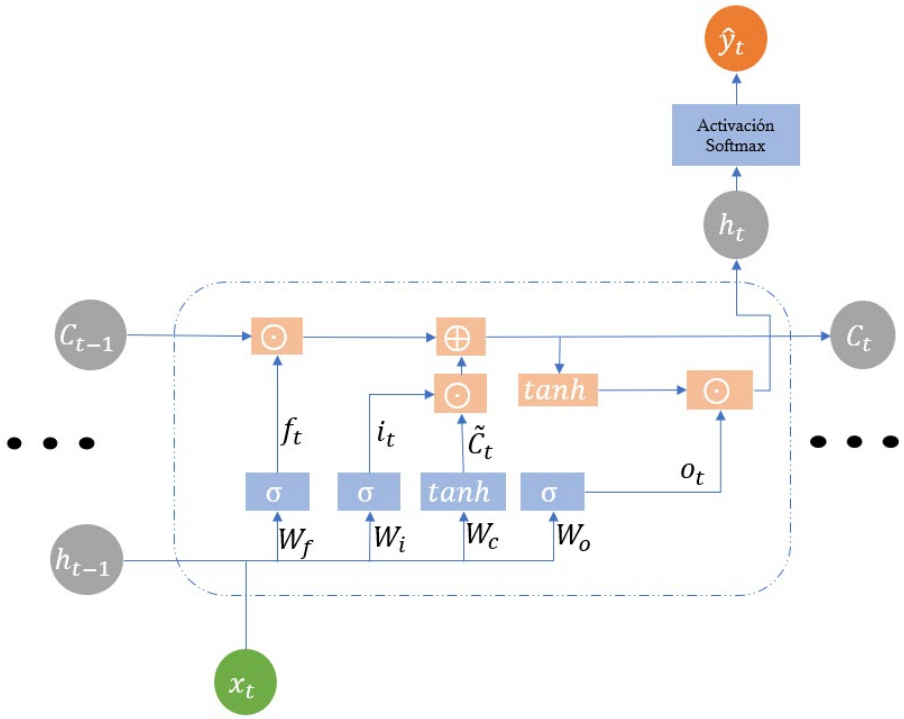


Figura 3.8: Diagrama de una célula LSTM.

El funcionamiento de las redes LSTM se basa en el control del flujo de información hacia dentro y fuera de la célula de memoria [50]. En las compuertas anteriormente mencionadas, reside una capa de activación con la función sigmoide (σ) seguida de una operación de producto punto. Cada compuerta es controlada por el estado de la red en el periodo de tiempo previo h_{t-1} y la entrada de datos actual x_t. La compuerta de olvido decide qué información será desechada del estado interno C_t. Por otro lado, la compuerta de entrada decide qué información será almacenada en la célula. Por último, la compuerta de salida determina el nuevo estado h_t. Las ecuaciones 3.19 a 3.24 describen las operaciones que se realizan dentro de una célula LSTM.

$$f_t = \sigma(\mathbf{W}_f \cdot [h_{t-1}, x_t] + \mathbf{b}_f) \tag{3.19}$$

$$i_t = \sigma(\mathbf{W}_i \cdot [h_{t-1}, x_t] + \mathbf{b}_i) \tag{3.20}$$

$$o_t = \sigma(\mathbf{W}_o \cdot [h_{t-1}, x_t] + \mathbf{b}_o) \tag{3.21}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{3.22}$$

$$\tilde{C}_t = \tanh(\mathbf{W}_c \cdot [h_{t-1}, x_t] + \mathbf{b}_c) \tag{3.23}$$

$$h_t = o_t * \tanh(C_t) \tag{3.24}$$

Las matrices $\mathbf{W}_f$, $\mathbf{W}_i$, $\mathbf{W}_o$ y $\mathbf{W}_c$; así como los vectores $\mathbf{b}_f$, $\mathbf{b}_i$, $\mathbf{b}_o$ y $\mathbf{b}_c$; se ajustarán dentro de la red durante la fase de entrenamiento.

3.3.6. Algoritmo XGBoost

XGBoost es un algoritmo de ML basado en árboles de decisión y aumento del gradiente, su nombre le es otorgado del inglés *Extreme Gradient Boosting*. Este algoritmo fue propuesto por Chen, T. y Guestrin, C. en 2016 [17]. El funcionamiento tanto del aumento del gradiente como el funcionamiento del algoritmo XGBoost, son descritos en [52].

Inicialmente, el algoritmo XGBoost crea un árbol de regresión a partir de los datos de entrada para obtener el primer resultado estimado Y_1. Un segundo árbol es aprendido Y con base en la diferencia entre los valores o etiquetas reales y los predecidos en el paso anterior. El funcionamiento matemático se describe en las ecuaciones 3.25 a 3.29, resaltando el cálculo del flujo del gradiente. El comportamiento del modelo, ayuda a reducir el sobreajuste y mejora la habilidad del modelo de generalizar [24]. Posteriormente, los primeros tres términos de la fórmula de Taylor, que contienen términos constantes además de primeras y segundas derivadas.

$$\text{Obj}^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \omega(f_t) + C \tag{3.25}$$

La función objetivo definida en la ecuación 3.25 se calcula en cada ronda, f_t es elegida con la intención de minimizar la función objetivo, es decir, el error entre el resultado predecido y el valor actual se reduce tras agregar f_t. l es la función de error, mientras que ω es un término de regularización. La función de error trata de ajustar los datos de entrenamiento tanto como sea posible, el término de regularización se encarga de encontrar un modelo más simple. Al encontrar un modelo simple, la aleatoriedad de los resultados tras el ajuste es relativamente chico, lo cual conduce a la prevención del sobreajuste para finalmente, obtener un modelo más estable.

Cuando la función de error l no es cuadrática, los primeros tres términos de la expansión de Taylor son usados para aproximar la función objetivo original mediante la ecuación 3.26.

$$\text{Obj}^{(t)} = \sum_{i=1}^{n} \left[l\left(y_i, \hat{y}_i^{(t-1)}\right) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \omega(f_t) + C \tag{3.26}$$

en donde g_i es la primer derivada de la función de error, mientras que h_i es la segunda derivada de la misma función.

$$g_i = \partial_{\hat{y}_i^{(t-1)}} l\left(y_i, \hat{y}_i^{(t-1)}\right) \tag{3.27}$$

$$h_i = \partial^2_{\hat{y}_i^{(t-1)}} l\left(y_i, \hat{y}_i^{(t-1)}\right) \tag{3.28}$$

Al remover el término constante, es decir, la diferencia entre el valor real y el predecido de la ronda previa, la función objetivo depende únicamente de la primera y segunda derivada de la función de error en cada punto de los datos.

$$\text{Obj}^{(t)} = \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \omega(f_t) \tag{3.29}$$

El algoritmo primero evalúa y clasifica los valores propios, el modelo del árbol necesita determinar los mejores puntos de segmentación para después, almacenarlos en bloques. Esta estructura será reusada en las siguientes iteraciones del algoritmo, reduciendo la complejidad computacional.

3.3.7. Redes LSTM Convolucionales (ConvLSTM)

Las Redes LSTM Convolucionales fueron propuestas por Shi, X., Chen, Z. Wang, H y Yeung, D., en [53], al buscar una solución para el pronóstico de precipitaciones. En su propuesta, los autores describen el modelo ConvLSTM como estructuras convolucionales en las transiciones de entrada a estado y entre estados de las capas LSTM. El entrenamiento de dicha red se logra al apilar múltiples capas ConvLSTM para formar una estructura apta para la generación de pronósticos.

En estas redes ConvLSTM todas las entradas $X_1, ..., X_t$, salidas de celda $C_1, ..., C_t$, estados ocultos $H_1, ..., H_t$ y compuertas i_t, f_t, o_t; son tensores de tres dimensiones, en los cuales, las dos últimas dimensiones son espaciales. El estado futuro de una determinada celda en la red es calculado mediante las entradas y los estados pasados de sus vecinos locales. Esto se logra mediante el uso de un operador de convolución en las transiciones de estado a estado y de entrada a estado. Existe ya la implementación de las redes ConvLSTM como una clase en la librería de Keras [54], en donde se detallan los argumentos a elegir para su funcionamiento.

3.3.8. Redes LSTM combinadas con Redes Neuronales Convolucionales (CNN-LSTM)

La combinación de las Redes Neuronales Convolucionales y las Redes LSTM ha sido aplicada tanto en problemas de clasificación como en análisis de regresión. La idea de esta combinación tiene su base en el uso de capas convolucionales vinculadas a capas LSTM. La capa de entrada de este modelo es de tipo convolucional mientras que la de salida es una capa de tipo LSTM con varias capas convolucionales totalmente conectadas entre estas dos [55].

La ventaja de este modelo radica en la habilidad de las capas convolucionales de extraer conocimiento, filtrar ruidos y aprender la representación interna de las series temporales. Además, las redes LSTM tienen un alto grado de efectividad al identificar dependencias y patrones temporales [56].

Actualmente, no existe una librería en Python con la implementación de este modelo, por lo que se debe construir haciendo uso de las capas Conv1D [57] y LSTM [58] de Keras.

3.4. Algoritmos metaheurísticos

Antes de definir a los algoritmos metaheurísticos, es necesario explicar el objetivo de éstos el cual, es encontrar una solución óptima en un espacio de búsqueda definido y dentro del cumplimiento de las restricciones [59]. Una solución es óptima si minimiza o maximiza la función objetivo. Por otro lado, un espacio de soluciones es el conjunto de todas las posibles soluciones, mientras que, una solución factible es aquella que cumple con todas las restricciones definidas por el problema de optimización.

Los algoritmos metaheurísticos son procedimientos iterativos de aproximación, éstos guían una heurística subordinada combinando distintos conceptos para explorar el espacio de búsqueda [60]. Son empleados, usualmente, en problemas complejos de optimización combinatoria o de ingeniería con la intención de encontrar una solución factible. Algunas de las aplicaciones más frecuentes son:

- Encontrar el camino más corto entre varios puntos.
- Realizar una asignación óptima de trabajadores a tareas por realizar.
- Hallar una secuencia óptima de trabajos o procesos en una línea de producción.

Los metaheurísticos se usan en las aplicaciones mencionadas porque los problemas a solucionar suelen presentar una gran complejidad computacional, tiempos de solución y costos computacionales altos al buscar una solución exacta [61]. Además, son de fácil implementación y de bajo costo computacional. Sin embargo, usualmente no son capaces de proporcionar una solución exacta, pero sí pueden encontrar una o varias soluciones factibles. Es atractivo el uso de algoritmos metaheurísticos cuando no existe un método exacto de solución al problema o cuando no se está buscando una solución óptima.

Un buen algoritmo metaheurístico debe establecer un balance entre la **intensificación** y la **diversificación** para encontrar buenas soluciones. Por un lado se habla de **intensificación**, cuando el esfuerzo empleado se aplica en la búsqueda de soluciones en una región o área actual del espacio de búsqueda. Por el otro, se le llama **diversificación**, a la búsqueda de soluciones en distintas regiones del espacio de búsqueda. El equilibrio entre intensificación y diversificación es útil para identificar, de manera temprana, regiones del espacio con soluciones de buena calidad y para no dedicar mucho tiempo en regiones del espacio no prometedoras o ya exploradas [60].

De acuerdo con [62], los metaheurísticos presentan las siguientes propiedades:

- La función objetivo $f(x)$ debe ser calculada para todos los valores de $x \in S$, en donde S representa al espacio de búsqueda.
- Emplean pocos parámetros para guiar la búsqueda. A pesar de que dichos parámetros influyen en las soluciones encontradas y en la velocidad de convergencia, los valores óptimos se desconocen y se ajustan de manera empírica, basados en conocimiento previo o como resultado de alguna experimentación.
- Se debe especificar un punto de arranque o inicio aunque, algunas veces, la solucion inicial es elegida de manera aleatoria.
- Se define un criterio de paro usualmente basado en la cantidad de iteraciones, evaluaciones de $f(x)$ o cuando se ha dejado de presentar mejoras a la solución durante cierto número de iteraciones.
- Generalmente son algoritmos de fácil implementación y ejecución paralela eficiente.

A pesar de que el uso más frecuente de los algoritmos metaheurísticos está directamente relacionado con problemas de optimización, en esta tesina se hace uso de dichos algoritmos para el ajuste de los hiperparámetros requeridos por los

algoritmos de ML. Al no existir una solución exacta para el ajuste mencionado ni un procedimiento definido formalmente, se experimentará con los algoritmos metaheurísticos con la intención de encontrar una solución factible que proporcione los mejores resultados evaluados con las métricas de desempeño, mismas que fungirán como funciones objetivo. Se decidió usar tres algoritmos metaheurísticos diferentes para emplearlos con los algoritmos de ML: búsqueda aleatoria, recocido simulado y estrategias evolutivas.

3.4.1. Recocido simulado

Esta metaheurística está inspirada en un proceso físico de recocido usado para encontrar estados de baja energía en sólidos [63]. El recocido es un proceso en el que un sólido es fundido aumentando su temperatura, después, la misma temperatura es reducida de manera progresiva con la intención de encontrar un estado sólido de menor energía. Si el enfriamiento ocurre muy rápido, se dan irregularidades en la estructura general del sólido, esto conlleva a estados energéticos relativamente altos. Por otro lado, un proceso de recocido que es llevado de manera cuidadosa en donde la temperatura se mantiene el suficiente tiempo en cada nivel para alcanzar un equilibrio, conduce a estructuras regulares asociadas con estados energéticos bajos. Resulta menos probable que el proceso quede atrapado en un estado energético alto cuando se evita que la temperatura se aleje demasiado del nivel de energía actual.

Desde el punto de vista de la optimización, una solución corresponde a un estado del sistema y el costo de dicha solución corresponde a la energía del sistema. En cada iteración, la solución actual es modificada al elegir aleatoriamente un movimiento de una clase particular de transformaciones. Si la nueva solución muestra una mejora, es entonces aceptada y se convierte en la nueva solución [64]. De otro modo, la solución es aceptada de acuerdo al criterio de Metrópolis, en donde la probabilidad de aceptación está relacionada con la magnitud del incremento del costo y del parámetro de temperatura. Dicho parámetro es reducido de manera progresiva de acuerdo con un esquema de enfriamiento, además, una cantidad definida de iteraciones se ejecutan en cada nivel de temperatura [62]. Cuando la temperatura es suficientemente baja, únicamente se aceptan soluciones que mejoran el costo.

El criterio de aceptación de Metrópolis determina si una solución nueva $\hat{x}$ se acepta o se rechaza para la iteración $k + 1$. Para un problema de minimización, dicho criterio se define de acuerdo a la siguiente probabilidad:

$$P(\text{aceptar } \hat{x}) = \begin{cases} 1 & \text{si } f(\hat{x}) < f(x_k) \\ \exp\left(\frac{f(x_k)-f(\hat{x})}{T}\right) & \text{si } f(\hat{x}) \geq f(x_k) \end{cases} \tag{3.30}$$

si $f(\hat{x}) \geq f(x_k)$, entonces se realiza lo siguiente:

$$\text{Si } \mathcal{U}(0,1) < \exp\left(\frac{f(x_k) - f(\hat{x})}{T}\right)$$

entonces se define $x_{k+1} \leftarrow \hat{x}$. De lo contrario, se define $x_{k+1} \leftarrow x_k$.

Es importante resaltar que T es el valor de la temperatura inicial, $\mathcal{U}(0,1)$ es un número aleatorio uniformemente distribuido entre 0 y 1, $f(\hat{x})$ es el valor de la función objetivo evaluado con la solución $\hat{x}$ y $f(x_k)$ es el valor de la función objetivo evaluado en la solución de la iteración actual k.

El esquema de enfriamiento se puede definir empleando cualquiera de las siguientes tasas:

- Enfriamiento exponencial o geométrico: $T_k = \alpha T_{k-1}$, en donde $\alpha \in (0.8, 1)$.
- Enfriamiento lineal simple: $T_k = \left(\frac{K-k}{K}\right) T_0$, donde K es el número máximo de iteraciones, k es la iteración actual y T_0 es la temperatura inicial.
- Enfriamiento lineal: $T_k = T_0 - \psi k$, donde $\psi = \frac{T_0 - T_{\text{mín}}}{T_{\text{mín}}}$ y $T_{\text{mín}}$ es la temperatura mínima final deseada.
- Enfriamiento inverso: $T_k = \frac{T_{k-1}}{1 + \beta T_{k-1}}$ en donde β es una constante del orden de 0.001.
- Enfriamiento logarítmico: $T_k = \frac{c}{\ln k}$, en donde c es una constante.
- Enfriamiento lineal inverso: $T_k = T_0/k$.

El algoritmo 1 describe el flujo de la información durante su ejecución así como

sus tareas. Como resultado, se obtiene una solución al problema de optimización.

Algoritmo 1: Recocido Simulado.

Paso 1: Generar un estado inicial x_0;
Paso 2: Fijar la temperatura inicial T;
Paso 3: Definir la iteración $k \leftarrow 0$;
Paso 4: **mientras** $k < K$ *o* $T > T_{\text{mín}}$ **hacer**
  Generar un vector independiente $d_k \sim \mathcal{N}(0, \sigma^2)$;
  Generar solución nueva $\hat{x} \leftarrow x_k + d_k$;
  si $f(\hat{x}) < f(x_k)$ **entonces**
    Definir $x_{k+1} \leftarrow \hat{x}$;
  en otro caso
    si $\mathcal{U}(0,1) < \exp[(f(x_k) - f(\hat{x}))/T]$ **entonces**
      Definir $x_{k+1} \leftarrow \hat{x}$;
    en otro caso
      Definir $x_{k+1} \leftarrow x_k$;
    fin
  fin
  Hacer $k \leftarrow k + 1$;
fin
Paso 5: Reducir la temperatura T de acuerdo al programa de enfriamiento;
Paso 6: Volver desde el Paso 3;
Paso 7: Detener el algoritmo en las K iteraciones o en un valor mínimo de T;
Resultado: Recuperar última solución.

3.4.2. Búsqueda aleatoria

El algoritmo metaheurístico de búsqueda aleatoria se caracteriza por ser un método simple de búsqueda de soluciones [62]. Durante su ejecución, la siguiente solución es elegida aleatoriamente en el espacio de búsqueda siguiendo alguna distribución de probabilidad. La solución final es la hallada tras la ejecución del algoritmo durante cierto número de iteraciones.

Como ventajas, este algoritmo es de muy fácil implementación, no requiere de ajuste paramétrico y se puede aplicar a cualquier función objetivo. Por otro lado, puede necesitar un número elevado de iteraciones para encontrar una buena solución en espacios de búsqueda grandes [62].

El algoritmo 2 describe los pasos a seguir para la implementación de este

metaheurístico. Hay otros algoritmos relacionados con la búsqueda aleatoria como la Búsqueda Aleatoria Localizada, la cual, hace uso de una distribución de probabilidad normal para modificar el vector de solución con el riesgo de salir del espacio de búsqueda. La Búsqueda Aleatoria Localizada Mejorada tiene sus bases en la Búsqueda Aleatoria Localizada además de integrar dos componentes adicionales: paso inverso, que eleva el valor de la función de costo y un término de sesgo, que ayuda a direccionar la búsqueda de soluciones.

Algoritmo 2: Búsqueda Aleatoria Simple.

Paso 1: Elegir un vector inicial x_0;
Paso 2: Calcular $f(x_0)$;
Paso 3: Definir $k \leftarrow 0$;
Paso 4: **mientras** $k \leq K$ **hacer**
  Generar un nuevo vector independiente $\hat{x}$ de acuerdo a una distribución de probabilidad;
  si $f(\hat{x}) < f(x_k)$ **entonces**
    Definir $x_{k+1} \leftarrow \hat{x}$;
  en otro caso
    $x_{k+1} \leftarrow x_k$;
  fin
  Hacer $k \leftarrow k + 1$;
fin
Resultado: Recuperar última solución.

3.4.3. Estrategias evolutivas

Los algoritmos de Estrategias Evolutivas fueron propuestos, originalmente, por I. Rechenberg y H.-P. Schwefel en 1965 [62] y forman parte de los algoritmos evolutivos. A este algoritmo se le conoce como Estrategias Evolutivas (1+1) ya que se crea inicialmente un vector padre y con la información de él se genera un solo hijo. El hijo se mantiene si resulta mejor que el padre evaluado en la función objetivo. El algoritmo de Estrategias Evolutivas (1+1) se compone de tres fases:

1. Inicialización.
2. Mutación.
3. Selección.

En la fase de inicialización se crea un vector, mejor conocido como *individuo*, que cumple con la función de padre. Dicho individuo padre $x_{j,k=0}$ se crea una sola vez al inicio del algoritmo de la siguiente manera:

$$x_j^L \leq x_{j,0} \leq x_j^U \tag{3.31}$$

en donde $j = 1, 2, ..., D$, $k = 0$ es la iteración inicial, y x_j^L, x_j^U son los límites inferiores y superiores que delimitan el espacio de búsqueda.

Para la etapa de mutación un nuevo individuo hijo debe generarse mediante el operador de mutación:

$$y_k = x_k + \mathcal{N}(0, \sigma) \tag{3.32}$$

en donde k hace referencia a la iteración actual, y $\mathcal{N}(0, \sigma)$ es un vector de números Gaussianos independientes con media cero y desviación estándar σ. Se debe tener un valor de σ lo suficientemente grande para que las mutaciones permitan explorar todas las áreas del espacio de búsqueda y, al mismo tiempo, lo suficientemente pequeño que permita encontrar una solución aceptable dentro de un área de búsqueda.

Para ayudar en el ajuste de σ, I. Rechenberg propuso en 1973 la llamada **Regla de éxito 1/5** [62], ésta regla permite ajustar σ durante el proceso evolutivo después de cada $h(> 10)$ iteraciones, garantizando así, una convergencia del algoritmo de acuerdo a:

$$\sigma = \begin{cases} \sigma/c & \text{si } p_s > 1/5 \\ \sigma \cdot c & \text{si } p_s < 1/5 \\ \sigma & \text{si } p_s = 1/5 \end{cases} \tag{3.33}$$

en donde p_s es el porcentaje de mutaciones exitosas, y $0.8 \leq c \leq 1$. El algoritmo 3 describe a detalle el funcionamiento de las Estrategias Evolutivas (1+1).

3.5. Métricas para la evaluación del desempeño

Se utilizaron las siguientes medidas de desempeño también llamadas métricas, ya que no existe información en la literatura sobre cuál de éstas es recomendable al estudiar series temporales cortas.

- **Error absoluto medio:** Se abrevia como MAE (del inglés *Mean Absolute Error*) y se caracteriza por darle menor peso a los datos atípicos [65]:

$$MAE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n} \tag{3.34}$$

Algoritmo 3: Estrategias Evolutivas (1+1).

Paso 1: Definir σ, c;
Paso 2: Definir $k \leftarrow 0$, $p_s \leftarrow 0$, $\gamma \leftarrow 0$;
Paso 4: **mientras** $k \leq K$ **hacer**
  Seleccionar aleatoriamente un vector $r \sim \mathcal{N}_j(0, \sigma), \forall j \in [1, 2, ...D]$;
  Hacer $\hat{x_k} \leftarrow x_k + r$;
  si $f(x_k) < f(\hat{x_k})$ **entonces**
    Definir $x_{k+1} \leftarrow x_k$;
  en otro caso
    Definir $x_{k+1} \leftarrow \hat{x_k}$;
    Definir $\gamma \leftarrow \gamma + 1$;
  fin
  si $k \mod h = 0$ **entonces**
    Hacer $p_s \leftarrow \gamma / h$;
    Hacer $\sigma \leftarrow \begin{cases} \sigma/c & \text{si } p_s > 1/5 \\ \sigma \cdot c & \text{si } p_s < 1/5 \\ \sigma & \text{si } p_s = 1/5 \end{cases}$;
    Hacer $\gamma \leftarrow 0$;
  fin
  Hacer $k \leftarrow k + 1$;
fin
Resultado: Recuperar última solución.

- **Error cuadrático medio:** Se abrevia como MSE (del inglés *Mean Squared Error*) funciona bien con sesgos y varianza sobre las predicciones [65]:

$$MSE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n} \tag{3.35}$$

- **Porcentaje de error medio absoluto:** Esta métrica se normaliza a través de las observaciones reales y juega un papel importante en la precisión con series de tiempo [65], su abreviación viene del inglés *Mean Absolute Percentage Error* (MAPE):

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \frac{\hat{y}_i}{y_i}\right| * 100 \tag{3.36}$$

- **Puntuación de varianza explicada**: Similar al valor R^2 empleado en el análisis de regresión, esta métrica evalúa qué tan bien puede el modelo de ML creado explicar la varianza. Es una función integrada de la librería `scikit-learn` [66] implementada en Python. Mientras más cercana se encuentre esta métrica a 1, mejor será el modelo en términos de varianza explicada. Su abreviación proviene del inglés *Explained Variance Score* (EVS):

$$EVS = 1 - \frac{Var(y_i - \hat{y}_i)}{Var(y_i)} \tag{3.37}$$

Cada una de estas métricas permite evaluar la predicción obtenida de los modelos de ML en sus propios términos. Debe tomarse en consideración que tanto el MSE como el MAE se expresarán en términos de los valores de venta, por lo que sirven como un parámetro de comparación entre modelos. Por otro lado, el MAPE determinará la precisión porcentual de cada modelo al devolver el porcentaje de error encontrado. Finalmente, el EVS ayudará para entender qué tan bien se ajusta un modelo en términos de varianza de acuerdo a los datos disponibles. Con estas cuatro métricas se garantiza una perspectiva general sobre los resultados.

Capítulo 4

Resultados

En este capítulo se explica el trabajo realizado mediante las técnicas de aprendizaje de máquina y sus correspondientes ajustes usando algoritmos metaheurísticos para la elaboración de pronósticos de ventas aplicados a series de tiempo cortas. El presente capítulo se divide en cinco secciones: en la primer sección se realiza un análisis exploratorio de los datos, en la segunda se realiza la implementación de la función auto_arima como para establecer un punto de referencia para las siguientes secciones; en la tercer sección se aplican los algoritmos de ML y se analizan los resultados obtenidos, en la cuarta sección se experimenta con la combinación de algoritmos de ML y metaheurísticos para, finalmente, en la quinta sección se evalúa estadísticamente los resultados obtenidos y elaborar conclusiones.

4.1. Análisis exploratorio de los datos

Se tiene información de ventas para cinco productos diferentes durante los mismos periodos de tiempo, desde diciembre de 2016 a noviembre de 2019. Cada producto cuenta con la cantidad total de ventas mensuales por 36 periodos. Son productos totalmente diferentes entre sí, por lo que no resulta extraño que muestren volúmenes muy distintos de venta. La figura 4.1 muestra brevemente el comportamiento de cada producto. Estos cinco productos son los que la empresa está principalmente interesada en estudiar y predecir, principalmente porque requieren mayor precisión en su pronóstico al incluir componentes de importación. Por motivos de confidencialidad, no se pueden especificar los productos ni el nombre de la empresa que proporcionó la información.

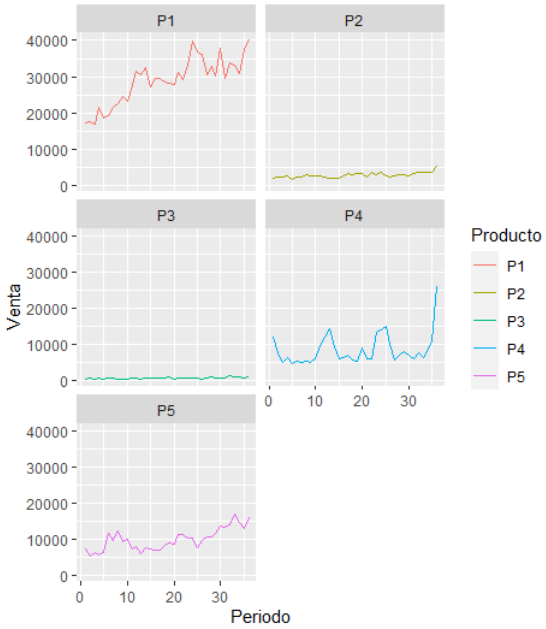


Figura 4.1: Venta de productos

4.1.1. Producto 1

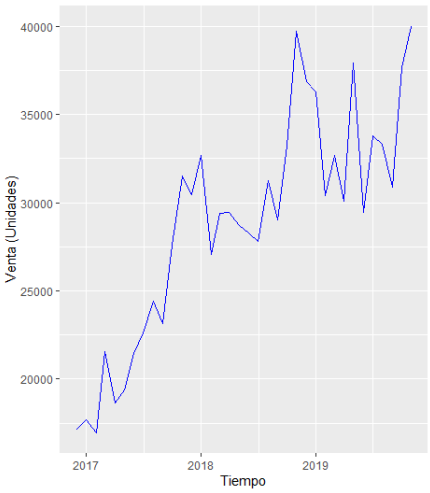
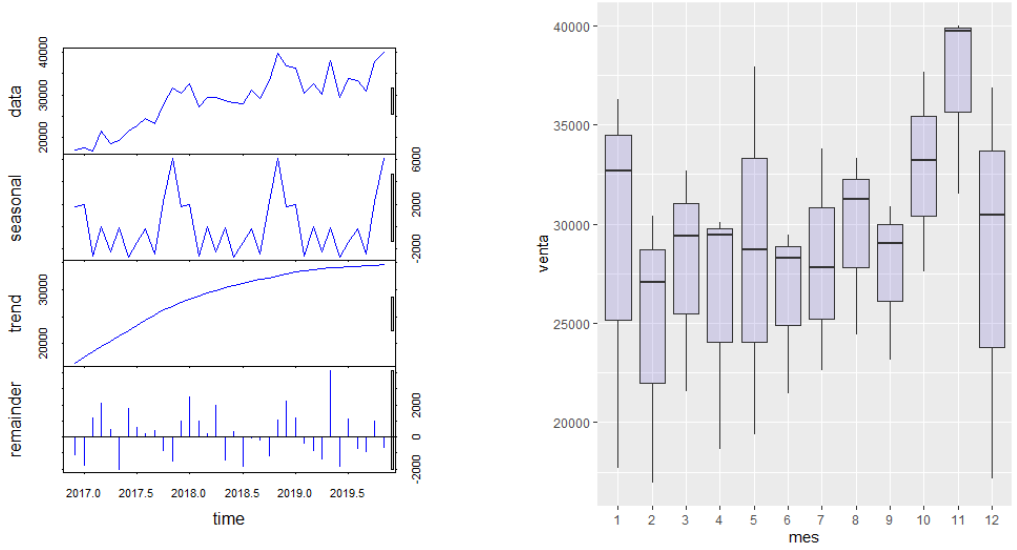


Figura 4.2: Venta del producto 1

El producto 1 tiene un comportamiento con tendencia ascendente como lo muestra la figura 4.2. A primera vista no se aprecia algún efecto estacional, pero al realizar el análisis del diagrama de caja en la figura 4.3(b), se puede notar

que históricamente noviembre o el mes 11, es el que mayores ventas acumula con relación con otros meses. Mientras que diciembre es el mes con más dispersión en las ventas, es decir que acumula la mayor variabilidad.

Al realizar una descomposición STL de la serie temporal (ver figura 4.3(a)), se puede validar el fuerte comportamiento de crecimiento en su tendencia. Por otro lado, también se puede notar un comportamiento estacional sin ser igual para todos los periodos de tiempo, que en este caso es un año.



(a) Descomposición de serie temporal

(b) Venta por mes del producto 1

Figura 4.3: Análisis del producto 1

4.1.2. Producto 2

El producto 2 tiene un comportamiento ligeramente creciente como puede notarse en la figura 4.4, con un crecimiento atípico para el final de año. No se puede determinar a primera vista alguna situación estacional, sin embargo en el análisis de dispersión de la figura 4.5(b) se nota un alto valor medio de ventas para el mes 11, éste mismo mes es también el que tiene una mayor variación de ventas.

En el análisis de la descomposición STL de la serie para el producto 2 de la figura 4.5(a), se observa una tendencia creciente con un comportamiento estacional, ambos con preferencia al crecimiento.

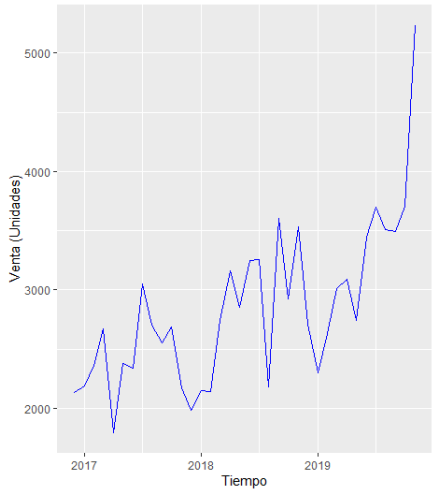
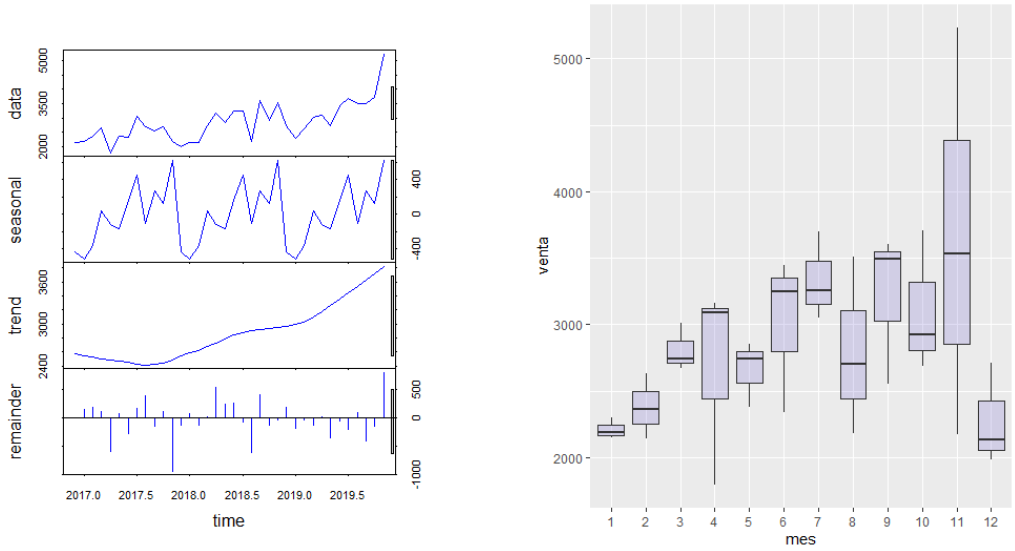


Figura 4.4: Venta del producto 2



(a) Descomposición de serie temporal

(b) Venta por mes del producto 2

Figura 4.5: Análisis del producto 2

4.1.3. Producto 3

El tercer producto tiene un comportamiento creciente a lo largo del tiempo como se visualiza en la figura 4.6, con un crecimiento notable en el año 2019. Al realizar la descomposición STL de la serie temporal, se observa en la figura 4.7(a) una tendencia creciente a partir del año 2018, del mismo modo que se aprecia un

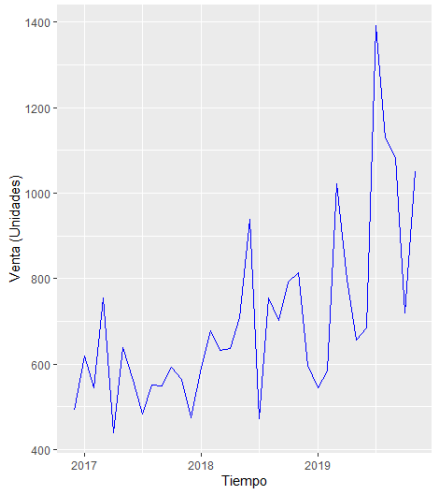
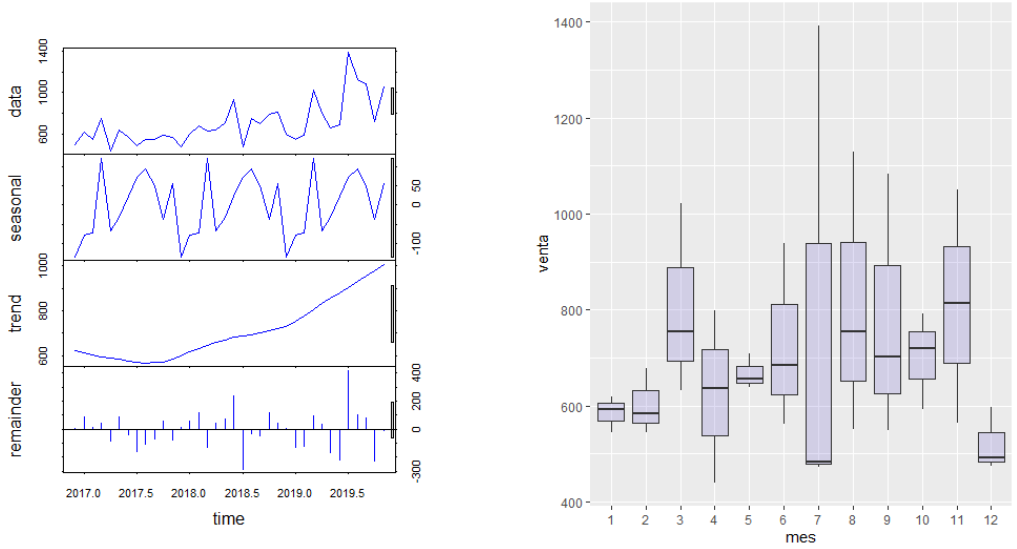


Figura 4.6: Venta del producto 3

patrón estacional marcado para el mismo periodo de tiempo.

En la figura 4.7(b) se observa que este producto tiene un comportamiento muy disperso para el mes de julio al tener una media de ventas muy baja al mismo tiempo que el valor más alto de venta. Es de igual importancia hacer notar que noviembre es el mes con la media más alta en venta.



(a) Descomposición de serie temporal (b) Venta por mes del producto 3

Figura 4.7: Análisis del producto 3

4.1.4. Producto 4

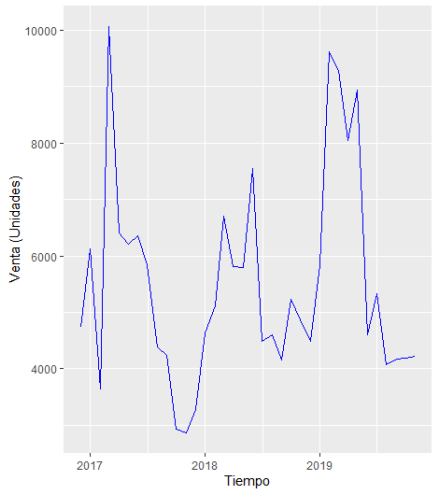


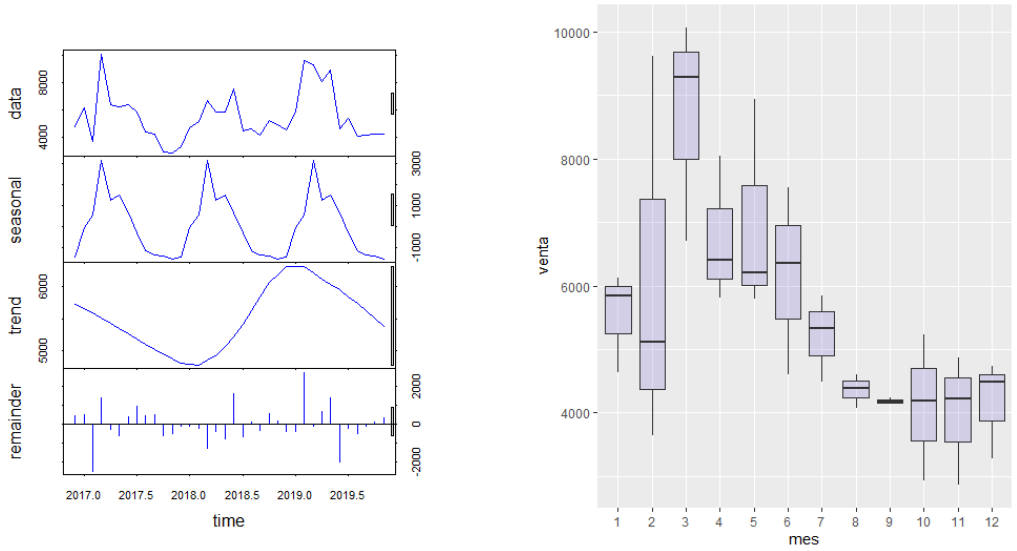
Figura 4.8: Venta del producto 4

El producto 4 tiene un comportamiento estacional y aleatorio, es decir que a simple vista y observando la figura 4.8 no se aprecia un patrón predominante. A pesar de eso, este producto tiene dos picos en ventas a principios de 2017 y 2019. Al realizar un análisis del comportamiento de ventas por mes (figura 4.9(b)) se puede visualizar un marcado comportamiento ascendente para los tres primeros meses del año y un decrecimiento para los restantes. En el análisis de la descomposición STL de la serie temporal de la figura 4.9(a) no se tienen los datos suficientes para determinar una tendencia aunque se aprecia un comportamiento del tipo sinusoidal con una marcada estacionalidad de los datos.

4.1.5. Producto 5

El quinto y último producto muestra un comportamiento claramente ascendente para los años 2018 y 2019 (figura 4.10) en comparación con los récords registrados en 2017. Este producto además tiene un pico de venta a comienzos del año 2020 que se espera no afecte el desempeño de los modelos de ML a aplicar.

Por otro lado, en el análisis del comportamiento mensual de la figura 4.11(b) se observan ventas bajas desde diciembre hasta abril con valores máximos en agosto y una media máxima en julio. Además en la descomposición STL de la figura 4.11(a) se observa una tendencia creciente junto con un comportamiento estacional con picos pasada la mitad del año.



(a) Descomposición de serie temporal (b) Venta por mes del producto 4

Figura 4.9: Análisis del producto 4

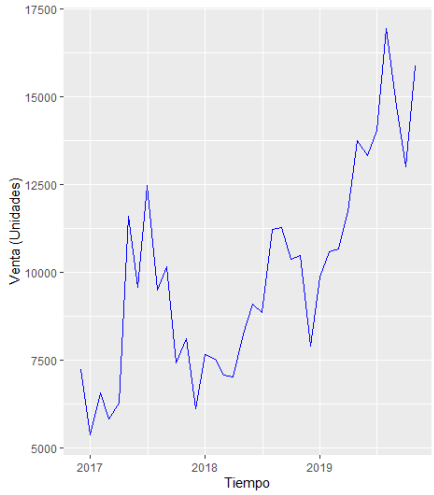
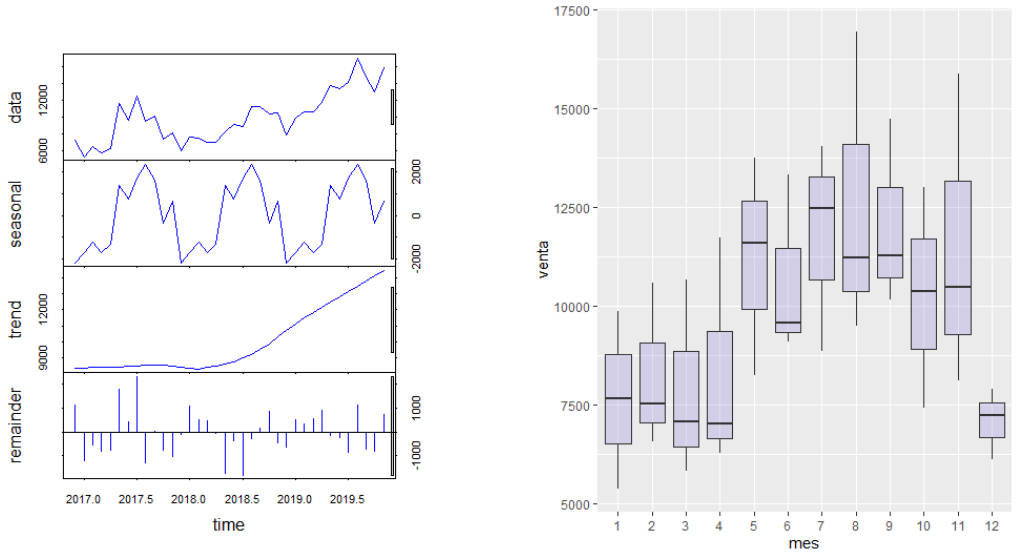


Figura 4.10: Venta del producto 5

Después de este análisis inicial se valida el tipo de información referente a las ventas de los productos. Será de interés observar el comportamiento de los algoritmos de ML aplicados al producto 4 al tener un comportamiento sinusoidal en su tendencia y no contar con más registros para validarlo hacia el pasado. También se deberá prestar atención a lo que ocurrirá con el producto 5 que muestra en su tendencia un inicio estable sin cambios, con un crecimiento repentino y



(a) Descomposición de serie temporal (b) Venta por mes del producto 5

Figura 4.11: Análisis del producto 5

prácticamente lineal desde mediados de 2018.

4.2. Implementación de la función auto_arima

Para tener un punto de referencia y poder comparar la precisión de cada uno de los algoritmos de ML mejorados por metaheurísticas, se hizo uso de la función `auto_arima` de la librería `pmdarima` [67] en Python. La característica principal de esta librería es que se ajusta automáticamente para encontrar los parámetros p, d, q más adecuados a la serie temporal de acuerdo a algún criterio de información establecido (Criterio de Información de Akaike, Criterio de Información Bayesiana o Criterio de Información Hannan-Quinn), siempre en búsqueda del valor más bajo.

No está dentro del alcance de esta tesina la evaluación del comportamiento de la librería y sus funciones por lo que se tomaron los resultados tal cual fueron devueltos por la función. Se decidió usar esta librería ya que es un referente para el pronóstico de series temporales. Se emplearon 31 registros de venta para la autoconfiguración de la función y se pronosticaron 5 registros que se validaron con los originales. Por practicidad se describen los resultados con base en el error cuadrático medio obtenido (MSE) y el error porcentual medio absoluto (MAPE).

Producto 1

Al hacer uso de la función `auto_arima` en el registro de ventas del producto 1, se encontraron los resultados descritos en la tabla 4.1 para cada una de las métricas previamente descritas.

Conjunto de datos	Métrica	Auto ARIMA
Producto 1	MSE	6,787,550.67
	MAE	2,379.49
	EVS	0.50
	MAPE	11.1761 %

Tabla 4.1: Métricas Producto 1 auto_arima.

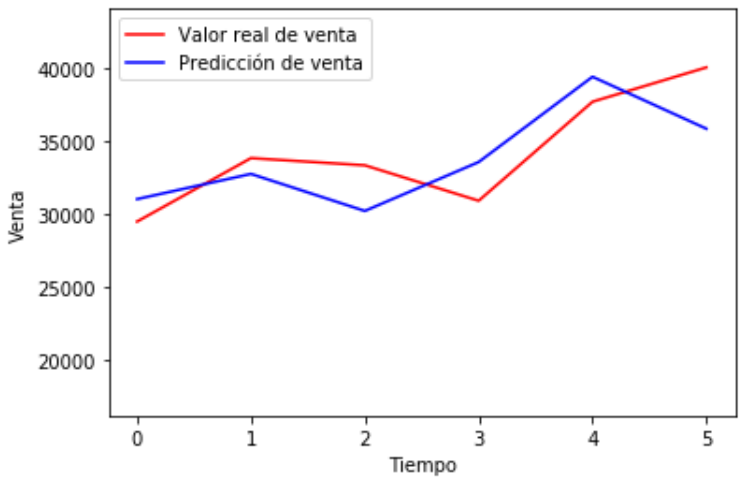


Figura 4.12: Predicciones de la función auto_arima contra los valores reales observados de ventas en los últimos cinco periodos de tiempo para el producto 1.

En la figura 4.12 se puede observar que aunque la función `auto_arima` es un referente en el análisis de series temporales, la predicción no se ajustó con precisión al valor real de las ventas al fluctuar alrededor de dichos valores para los periodos de tiempo de interés. Esto es un indicio de la falta de datos para el correcto funcionamiento de la función. Por otro lado, el MAPE indica una diferencia de 11.176 % con respecto a los valores reales.

Producto 2

Los valores encontrados en las métricas de evaluación para el producto 2 son descritos en la tabla 4.2. Se destaca el valor negativo en la métrica EVS que mide la varianza explicada. Este valor negativo indica que no se tiene un buen desempeño del modelo.

Conjunto de datos	Métrica	Auto ARIMA
Producto 2	MSE	940,353.77
	MAE	623.89
	EVS	-0.67
	MAPE	13.2017 %

Tabla 4.2: Métricas Producto 2 auto_arima.

La figura 4.13 muestra el desempeño de la función `auto_arima` frente a los valores reales. En este caso también se tiene una MSE alto junto con un error en términos porcentuales de 13.2 %, el cual sigue siendo muy alto.

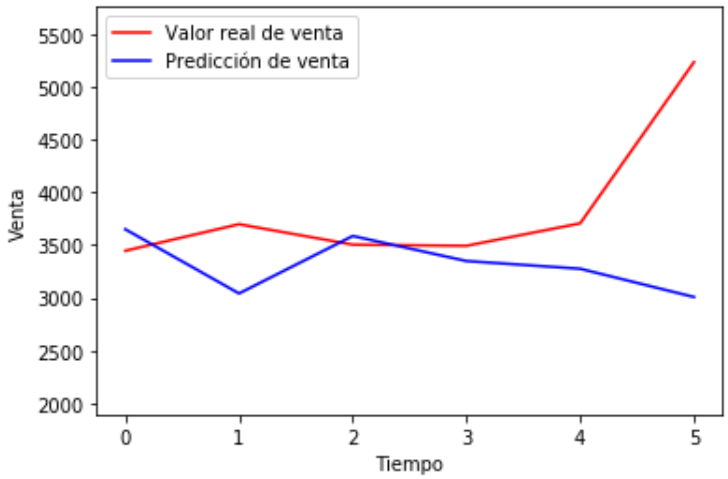


Figura 4.13: Predicciones de la función auto_arima contra los valores reales observados de ventas en los últimos cinco periodos de tiempo para el producto 2.

Producto 3

El desempeño del producto 3 en sus métricas muestra un desempeño muy bajo en términos porcentuales al tener un MAPE de 28.16 % como se muestra

en la tabla 4.3. Por otro lado según el EVS la función `auto_arima` pudo explicar mejor la varianza en comparación con los productos anteriores.

En la gráfica de desempeño de la figura 4.14 se puede observar que el pronóstico obtenido no es confiable al estar muy alejado de los valores de venta reales. Además, la predicción no fue capaz de acertar en los picos altos y bajos de venta.

Conjunto de datos	**Métrica**	**Auto ARIMA**
Producto 3	MSE	133,888.14
	MAE	311.60
	EVS	0.14
	MAPE	28.1628 %

Tabla 4.3: Métricas Producto 3 auto_arima.

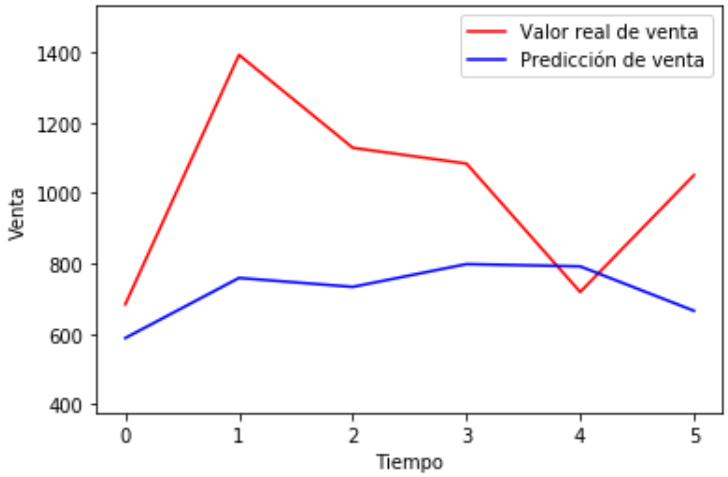


Figura 4.14: Predicciones de la función auto_arima contra los valores reales observados de ventas en los últimos cinco periodos de tiempo para el producto 3.

Producto 4

Las métricas obtenidas y reflejadas en la tabla 4.4 muestran que la función `auto_arima` no tuvo un buen desempeño en términos porcentuales ya que se registró un MAPE de 55.54 % que refleja una alta probabilidad de error. Además obtuvo valores relativamente bajos en las métricas MSE y MAE.

Conjunto de datos	Métrica	Auto ARIMA
Producto 4	MSE	29,545,473.32
	MAE	4,055.43
	EVS	0.39
	MAPE	55.5434 %

Tabla 4.4: Métricas Producto 4 auto_arima.

En la figura 4.15 se observa un comportamiento aceptable sobre todo en el ajuste de la tendencia de los valores pronosticados. Sin embargo, se puede notar la diferencia entre lo pronosticado y lo real en la mayoría de los puntos temporales.

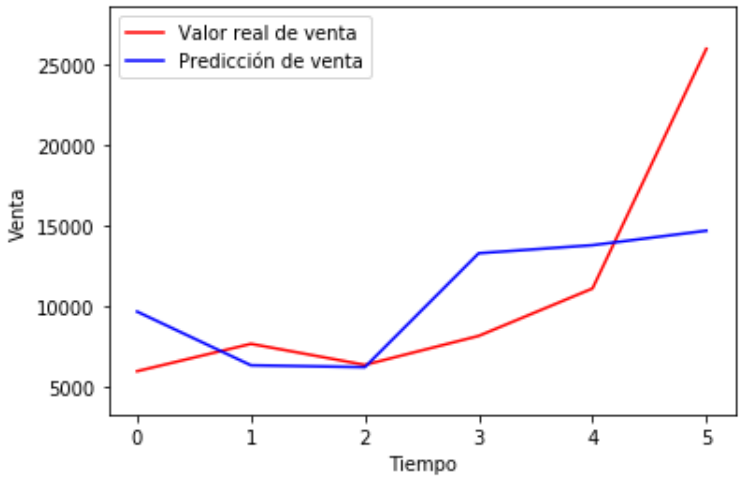


Figura 4.15: Predicciones de la función auto_arima contra los valores reales observados de ventas en los últimos cinco periodos de tiempo para el producto 4.

Producto 5

El comportamiento de la función `auto_arima` para este producto es bastante aceptable al tener un MAPE de 8.41 % como se nota en la tabla 4.5. Este dato es concordante con el MAE que está en términos de las unidades vendidas.

Conjunto de datos	Métrica	Auto ARIMA
Producto 5	MSE	2,267,897.33
	MAE	1,360.79
	EVS	-0.09
	MAPE	8.4129 %

Tabla 4.5: Métricas Producto 5 auto_arima.

En la figura 4.16 es posible observar el comportamiento de los datos pronosticados contra los reales.

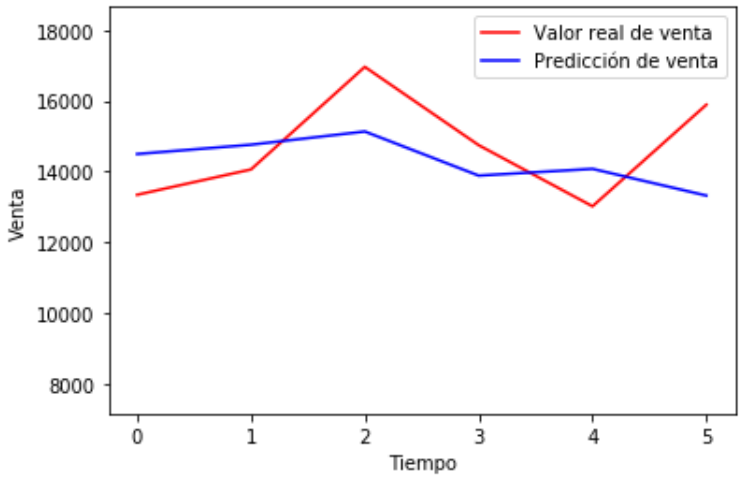


Figura 4.16: Predicciones de la función auto_arima contra los valores reales observados de ventas en los últimos cinco periodos de tiempo para el producto 5.

A manera de resumen sobre la aplicación de la función `auto_arima`, la tabla 4.6 muestra los resultados obtenidos en las métricas para cada uno de los productos. Se observa que únicamente el producto 5 estuvo dentro del margen de precisión de 90 %. Los resultados obtenidos hacen pensar que faltan registros históricos para poder observar resultados satisfactorios con esta librería.

Producto	Métricas	
Producto 1	MSE	6,787,550.67
	MAE	2,379.49
	EVS	0.50
	MAPE	11.1761 %
Producto 2	MSE	940,353.77
	MAE	623.89
	EVS	-0.67
	MAPE	13.2017 %
Producto 3	MSE	133,888.14
	MAE	311.60
	EVS	0.14
	MAPE	28.1628 %
Producto 4	MSE	29,545,473.32
	MAE	4,055.43
	EVS	0.39
	MAPE	55.5434 %
Producto 5	MSE	2,267,897.33
	MAE	1,360.79
	EVS	-0.09
	MAPE	8.4129 %

Tabla 4.6: Resumen de resultados para la función auto_arima en los cinco productos.

4.3. Aplicación de algoritmos de ML

Para la implementación de los algoritmos de ML se siguió, de manera general, la misma metodología de trabajo con cada uno de los productos:

1. Lectura de los registros de venta por producto.
2. Estandarización de los datos.
3. Transformación de serie temporal a aprendizaje supervisado.
4. Implementación de los algoritmos de ML mediante sus librerías como funciones.
5. Transformación inversa de datos para regresar a los valores originales.
6. Evaluación de desempeño.

Toda la experimentación se realizó empleando un equipo de cómputo HP Omen con sistema operativo Pop OS a 64 bits basado en Ubuntu 20.04 LTS, con 8.00GB de memoria RAM y un procesador Intel Core i7 de 6 núcleos físicos (12 núcleos lógicos) a 2.20GHz. Se hizo uso del lenguaje de programación Python en su versión 3.7.3 y de los paquetes de librerías `Numpy, Pandas, Sklearn` y `Keras` en su versión más reciente hasta octubre de 2020.

Se decidió establecer un grupo inicial de comparación para distinguir si existen mejoras en la calidad de los pronósticos al emplear los algoritmos metaheurísticos. Por esta razón es que se obtuvieron las métricas correspondientes de cada uno de los productos empleando hiperparámetros aleatorios que es una situación usual en la aplicación de los algoritmos de ML, sobre estos se suele experimentar y determinar así los hiperparámetros que mejor se adapten a la serie temporal. En la continuación de esta sección se describe el comportamiento de cada uno de los algoritmos de ML empleados y se analiza brevemente el resultado de las métricas.

4.3.1. Soporte de regresión vectorial

Para la aplicación del algoritmo de soporte de regresión vectorial se empleó la función `SVR` de la librería `sklearn.svm` [66]. Se siguió la misma metodología descrita anteriormente.

En la tabla 4.7 se muestran los resultados obtenidos de la aplicación de la función `SVR`. Se nota un incremento con respecto a la función `auto_arima` en

las métricas MSE y MAE aunque un mejor desempeño en el EVS. Aunque en términos de MAPE se está dentro del objetivo de precisión de 90 % en la figura 4.17(a) se nota la baja asertividad del algoritmo sobre todo en los últimos dos periodos.

El producto 2 tiene valores de métricas bastante elevados en comparación con los obtenidos de la función `auto_arima`, se pueden apreciar en la tabla 4.7. Es evidente sin embargo, la mejora de la varianza explicada en la métrica EVS. En la figura 4.17(b) es posible notar una gran diferencia entre lo predecido y los valores reales de venta. Este comportamiento se ve traducido en la métrica MAPE con 30.85 % de error, lo cual se encuentra muy lejos de lo esperado en términos de precisión.

Para el producto 3 se obtienen los valores de métricas descritos en la tabla 4.7 y se nota un incremento en todos sus términos en comparación con lo registrado en la sección anterior para el mismo producto. En términos de error porcentual, el MAPE se incrementó llegando casi a 37 %. Por lo tanto, no se aprecia una mejora con la aplicación del SVR en este producto comparado con lo hallado en la sección anterior. La figura 4.17(c) muestra la comparación entre los valores reales y los pronosticados para este producto empleando SVR.

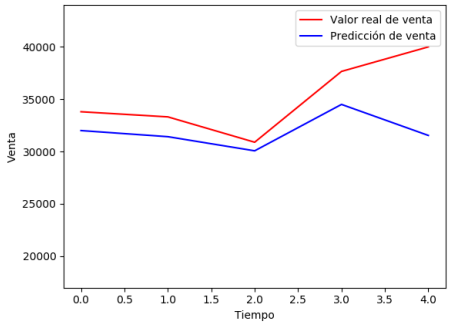
Con respecto a las métricas obtenidas del producto 4 y mostradas en la tabla 4.7 se puede observar una mejora con respecto al desempeño de la función `auto_arima`. En cuanto al error porcentual, el MAPE mejoró al obtener un error porcentual medio absoluto de 17.24 % que es una mejora sustancial con respecto al desempeño de la sección anterior para este producto. Sin embargo, como se muestra en la figura 4.17(d) el SVR no fue capaz de pronosticar el crecimiento para el último periodo de tiempo de manera adecuada.

En el caso del producto 5, las métricas obtenidas de la aplicación del SVR no mostraron una mejora en comparación con la sección anterior. Se puede verificar en la tabla 4.7 que hubo un incremento del MAPE con una magnitud casi tres veces mayor comparado con la función `auto_arima`. La diferencia entre lo predecido y lo real se muestra en la figura 4.17(e). Se observa una deficiencia en el ajuste y en la capacidad predictiva del SVR para este producto.

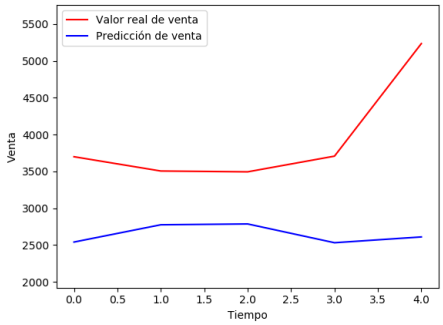
A manera de resumen sobre la aplicación de la SVR, se observa que únicamente el producto 1 estuvo dentro del margen de precisión de 90 %.

Producto	Métrica	
Producto 1	MSE	17,820,186.95
	MAE	3,226.78
	EVS	0.30
	MAPE	8.6413 %
Producto 2	MSE	2,129,530.46
	MAE	1,279.28
	EVS	-0.13
	MAPE	30.8495 %
Producto 3	MSE	224,900.04
	MAE	426.48
	EVS	0.07
	MAPE	36.9458 %
Producto 4	MSE	56,503,194.82
	MAE	3,781.03
	EVS	0.18
	MAPE	17.2404 %
Producto 5	MSE	15,748,281.26
	MAE	3,349.55
	EVS	-1.39
	MAPE	21.4268 %

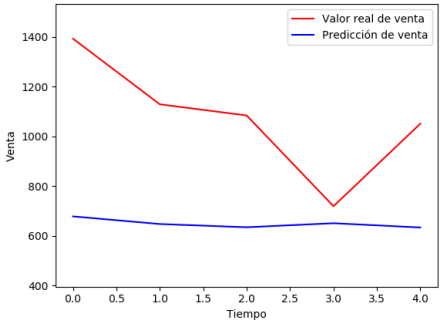
Tabla 4.7: Resultados para Soporte de Regresión Vectorial en los cinco productos.



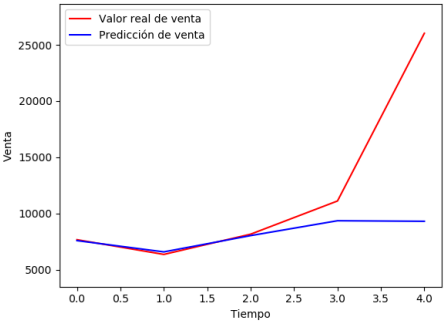
(a) Producto 1



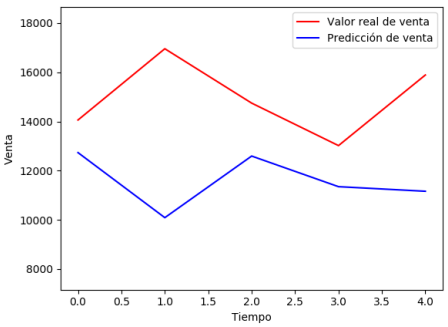
(b) Producto 2



(c) Producto 3



(d) Producto 4



(e) Producto 5

Figura 4.17: Predicciones del SVR contra los valores reales de venta observados en los últimos cinco periodos temporales.

4.3.2. Red neuronal de una capa oculta con una neurona

Para la creación y uso de esta red neuronal se empleó un modelo secuencial con una capa oculta del tipo densa (Dense) o conectada totalmente. Además, la función de activación fue *tanh* y se compiló el modelo empleando el optimizador *Adam*, se especificó un tamaño de lote de 16 con 20 iteraciones en el entrenamiento y la pérdida estuvo evaluada mediante el error cuadrático medio. Todas estas configuraciones están disponibles en la librería de `keras` y corre sobre la plataforma `TensorFlow 2.0`. Se eligieron estos parámetros debido a que en un proceso previo de experimentación, se encontraron dichos parámetros como los que menos inconvenientes presentaron en términos de errores con los datos. Es el mismo tipo de experimentación aleatoria que realizaría un científico de datos al ajustar un modelo a la información disponible.

Producto 1

La red neuronal aplicada tuvo un desempeño muy por debajo de lo esperado, el MAPE obtenido fue de más de 80 % y queda bastante alejado de lo requerido en dicho término. La tabla 4.8 describe las métricas obtenidas de la aplicación de esta NN a este producto.

Conjunto de datos	Métrica	NN
Producto 1	MSE	851,615,764.88
	MAE	28,877.13
	EVS	-0.67
	MAPE	81.8952 %

Tabla 4.8: Métricas Producto 1 NN.

No se presenta un gráfico comparativo debido a que la diferencia entre lo pronosticado y lo real es grande, sin embargo los valores obtenidos en el pronóstico se presentan en la tabla 4.9 junto con los valores reales de venta.

Periodo	Valores predecidos	Valores reales
Mes 1	8463	33801
Mes 2	5710	33312
Mes 3	6002	30888
Mes 4	7521	37656
Mes 5	3576	40001

Tabla 4.9: Comparación pronosticado contra real: Producto 1 NN.

Producto 2

El desempeño de la red neuronal para este producto se muestra en la métricas de la tabla 4.10. Se obtienen resultados muy elevados en cada una de estas métricas, especialmente se tiene un MAPE de casi 90 %, lo que queda muy alejado de los objetivos de esta tesina.

Conjunto de datos	Métrica	NN
Producto 2	MSE	13,041,407.44
	MAE	3,545.49
	EVS	-0.08
	MAPE	89.9513 %

Tabla 4.10: Métricas Producto 2 NN.

No se muestra un gráfico comparativo entre lo real y lo pronosticado debido a que lo predecido estuvo muy por debajo del desempeño esperado. En la tabla 4.11 se muestran los valores pronosticados para cada uno de los periodos de tiempo con los que se está experimentando.

Periodo	Valores predecidos	Valores reales
Mes 1	437	3699
Mes 2	328	3505
Mes 3	409	3494
Mes 4	414	3707
Mes 5	325	5235

Tabla 4.11: Comparación pronosticado contra real: Producto 2 NN.

Producto 3

Este producto no mejoró sus métricas en comparación con los valores del SVR. La métrica MAPE tuvo como resultado un valor de 48 % muy alejado de lo esperado en dicho término. La tabla 4.12 muestra el resto de los valores encontrados en cada una de las métricas.

La figura 4.18 muestra la diferencia entre lo pronosticado y lo real, se puede apreciar que la línea de pronóstico no se ajusta a la forma de las ventas reales. Esto coincide con los valores encontrados en las métricas de este producto.

Conjunto de datos	Métrica	NN
Producto 3	MSE	348,730.54
	MAE	544.45
	EVS	-0.13
	MAPE	48.1218 %

Tabla 4.12: Métricas Producto 3 NN.

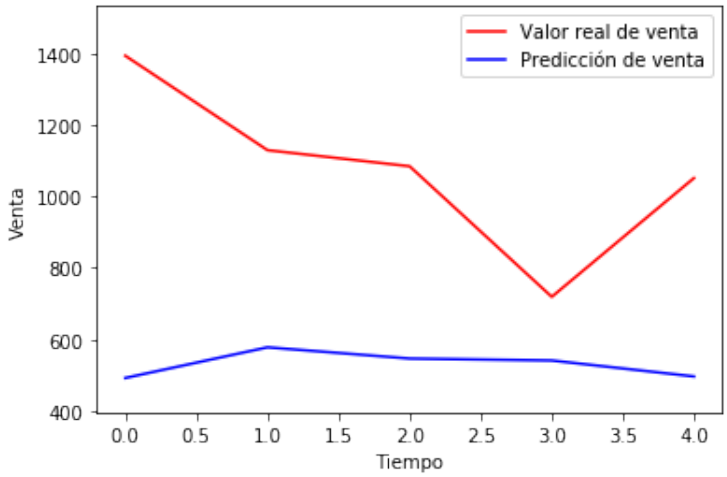


Figura 4.18: Predicciones del NN contra los valores reales observados de ventas en los últimos cinco periodos de tiempo para el Producto 3.

Producto 4

El desempeño de esta NN aplicado al producto 4 se muestra en la tabla 4.13. Se observan resultados en términos de sus métricas por debajo de lo obtenido con el SVR. El MAPE para este producto fue de 28 % que si bien está cercano al objetivo de precisión de 90 %, aún no se considera bueno.

Conjunto de datos	Métrica	NN
Producto 4	MSE	45,406,778.03
	MAE	4,352.37
	EVS	0.38
	MAPE	28.7071 %

Tabla 4.13: Métricas Producto 4 NN.

La figura 4.19 muestra una comparación entre los valores reales de venta y los pronosticados. Debe tomarse en consideración el valor atípico del último mes

de venta real, el cual es un valor atípico.

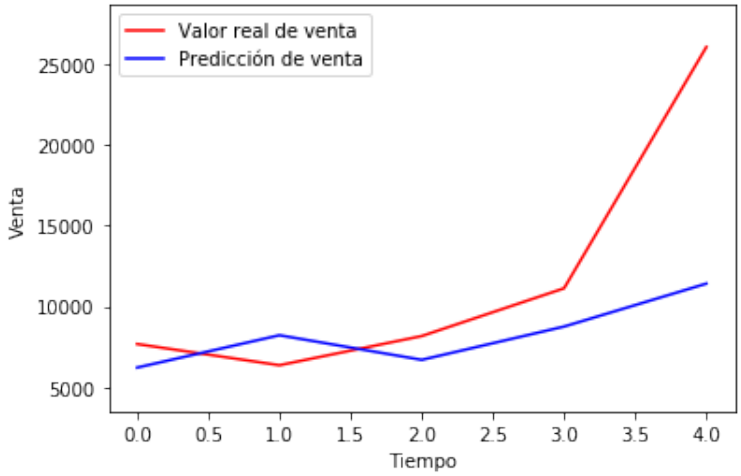


Figura 4.19: Predicciones del NN contra los valores reales observados de ventas en los últimos cinco periodos de tiempo para el Producto 4.

Producto 5

La NN aplicada al producto 5 obtuvo resultados sobresalientes en comparación con las métricas del SVR, dichos resultados se muestran en la tabla 4.14. Sin embargo, todavía no se ha podido obtener un EVS mayor a 0.

Conjunto de datos	Métrica	NN
Producto 5	MSE	5,476,009.70
	MAE	1,911.46
	EVS	-0.14
	MAPE	12.0938 %

Tabla 4.14: Métricas Producto 5 NN.

En la figura 4.20 se muestra la comparación entre los valores reales de venta y los pronosticados. Se nota una evidente falta de ajuste y aparentemente se mantiene estable el pronóstico con respecto al tiempo.

A manera de resumen sobre la aplicación de la Red Neuronal, la tabla 4.15 muestra los resultados obtenidos en las métricas para cada uno de los productos. Se observa que ningún producto estuvo dentro del margen de precisión de 90 %, sin embargo, el más cercano a este objetivo fue el producto 5.

Producto	Métrica	
Producto 1	MSE	851,615,764.88
	MAE	28,877.13
	EVS	-0.67
	MAPE	81.8952 %
Producto 2	MSE	13,041,407.44
	MAE	3,545.49
	EVS	-0.08
	MAPE	89.9513 %
Producto 3	MSE	348,730.54
	MAE	544.45
	EVS	-0.13
	MAPE	48.1218 %
Producto 4	MSE	45,406,778.03
	MAE	4,352.37
	EVS	0.38
	MAPE	28.7071 %
Producto 5	MSE	5,476,009.70
	MAE	1,911.46
	EVS	-0.14
	MAPE	12.0938 %

Tabla 4.15: Resumen de resultados para Red Neuronal de una capa oculta con una neurona en los cinco productos.

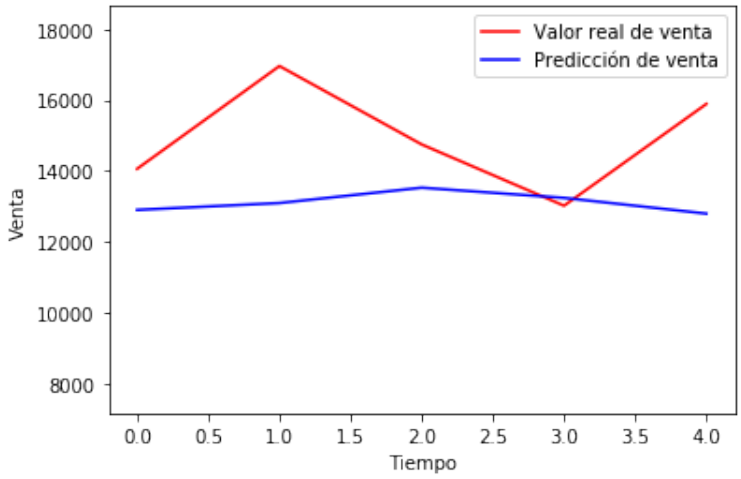


Figura 4.20: Predicciones del NN contra los valores reales observados de ventas en los últimos cinco periodos de tiempo para el Producto 5.

4.3.3. Red neuronal multicapa

La red neuronal empleada se caracteriza por tener dos capas ocultas del tipo *Dense* empleando la función de activación $ReLU$ y 50 neuronas en cada una de estas capas, todas las capas empleadas se conectan entre sí de manera secuencial. Se tiene en la capa de salida a una sola neurona del tipo *Dense* también. En la compilación del modelo se usó el optimizador *Adam* y el parámetro de pérdida se fijó en el error cuadrático medio. Se usó para el ajuste un tamaño de lote de 16 con 20 iteraciones. Todas estas configuraciones están disponibles en la librería de `keras` y corre sobre la plataforma `TensorFlow 2.0`.

La aplicación de esta red neuronal multicapa al producto 1 proporciona mejores resultados en comparación con la NN de la sección anterior. En la tabla 4.16 se describen las métricas obtenidas para todos los productos. Para el producto 1 se destaca el MAPE de 10.42 % que es casi 8 veces menor comparado con el encontrado en la sección anterior para este mismo producto. Además, se tiene una mejora del EVS con respecto a lo encontrado en la sección anterior y drásticas reducciones en las métricas MSE y MAE. La figura 4.21(a) muestra la comparación entre los valores reales y los predecidos, si bien se aprecia un ajuste en algunos puntos temporales donde coinciden ambas líneas, también existe un desajuste para los últimos dos meses.

En comparación con la NN aplicada en la anterior sección, el esquema de dos capas ocultas con 50 neuronas fue capaz de mejorar todas las métricas de evaluación para el prodcuto 2. En la tabla 4.16 se describen dichas métricas y si

bien este modelo no alcanzó a estar dentro del objetivo de precisión, no se quedó muy alejado ya que el MAPE obtenido fue de poco más de 15 %. En la figura 4.21(b) se muestra la diferencia entre la predicción y los valores reales. Es de destacar el desfase ocurrido hacia el último mes en donde el valor real superó por mucho el pronosticado. Sin embargo, se mantiene una cercanía significativa para los tres primeros meses.

En la comparación de la aplicación de esta NN con la aplicada en la sección anterior se nota una diferencia en sus métricas como lo indica la tabla 4.16. En específico se redujo en casi 20 % el MAPE y en casi la mitad el MSE. Se puede decir que para este producto la NN propuesta en esta sección ofrece mejores resultados. En la figura 4.21(c) se muestra el comportamiento de la predicción empleando la NN en contraste con los valores reales. Es de destacarse la diferencia en la observación del primer periodo, en donde se registran alrededor de 800 unidades de venta entre el valor real y el pronosticado.

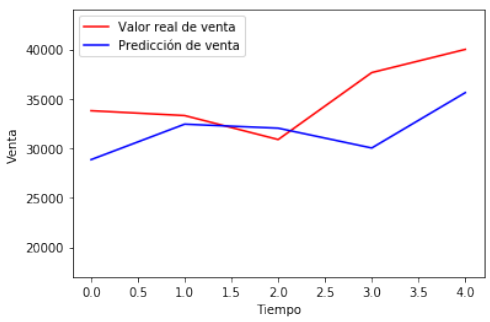
En la aplicación de la NN para el producto 4 en comparación con la sección anterior, se observa una caída en el rendimiento del MSE y un incremento en su EVS. Por otro lado, su MAE y MAPE se mantienen similares a los valores encontrados en la sección anterior. La tabla 4.16 muestra el resultado obtenido en las métricas de evaluación para este producto. La figura 4.21(d) muestra el comportamiento observado entre los valores reales de venta y los predecidos mediante esta NN. Se puede observar la diferencia en el último periodo temporal, en donde la predicción no se acerca al valor real.

Para la aplicación de esta NN en el producto 5, no se encontraron mejoras significativas en los resultados de las métricas. La tabla 4.16 contiene los valores de las métricas obtenidos tras la ejecución de la NN. La figura 4.21(e) muestra gráficamente la diferencia entre los valores de venta reales y los pronosticados. Se aprecian diferencias significativas en el momento temporal uno y el último, lo cual explica en gran medida, los resultados numéricos de las métricas.

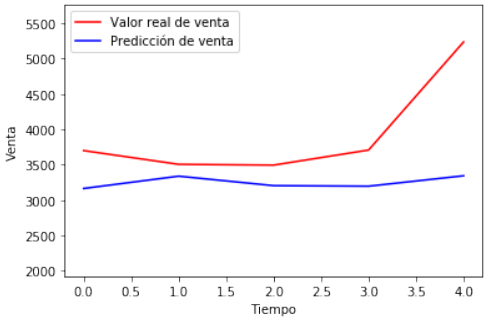
A manera de resumen sobre la aplicación de la Red Neuronal Multicapa, se observa que ningún producto estuvo dentro del margen de precisión de 90 %, sin embargo, el más cercano a este objetivo fue el producto 1.

Producto	Métrica	
Producto 1	MSE	20,748,241.43
	MAE	3,791.69
	EVS	0.09
	MAPE	10.4266 %
Producto 2	MSE	847,984.96
	MAE	679.28
	EVS	0.11
	MAPE	15.4988 %
Producto 3	MSE	164,529.43
	MAE	330.75
	EVS	-0.77
	MAPE	28.5397 %
Producto 4	MSE	54,984,858.67
	MAE	4,592.18
	EVS	0.27
	MAPE	28.1423 %
Producto 5	MSE	5,940,161.78
	MAE	2,046.12
	EVS	-1.05
	MAPE	13.0958 %

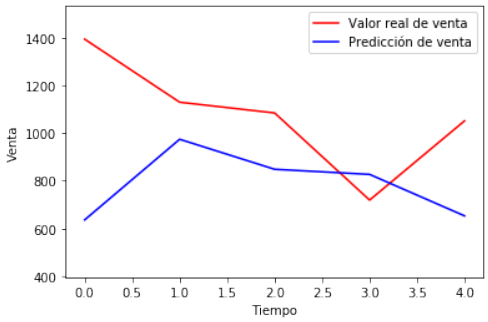
Tabla 4.16: Resumen de resultados para Red Neuronal Multicapa en los cinco productos.



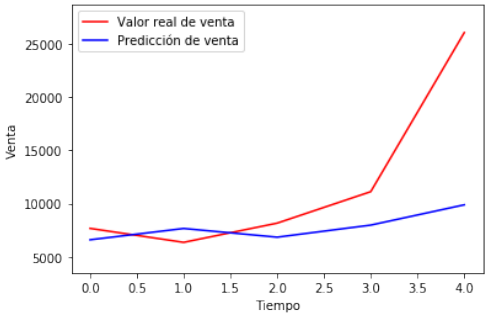
(a) Producto 1



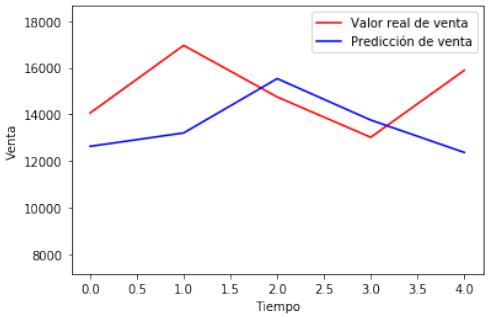
(b) Producto 2



(c) Producto 3



(d) Producto 4



(e) Producto 5

Figura 4.21: Predicciones del NN Multicapa contra los valores reales de venta observados en los últimos cinco periodos temporales.

4.3.4. Red neuronal LSTM con 50 neuronas

La configuración elegida para esta Red Neuronal LSTM es un modelo secuencial con una capa LSTM de 50 neuronas, seguida de una capa del tipo *Dense* con una neurona de salida. Se empleó el optimizador *rmsprop*, se empleó la métrica de pérdida para el entrenamiento de error cuadrático medio. Finalmente, el modelo

se entrenó con un tamaño de lote de 5 y 100 iteraciones. Dichas configuraciones están disponibles en la librería de `keras` corriendo sobre `TensorFlow 2.0.`

El comportamiento en métricas de la aplicación de esta NN aplicada al producto 1 se muestra en la tabla 4.22. En comparación con el resultado obtenido en la sección anterior, se observa una mejora significativa en todas sus métricas. Es de destacar el valor obtenido en la MAPE, la cual, se encuentra dentro del objetivo de precisión. La figura 4.22(a) muestra el comportamiento gráfico de los valores de venta predecidos en comparación con los valores reales. Se observa un buen ajuste entre ambas líneas con poca diferencia en sus observaciones temporales. Esto es congruente con los resultados numéricos de las métricas.

La aplicación de esta NN al producto produjo los resultados de la tabla 4.22 en términos de sus métricas de evaluación. Se observa, en comparación con la sección anterior, una reducción en el MSE. Sin embargo el MAPE tuvo in incremento de poco más de un punto porcentual, por lo que sigue estando fuera del objetivo establecido en esta tesina. El comportamiento gráfico de la comparación entre los valores reales y los pronosticados se muestran en la figura 4.22(b). A pesar de la diferencia entre los valores temporales de cada una de las líneas, es posible notar un indicio de ajuste en la curva de predicción.

En el caso de este producto, la tabla 4.22 muestra mejoras significativas en comparación con los valores hallados en la sección anterior para sus métricas MSE y EVS. Sin embargo, el MAPE y el MAE quedaron prácticamente iguales. La figura 4.22(c) muestra el comportamiento gráfico entre los valores de venta reales y predecidos. Se observa una gran diferencia entre ambas líneas de venta congruente con los resultados numéricos de evaluación.

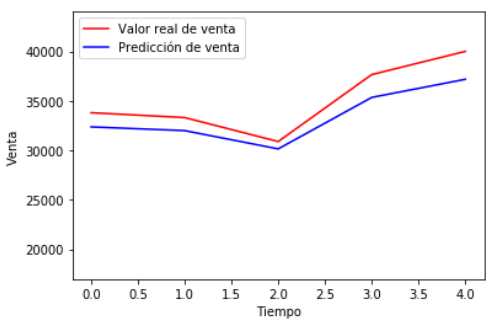
El producto 4 obtuvo en sus métricas de evaluación los resultados de la tabla 4.22. En comparación con los resultados obtenidos en la sección anterior, se observa una mejora en todas las métricas acercándose, incluso, al objetivo de precisión del MAPE. La figura 4.22(d) muestra el comportamiento observado entre el volumen de ventas real y el predecido. A pesar de la diferencia existente para el último periodo temporal, se observa un ajuste de la curva de predicción acorde al crecimiento real.

El producto 5 mostró una mejora significativa en sus métricas de evaluación con respecto a las mismas de la sección anterior. La tabla 4.22 agrupa los valores recogidos de las métricas para este producto, en donde el MAPE queda dentro del objetivo de precisión de establecido. La figura 4.22(e) muestra la diferencia gráfica entre los valores reales y los predecidos para las ventas del producto 5. Se observa un ajuste significativo en términos de la tendencia de la predicción, lo cual, es congruente con el valor de la métrica EVS obtenido previamente.

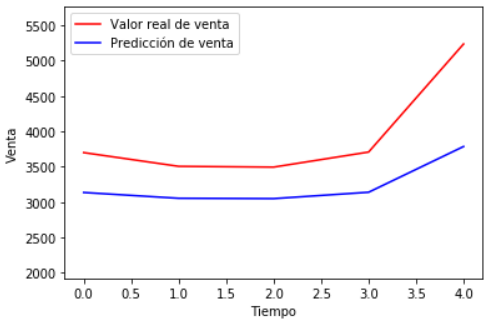
A manera de resumen sobre la aplicación de la Red Neuronal LSTM con 50 neuronas, se observa que únicamente los productos 1 y 5 estuvieron dentro del margen de precisión de 90 %.

Producto	**Métrica**	
Producto 1	MSE	3,528,624.58
	MAE	1,725.48
	EVS	0.95
	MAPE	4.7591 %
Producto 2	MSE	628,955.47
	MAE	695.80
	EVS	0.67
	MAPE	16.7805 %
Producto 3	MSE	138,449.96
	MAE	328.98
	EVS	0.35
	MAPE	28.2430 %
Producto 4	MSE	12,458,818.89
	MAE	2,132.83
	EVS	0.84
	MAPE	12.3248 %
Producto 5	MSE	1,209,724.54
	MAE	1,075.23
	EVS	0.97
	MAPE	7.1174 %

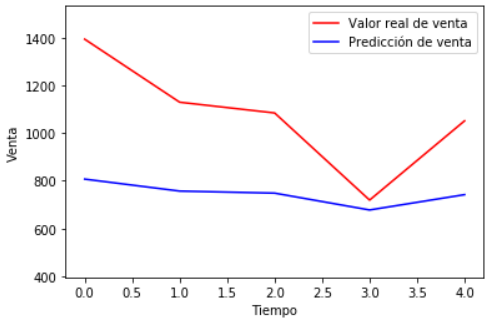
Tabla 4.17: Resumen de resultados para Red Neuronal LSTM con 50 neuronas en los cinco productos.



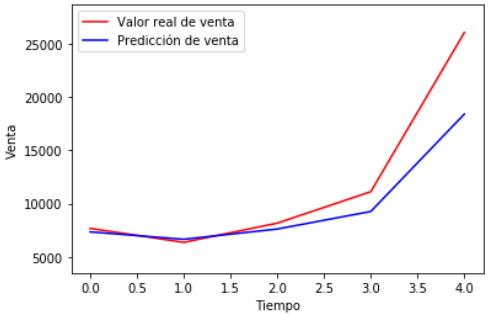
(a) Producto 1



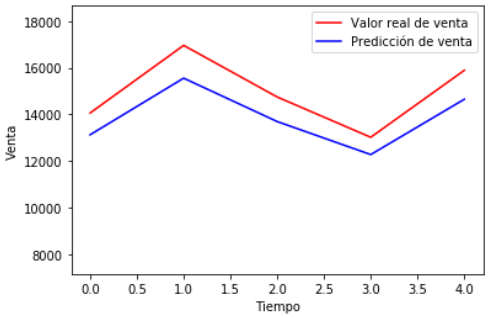
(b) Producto 2



(c) Producto 3



(d) Producto 4



(e) Producto 5

Figura 4.22: Predicciones del NN LSTM con 50 neuronas contra los valores reales de venta observados en los últimos cinco periodos temporales.

4.3.5. Red neuronal LSTM apilado

Para el funcionamiento de esta Red Neuronal, se configuró un modelo secuencial con dos capas LSTM, 50 neuronas y función de activación *relu* en cada capa. Además una capa del tipo *Dense* con una neurona como salida de la red. Se empleó el optimizador *rmsprop*, con el criterio de ajuste en pérdida del error

cuadrático medio, con tamaño de lote de 5 y 100 iteraciones en el entrenamiento. Del mismo modo que en la sección anterior, se hizo uso de la librería de `keras` sobre `TensorFlow 2.0`.

El desempeño de este modelo apilado para el producto 1 es notoriamente mejor en comparación con el modelo de la sección anterior. Las métricas registradas en la tabla 4.23 muestran una reducción significativa en el MSE, MAE y MAPE. La figura 4.23(a) muestra el comportamiento gráfico de los valores de venta reales contra los predecidos. Se observa un buen ajuste entre ambas líneas, conforme a los resultados numéricos obtenidos previamente.

El resultado obtenido en las métricas para el producto 2, muestra una mejora en comparación con los valores obtenidos en la sección anterior. La tabla 4.23 contiene el resultado de la evaluación de desempeño. El valor obtenido en el MAPE se acerca al objetivo de precisión de 90 %. El comportamiento gráfico de la predicción contra los valores de venta reales se muestra en la figura 4.23(b). Se observa una diferencia entre ambas líneas, sin embargo, la línea de predicción adquiere la forma de la línea de ventas reales.

El desempeño de esta NN aplicado al producto 3, no mostró mejora significativa en su evaluación de desempeño en comparación con lo observado en la sección anterior. La tabla 4.23 muestra los resultados de las métricas para este producto, mientras que la figura 4.23(c) muestra gráficamente, la comparación entre los valores reales y los predecidos.

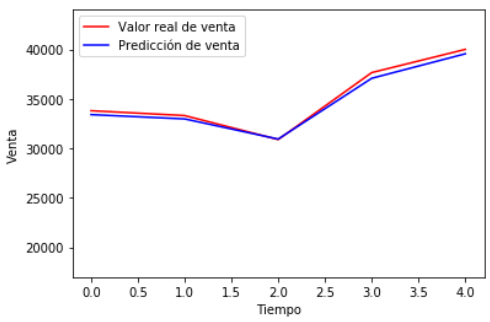
La aplicación de esta NN en el producto 4, mostró una mejora en comparación con lo observado en la sección anterior. La tabla 4.23 contiene los resultados de la evaluación de las métricas, el valor obtenido en el MAPE está dentro del objetivo de precisión. Por otro lado, en la figura 4.23(d) se observa la comparación gráfica entre los valores reales de venta contra los predecidos, en donde existe un buen ajuste de forma por parte de la línea de predicción.

La aplicación de esta NN en el producto 5 tuvo como resultado, en sus métricas, los valores mostrados en la tabla 4.23. En comparación con la NN aplicada en la sección anterior, se obtuvieron mejores resultados en todas las métricas. Resalta el valor de MSE que descendió en casi un millón de unidades. La figura 4.23(e) muestra el comportamiento y comparación gráfica entre los valores reales de venta y los predecidos. Se observa un ajuste entre ambas con poca diferencia pero coincidencia en la forma de las líneas.

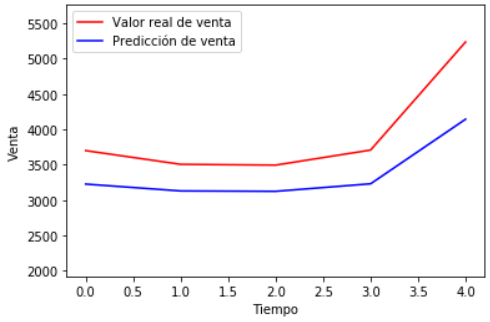
A manera de resumen sobre la aplicación de la Red Neuronal LSTM apilada, se observa que únicamente los productos 1 y 5 estuvieron dentro del margen de precisión de 90 %.

Producto	Métrica	
Producto 1	MSE	160,282.72
	MAE	360.02
	EVS	1.00
	MAPE	0.9950 %
Producto 2	MSE	384,622.17
	MAE	557.85
	EVS	0.83
	MAPE	13.5732 %
Producto 3	MSE	138,839.94
	MAE	328.00
	EVS	0.32
	MAPE	28.0748 %
Producto 4	MSE	1,270,427.75
	MAE	930.44
	EVS	0.99
	MAPE	8.0454 %
Producto 5	MSE	276,106.98
	MAE	371.54
	EVS	0.91
	MAPE	2.3110 %

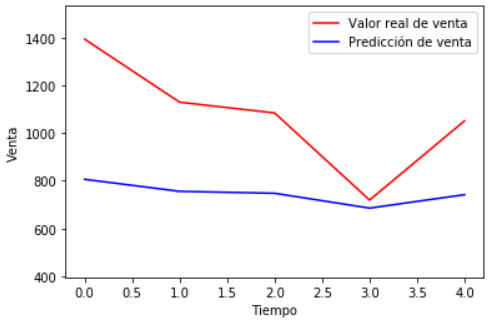
Tabla 4.18: Resumen de resultados para Red Neuronal LSTM apilado en los cinco productos.



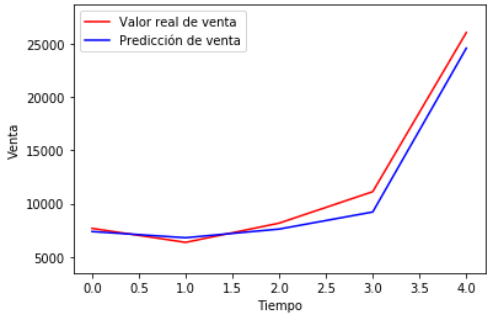
(a) Producto 1



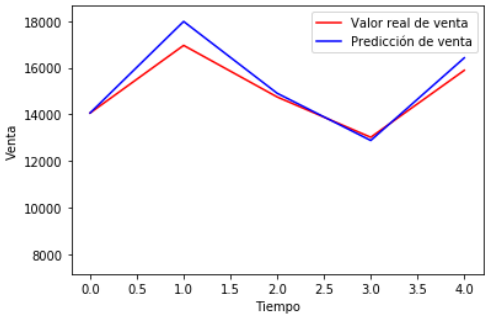
(b) Producto 2



(c) Producto 3



(d) Producto 4



(e) Producto 5

Figura 4.23: Predicciones de la NN LSTM apilado contra los valores reales de venta observados en los últimos cinco periodos temporales.

4.3.6. Red neuronal LSTM bidireccional

Esta Red Neuronal se configuró mediante un modelo secuencial, agregando una capa LSTM bidireccional con 50 neuronas y haciendo uso de la función de activación *relu*; además, se agregó una capa del tipo *dense* con una neurona como capa de salida. El optimizador elegido fue el *rmsprop* con el error cuadrático medio

como métrica de evaluación. Finalmente, se ajustó el modelo con 100 iteraciones y un tamaño de lote de 5. Al igual que en la sección anterior, se empleó la librería de `keras` sobre `TensorFlow 2.0`.

El desempeño de este modelo bidireccional LSTM para el producto 1 mostró un rendimiento inferior en comparación con el modelo de la sección anterior. Las métricas recolectadas en la tabla 4.19 muestran los valores numéricos de las métricas de evaluación. Sin embargo, éste modelo sí queda dentro del objetivo de precisión al tener obtener en el MAPE un valor de 1.9608 %. La figura muestra el comportamiento gráfico de la predicción contra los valores reales de venta. Es posible observar un buen ajuste por parte de la línea de predicción acorde con los valores numéricos obtenidos en las métricas.

La NN LSTM del tipo bidireccional aplicada a el producto 2 tuvo como resultado en sus métricas, un descenso en la calidad de la predicción realizada. Al observar los valores de la tabla 4.19, la precisión predictiva se alejó en su MAPE al 15.15 %, lo cual, lo aleja del objetivo en comparación con la sección anterior. En la figura 4.24(b) se muestra el comportamiento gráfico de los valores pronosticados contra los reales.

En el caso del producto 3, la aplicación de esta NN no mostró una mejora significativa con respecto a los valores numéricos obtenidos en la sección anterior. Se obtuvieron resultados similares en las métricas de evaluación con respecto a la sección anterior y la tabla 4.19 contiene dichos resultados. La figura 4.24(c) muestra la comparación entre los valores reales de venta y los pronosticados. Se observa un comportamiento similar al reportado por la NN aplicada en la sección anterior.

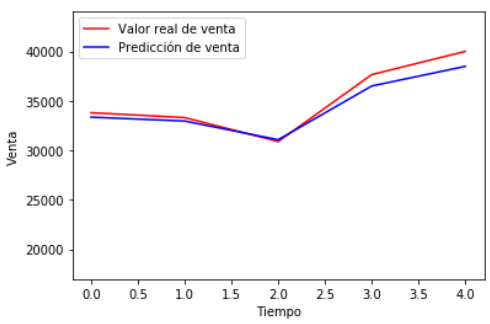
La aplicación de la NN LSTM bidireccional en el producto 4 tuvo los resultados mostrados en la tabla 4.19. Aunque su MAPE estuvo muy cerca del objetivo establecido previamente, no se encontró una mejora en comparación con los valores obtenidos en la sección anterior. La figura 4.24(d) informa el comportamiento de los valores reales de venta contra los pronosticados. Se observa un buen ajuste de la línea de predicción, sin embargo, ésta comienza a alejarse en los dos últimos periodos temporales.

La aplicación de esta NN al producto 5, tuvo como resultados los números reportados en la tabla 4.19. La métrica MAPE está dentro del objetivo establecido de precisión. Sin embargo, no se tienen mejoras en comparación con los hallazgos de la sección anterior. Por otro lado, la figura 4.24(e) muestra la comparación gráfica entre los valores reales y los pronosticados. Se observa un buen ajuste de la línea de predicción, la cual, semeja al comportamiento real de ventas sin caer en los mismos valores.

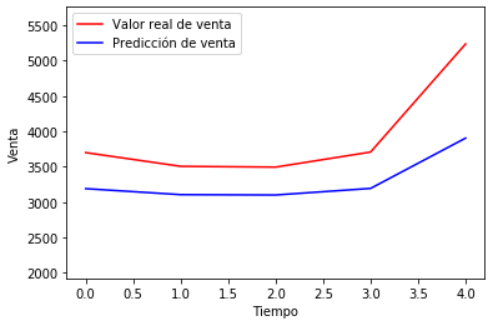
A manera de resumen sobre la aplicación de la Red Neuronal LSTM bidireccional, únicamente los productos 1 y 5 estuvieron dentro del margen de precisión de 90 %.

Producto	Métrica	
Producto 1	MSE	794,606.43
	MAE	728.82
	EVS	0.97
	MAPE	1.9608 %
Producto 2	MSE	522,265.49
	MAE	629.88
	EVS	0.71
	MAPE	15.1559 %
Producto 3	MSE	136,193.80
	MAE	323.12
	EVS	0.31
	MAPE	27.5744 %
Producto 4	MSE	9,991,866.62
	MAE	1,923.61
	EVS	0.87
	MAPE	11.3609 %
Producto 5	MSE	910,709.33
	MAE	932.85
	EVS	0.98
	MAPE	6.1746 %

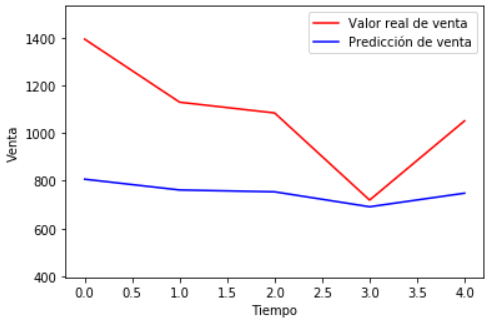
Tabla 4.19: Resumen de resultados para Red Neuronal LSTM bidireccional en los cinco productos.



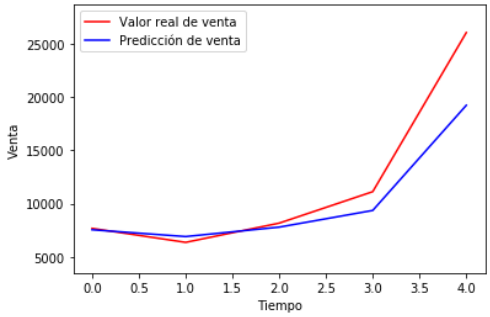
(a) Producto 1



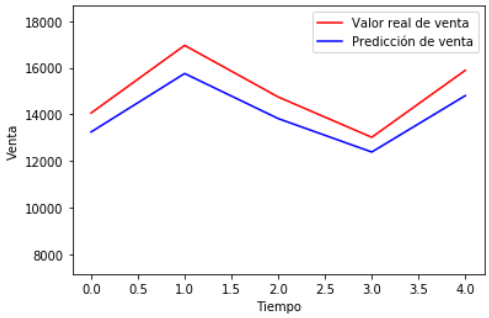
(b) Producto 2



(c) Producto 3



(d) Producto 4



(e) Producto 5

Figura 4.24: Predicciones la NN LSTM Bidireccional contra los valores reales de venta observados en los últimos cinco periodos temporales.

4.3.7. Algoritmo XGBoost

Este algoritmo se configuró mediante la función `XGBRegressor` de la librería `xgboost` disponible para Python. El objetivo en el entrenamiento de este modelo, fue minimizar el error cuadrático.

Los resultados de la aplicación de este algoritmo al producto 1, están contenidas en la tabla 4.20. Gracias al valor obtenido en la métrica MAPE, se puede confirmar que este modelo cumple con el objetivo de precisión. En la figura 4.25(a) se aprecia el comportamiento gráfico comparando los valores reales de venta contra los predecidos.

En la evaluación de este algoritmo para el producto 2, se obtuvieron los resultados en las métricas de evaluación, representados en la tabla 4.20. Al analizar el valor de su MAPE, se confirma que éste algoritmo no cumple con el objetivo inicial de predicción de 90 %. En la figura 4.25(b), se muestra el comportamiento gráfico de los valores reales de venta en contra de los predecidos por el algoritmo. Se destaca la falta de ajuste a la tendencia de la línea de predicción y fuertes diferencias en el periodo inicial, así como en el final.

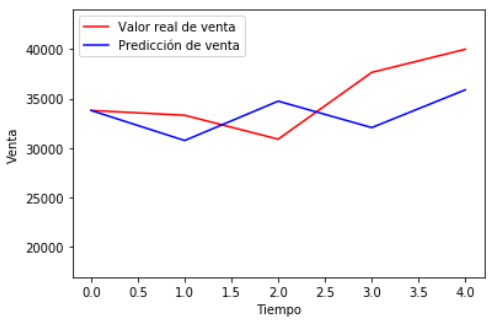
En el caso del producto 3, la evaluación de sus métricas se registraron en la tabla 4.20. En la métrica MAPE, el algoritmo XGBoost quedó alejado del objetivo de precisión definido previamente. La figura 4.25(c) muestra la comparación gráfica entre los valores reales de venta contra los predecidos para este producto.

La tabla 4.20 muestra los resultados de las evaluaciones realizadas al desempeño de este algoritmo aplicado al producto 4. Se observa un valor en la métrica MAPE, alejado del objetivo de precisión establecido. La figura 4.25(d) muestra el comportamiento gráfico de los valores reales de venta en comparación con los pronosticados. Se observa una gran diferencia entre ambas líneas de venta, específicamente en el último periodo temporal.

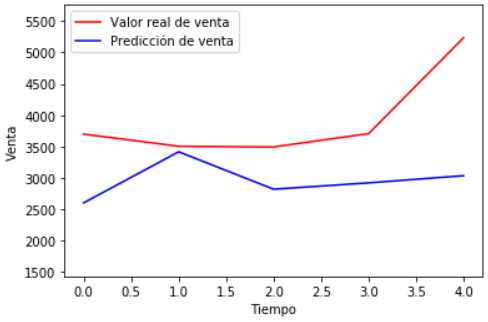
La tabla 4.20 contiene la información de los resultados de evaluación de las métricas del algoritmo XGBoost aplicado al producto 5. Llama la atención, el resultado del MAPE que está muy alejado del objetivo establecido de precisión. La figura 4.25(e) muestra gráficamente el comportamiento de la línea real de ventas y de la línea de predicción. Se observa una diferencia significativa entre ambas, específicamente un comportamiento a la baja para la línea de predicción que no coincide con el comportamiento real.

Producto	Métrica	
Producto 1	MSE	13,920,882.57
	MAE	3,224.46
	EVS	-0.04
	MAPE	9.1268 %
Producto 2	MSE	1,422,133.76
	MAE	967.97
	EVS	-0.11
	MAPE	22.8832 %
Producto 3	MSE	155,750.09
	MAE	326.23
	EVS	-0.07
	MAPE	30.5120 %
Producto 4	MSE	79,431,021.57
	MAE	5,852.37
	EVS	-0.36
	MAPE	33.3748 %
Producto 5	MSE	51,937,633.41
	MAE	6,853.80
	EVS	-1.62
	MAPE	45.4317 %

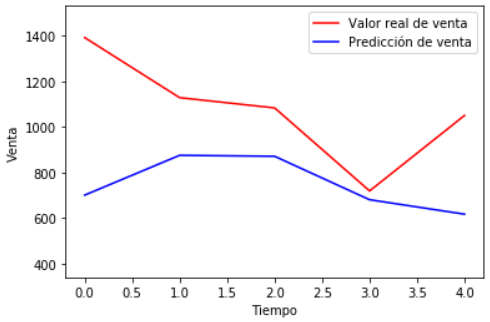
Tabla 4.20: Resumen de resultados para XGBoost en los cinco productos.



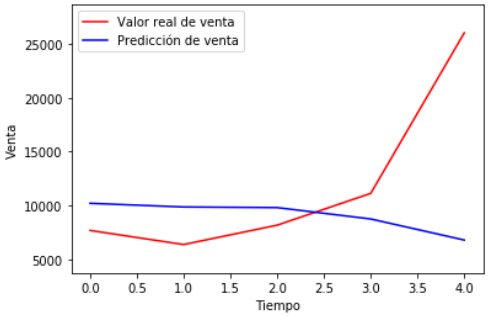
(a) Producto 1



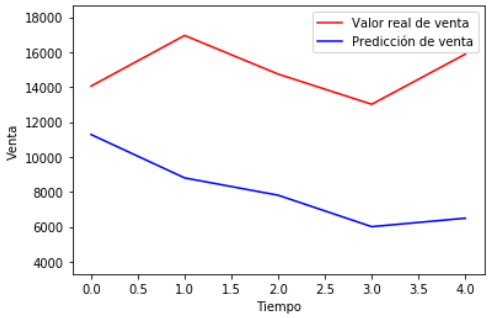
(b) Producto 2



(c) Producto 3



(d) Producto 4



(e) Producto 5

Figura 4.25: Predicciones del algoritmo XGBoost contra los valores reales de venta observados en los últimos cinco periodos temporales.

4.3.8. CNN LSTM

Este algortimo se configuró haciendo uso de la librería de `Keras` en sus funciones para creación de capas `Sequential, TimeDistributed, Conv1D, LSTM, Dense` y `Flatten`. El objetivo del entrenamiento fue reducir el error cuadrático medio.

En la tabla 4.21 se muestran los resultados de las métricas obtenidos en la aplicación de este modelo al producto 1. Con el valor encontrado de la métrica MAPE, se observa que el modelo cumple con el objetivo de precisión. La figura 4.26(a) muestra el comportamiento de la línea de ventas real y la línea de predicción. Se observa un buen ajuste a lo largo de ambas líneas con la diferencia del último periodo temporal, en el que se separan pero se mantiene la tendencia a la alza.

En la aplicación del modelo para el producto 2, se obtuvieron los resultados de las métricas de evaluación descritos en la tabla 4.21. La métrica MAPE se encuentra alejada del objetivo de precisión por un margen de catorce puntos porcentuales. La figura 4.26(b) muestra el comportamiento gráfico de las líneas de ventas, tanto la real como la pronosticada. Se observa que la línea de pronóstico se mantiene sin ajustarse al comportamiento real.

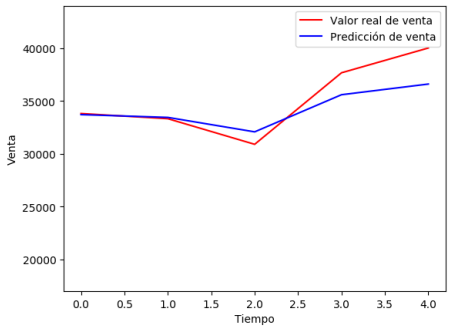
Los resultados obtenidos de la aplicación de este modelo al producto 3 se muestran en la tabla 4.21. Se aprecia un distanciamiento de casi 22 puntos porcentuales en la métrica MAPE, semejante a lo encontrado en los modelos XGBoost y SVR. Similar al producto anterior, se observa en la figura 4.26(c), un comportamiento de la línea de predicción sin ajuste a la línea de ventas real.

La aplicación de este modelo de ML al producto 4, tuvo como resultado en la evaluación de las métricas, lo reportado en la tabla 4.21. La métrica MAPE está muy cercana al objetivo de precisión, sin embargo el MSE está muy elevado en comparación del modelo LSTM Bidireccional o el LSTM de 50 neuronas. En la comparación gráfica de la figura 4.26(d), se observa el ajuste de tendencia por parte de la línea de predicción. Sin embargo, existe un desajuste en términos de precisión de ventas para el último periodo temporal.

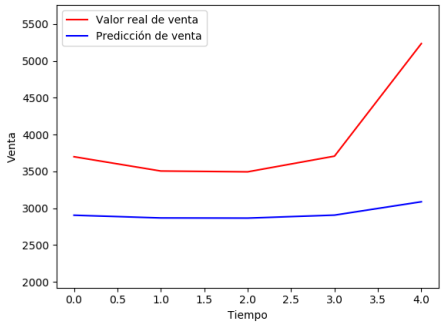
La evaluación de las métricas para la aplicación de este modelo en el producto 5, muestra sus resultados en la tabla 4.21. Dado que el valor de la métrica MAPE es de 17 puntos porcentuales, el modelo no cumple con los objetivos de precisión. En la figura 4.26(e) se describe la comparación gráfica entre los valores de venta reales y los pronosticados. Se observa que la línea de predicción se ajusta en la tendencia, pero no en la precisión de los valores de venta.

Producto	Métrica	
Producto 1	MSE	3,472,996.37
	MAE	1,378.96
	EVS	0.74
	MAPE	3.7079 %
Producto 2	MSE	1,336,871.32
	MAE	1,001.59
	EVS	0.23
	MAPE	24.0512 %
Producto 3	MSE	177,396.40
	MAE	374.71
	EVS	0.20
	MAPE	32.3060 %
Producto 4	MSE	16,019,940.69
	MAE	2,356.03
	EVS	0.79
	MAPE	13.3151 %
Producto 5	MSE	7,567,311.42
	MAE	2,661.09
	EVS	0.74
	MAPE	17.5400 %

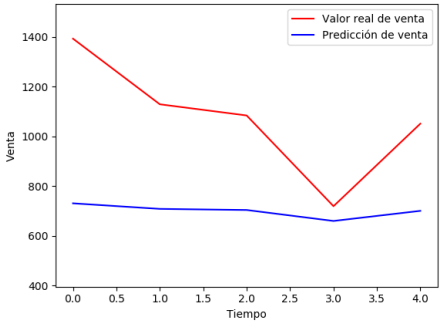
Tabla 4.21: Resumen de resultados para CNN LSTM en los cinco productos.



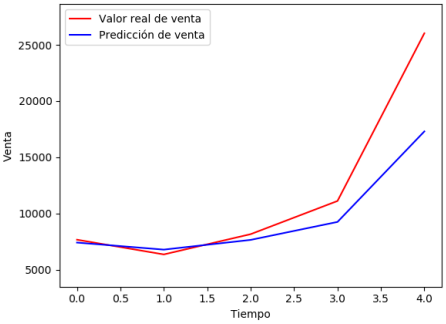
(a) Producto 1



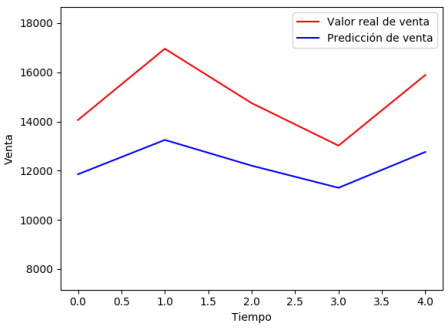
(b) Producto 2



(c) Producto 3



(d) Producto 4



(e) Producto 5

Figura 4.26: Predicciones del CNN LSTM contra los valores reales de venta observados en los últimos cinco periodos temporales.

4.3.9. Conv LSTM

Este modelo fue configurado recurriendo al uso de la librería `Keras` en sus funciones `Sequential, ConvLSTM2D, Flatten` y `Dense`. El objetivo del entrenamiento fue reducir el error cuadrático medio. La aplicación de este modelo para el producto 1 tuvo un resultado similar en su evaluación de métricas en comparación con lo obtenido en la sección anterior. Sin embargo, este modelo mostró un mejor desempeño en la métrica MSE en comparación con dicha sección. La tabla 4.22 muestra los resultados de la evaluación del modelo en sus métricas. El comportamiento gráfico de la línea de predicción en comparación con la línea real de ventas se muestra en la figura 4.27(a). Se observa un buen ajuste entre ambas líneas con una separación hacia el último periodo temporal.

Para el producto 2 se obtuvieron los resultados de la evaluación de métricas mostrados en la tabla 4.22. A pesar de no estar dentro del objetivo de precisión en la métrica MAPE, este modelo tuvo una mejora significativa en comparación con el modelo de la sección anterior. Por otro lado, la figura 4.27(b) muestra el comportamiento gráfico de las líneas de predicción y de venta real. A diferencia de lo ocurrido en la sección anterior, en este modelo se ajusto ligeramente mejor a la tendencia hacia el último periodo temporal.

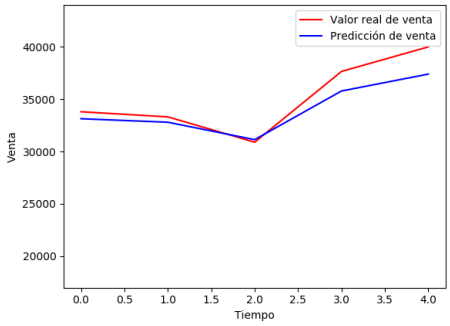
La aplicación de este modelo a los datos del producto 3 tuvo un resultado similar al modelo de la sección anterior. En la tabla 4.22 se muestran los resultados de la evaluación de las métricas obtenidas para este modelo. La figura 4.27(c) muestra el comportamiento gráfico de la línea de predicción y la línea de ventas real. El comportamiento de la predicción es similar a lo obtenido en la sección anterior.

La aplicación de este modelo al producto 4 tuvo como resultados de su evaluación en las métricas definidas, lo encontrado en la tabla 4.22. Se observa que el desempeño de la métrica MAPE estuvo cercano al objetivo de precisión y, en comparación con los resultados de la sección anterior, se obtuvieron mejores resultados para todas las métricas. La figura 4.27(d) muestra el comportamiento gráfico de las líneas de predicción y ventas real. Se distingue un buen ajuste en la tendencia de la predicción con un ligero desfase hacia el último periodo temporal.

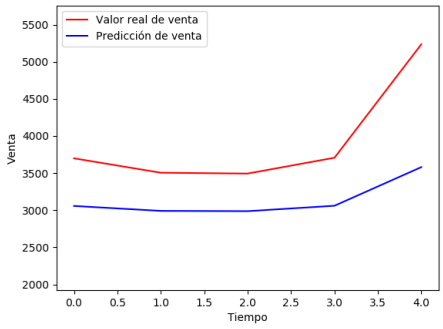
Este modelo aplicado al producto 5 tuvo los resultados de evaluación de métricas mostrados en la tabla 4.22. La métrica MAPE estuvo cercana al objetivo de predicción, además, se tuvo una mejora en el resto de las métricas en comparación con el modelo de la sección anterior. La comparación gráfica entre los valores reales de venta y los predecidos se muestran en la figura. Se observa un buen ajuste entre ambas líneas pero con diferencias en todos los periodos temporales.

Producto	Métrica	
Producto 1	MSE	2,208,614.16
	MAE	1,180.32
	EVS	0.90
	MAPE	3.1582 %
Producto 2	MSE	817,434.67
	MAE	792.56
	EVS	0.57
	MAPE	19.1087 %
Producto 3	MSE	183,824.66
	MAE	385.47
	EVS	0.24
	MAPE	33.4400 %
Producto 4	MSE	11,189,307.05
	MAE	1,933.81
	EVS	0.84
	MAPE	10.7845 %
Producto 5	MSE	2,691,845.38
	MAE	1,586.19
	EVS	0.91
	MAPE	10.4518 %

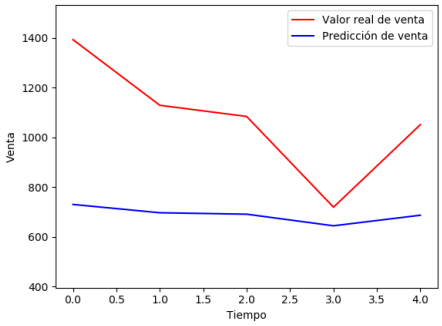
Tabla 4.22: Resumen de resultados para Conv LSTM en los cinco productos.



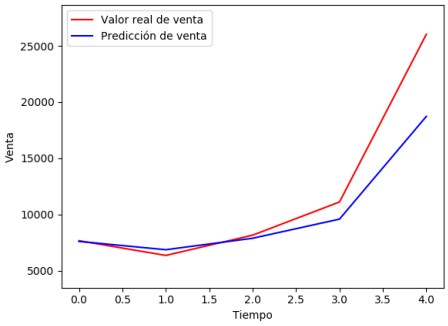
(a) Producto 1



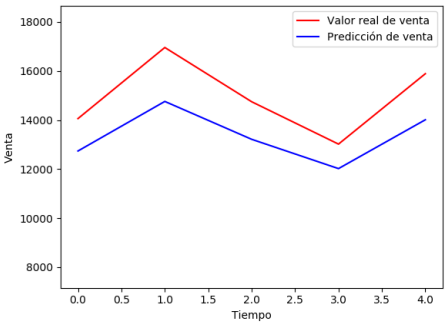
(b) Producto 2



(c) Producto 3



(d) Producto 4



(e) Producto 5

Figura 4.27: Predicciones del Conv LSTM contra los valores reales de venta observados en los últimos cinco periodos temporales.

4.4. Aplicación de algoritmos de ML y metaheurísticos

Para la implementación de los algoritmos metaheurísticos con los algoritmos de ML, se siguió la siguiente metodología:

1. Lectura de los registros de venta por producto.
2. Estandarización de los datos.
3. Transformación de serie temporal a aprendizaje supervisado.
4. Implementación de los algoritmos metaheurísticos y la ejecución de los modelos de ML como funciones.
5. Evaluación de los hiperparámetros obtenidos en cada iteración.
6. Selección del mejor conjunto de hiperparámetros para cada modelo de ML en cada ejecución del correspondiente metaheurístico.
7. Transformación inversa de los datos predecidos para su manipulación en los términos originales.

Para un análisis inicial de los resultados, el estudio se concentró en la evaluación del desempeño de cada modelo de ML ejecutado en cada algoritmo metaheurístico. De este modo, cada producto se evalúa en sus propios términos facilitando el futuro análisis estadístico. Ya que lo que se busca es ajustar los hiperparámetros a usar en los modelos de ML, cada uno de los metaheurísticos es configurado en sus propios parámetros de manera independiente, buscando en cada iteración la mejor combinación de hiperparámetros en los modelos de ML.

Los algoritmos metaheurísticos empleados fueron: búsqueda aleatoria simple, recocido simulado y estrategias evolutivas. Dichos algoritmos fueron seleccionados por los pocos parámetros a configurar para su funcionamiento. Por otro lado, en cada uno de los modelos de ML se buscó mejorar la precisión de las predicciones mediante el ajuste de ciertos hiperparámetros propios de cada algoritmo.

Los experimentos se realizaron agrupando como primer categoría al modelo de ML, la segunda categoría fue el algoritmo metaheurístico a usar, y por último, el producto al cual se le aplicó la combinación de las primeras dos categorías. En resumen, por cada modelo de ML se realizaron 15 experimentos en donde se evaluó el desempeño de cada combinación en términos de las métricas previamente establecidas. Los resultados numéricos de estos experimentos se muestran en el rango de tablas 4.23-4.31.

Conjunto de datos	Métrica	Búsqueda Aleatoria	Recocido Simulado	Estrategias Evolutivas
Producto 1	MSE	4,606,957.32	3,313,919.24	3,768,635.43
	MAE	1,831.58	1,469.71	1,813.50
	EVS	0.88	0.89	0.76
	MAPE	4.9655 %	3.9384 %	5.3122 %
Producto 2	MSE	801,595.92	1,954,419.45	1,954,419.45
	MAE	737.97	1,232.50	1,232.50
	EVS	-0.84	0.00	0.00
	MAPE	17.4705 %	29.7748 %	29.7748 %
Producto 3	MSE	57,540.38	164,760.20	164,760.20
	MAE	225.37	349.00	349.00
	EVS	0.85	0.00	0.00
	MAPE	20.4968 %	29.4913 %	29.4913 %
Producto 4	MSE	7,498,801.53	56,531,813.40	2,660,924.12
	MAE	2,041.93	4,960.60	1,394.86
	EVS	0.88	0.00	0.96
	MAPE	15.9392 %	35.5505 %	17.9946 %
Producto 5	MSE	1,991,384.11	5,269,152.42	3,357,006.08
	MAE	1,285.75	2,229.45	1,717.56
	EVS	0.31	0.84	0.79
	MAPE	8.9573 %	14.7172 %	11.9541 %

Tabla 4.23: Resultados de la mejora del modelo Soporte de Regresión Vectorial.

Conjunto de datos	Métrica	Búsqueda Aleatoria	Recocido Simulado	Estrategias Evolutivas
Producto 1	MSE	9765841.32	9332187.75	10615051.88
	MAE	2834.83	2689.76	2820.89
	EVS	0.08	0.13	0.13
	MAPE	8.2299 %	7.6324 %	7.8343 %
Producto 2	MSE	376653.07	370596.09	384806.87
	MAE	547.45	401.70	405.42
	EVS	0.20	0.18	0.18
	MAPE	13.6924 %	9.0805 %	9.0718 %
Producto 3	MSE	105041.19	74424.91	85608.56
	MAE	263.53	245.07	242.94
	EVS	-0.75	0.05	-0.69
	MAPE	24.4138 %	22.2143 %	25.1347 %
Producto 4	MSE	32862563.61	28599176.20	34882664.21
	MAE	4037.13	3577.13	4558.00
	EVS	0.58	0.57	0.34
	MAPE	29.3531 %	25.8569 %	39.9607 %
Producto 5	MSE	5082446.94	4913987.18	5187998.03
	MAE	1846.38	2064.79	2067.95
	EVS	-0.12	-1.50	-1.29
	MAPE	11.6923 %	13.6026 %	13.4371 %

Tabla 4.24: Resultados de la mejora del modelo Red Neuronal de una capa y una neurona.

Conjunto de datos	Métrica	Búsqueda Aleatoria	Recocido Simulado	Estrategias Evolutivas
Producto 1	MSE	10,792,134.00	9,472,878.02	9,842,090.47
	MAE	2,807.19	2,829.09	2,786.55
	EVS	-0.01	0.11	0.08
	MAPE	8.2426 %	8.1421 %	8.1963 %
Producto 2	MSE	811,439.54	744,080.35	603,912.56
	MAE	626.74	772.74	443.09
	EVS	0.04	0.33	0.05
	MAPE	14.0363 %	18.8903 %	9.2827 %
Producto 3	MSE	108,466.98	71,116.14	117,837.27
	MAE	261.75	203.40	302.45
	EVS	-0.44	-0.35	-0.03
	MAPE	23.0320 %	19.3689 %	26.2749 %
Producto 4	MSE	39,704,491.12	36,470,814.50	26,000,914.75
	MAE	3,907.02	3,765.44	4,707.88
	EVS	0.30	0.43	0.56
	MAPE	27.7608 %	25.6520 %	48.6888 %
Producto 5	MSE	3,277,759.18	2,813,505.81	2,695,936.95
	MAE	1,549.25	1,531.98	1,416.70
	EVS	-0.50	-0.48	-0.38
	MAPE	10.0667 %	10.2264 %	9.3493 %

Tabla 4.25: Resultados de la mejora del modelo Red Neuronal multicapa.

Conjunto de datos	Métrica	Búsqueda Aleatoria	Recocido Simulado	Estrategias Evolutivas
Producto 1	MSE	621,798.46	422,258.86	424,943.04
	MAE	597.07	530.97	490.49
	EVS	0.95	0.98	0.97
	MAPE	1.6169 %	1.4326 %	1.3242 %
Producto 2	MSE	166,407.14	466,326.07	255,863.62
	MAE	326.41	595.54	279.08
	EVS	0.62	0.74	0.56
	MAPE	7.7195 %	14.3321 %	5.7856 %
Producto 3	MSE	54,320.79	92,654.31	113,629.76
	MAE	203.64	264.93	292.10
	EVS	0.61	0.36	0.39
	MAPE	17.4484 %	22.6218 %	24.7683 %
Producto 4	MSE	4,766,989.13	6,638,489.99	6,025,584.09
	MAE	1,452.81	1,591.68	1,861.65
	EVS	0.94	0.90	0.89
	MAPE	9.4726 %	9.7962 %	16.1574 %
Producto 5	MSE	362,421.90	482,116.62	463,836.09
	MAE	581.45	670.55	602.93
	EVS	0.99	0.98	0.77
	MAPE	3.8297 %	4.4163 %	4.1619 %

Tabla 4.26: Resultados de la mejora del modelo Red Neuronal LSTM con 50 neuronas

Conjunto de datos	Métrica	Búsqueda Aleatoria	Recocido Simulado	Estrategias Evolutivas
Producto 1	MSE	802,491.06	112,763.63	427,776.37
	MAE	708.44	307.40	554.75
	EVS	0.93	0.99	0.98
	MAPE	1.9527 %	0.8496 %	1.5033 %
Producto 2	MSE	406,980.49	366,415.12	407,694.22
	MAE	569.93	546.44	565.36
	EVS	0.81	0.84	0.80
	MAPE	13.8328 %	13.3130 %	13.6782 %
Producto 3	MSE	119,580.37	99,428.68	124,071.14
	MAE	304.22	270.06	307.74
	EVS	0.42	0.43	0.37
	MAPE	26.0284 %	22.7449 %	26.2228 %
Producto 4	MSE	716,615.14	564,755.04	456,353.20
	MAE	655.07	560.40	534.09
	EVS	0.99	0.99	0.99
	MAPE	6.4488 %	5.7348 %	5.3518 %
Producto 5	MSE	25,319.64	123,402.56	46,574.25
	MAE	145.09	342.30	191.04
	EVS	1.00	1.00	0.98
	MAPE	1.0156 %	2.3528 %	1.3182 %

Tabla 4.27: Resultados de la mejora del modelo Red Neuronal LSTM apilado.

Conjunto de datos	Métrica	Búsqueda Aleatoria	Recocido Simulado	Estrategias Evolutivas
Producto 1	MSE	527,844.71	714,356.29	289,499.11
	MAE	655.26	783.47	494.78
	EVS	0.95	0.93	0.97
	MAPE	1.8662 %	2.2623 %	1.4168 %
Producto 2	MSE	258,763.37	449,130.80	183,338.00
	MAE	501.82	582.28	235.40
	EVS	0.56	0.75	0.69
	MAPE	12.9254 %	13.9952 %	4.8725 %
Producto 3	MSE	117,725.48	57,154.36	32,684.84
	MAE	300.99	213.34	151.68
	EVS	0.41	0.24	0.40
	MAPE	25.7156 %	19.3975 %	14.5786 %
Producto 4	MSE	7,959,921.35	5,314,555.46	456,353.20
	MAE	1,686.81	1,587.53	534.09
	EVS	0.89	0.91	0.99
	MAPE	9.8132 %	11.7923 %	5.3518 %
Producto 5	MSE	354,507.29	107,730.70	359,111.68
	MAE	525.86	286.45	541.20
	EVS	0.82	0.95	0.97
	MAPE	3.5919 %	1.9992 %	3.8147 %

Tabla 4.28: Resultados de la mejora del modelo Red Neuronal LSTM bidireccional.

Conjunto de datos	Métrica	Búsqueda Aleatoria	Recocido Simulado	Estrategias Evolutivas
Producto 1	MSE	301,574,594.87	10,911,801.76	4,799,646.57
	MAE	16,716.26	2,908.06	1,887.15
	EVS	-1.08	0.02	0.57
	MAPE	47.1352 %	8.8990 %	9.3256 %
Producto 2	MSE	5,951,228.07	1,216,394.01	1,275,823.42
	MAE	2,338.40	958.37	966.67
	EVS	-0.11	0.32	0.22
	MAPE	58.5865 %	22.6331 %	22.8491 %
Producto 3	MSE	179,964.47	111,430.65	135,039.17
	MAE	362.10	305.93	292.89
	EVS	-0.06	0.61	-0.06
	MAPE	30.5670 %	30.6757 %	27.7612 %
Producto 4	MSE	115,965,177.12	81,528,812.38	54,919,071.19
	MAE	7,691.61	6,090.58	5,435.15
	EVS	-0.08	-0.23	-0.04
	MAPE	55.0332 %	32.1319 %	44.3598 %
Producto 5	MSE	105,732,293.33	2,701,040.82	3,796,274.37
	MAE	10,167.46	1,304.70	1,593.48
	EVS	-0.24	0.00	0.00
	MAPE	67.8100 %	8.3352 %	10.1067 %

Tabla 4.29: Resultados de la mejora del modelo XGBoost.

Conjunto de datos	Métrica	Búsqueda Aleatoria	Recocido Simulado	Estrategias Evolutivas
Producto 1	MSE	619,322.86	1,133,902.49	204,094.29
	MAE	730.92	977.12	381.57
	EVS	0.95	0.89	0.98
	MAPE	2.1308 %	2.7978 %	1.0685 %
Producto 2	MSE	662,191.20	712,725.46	884,296.23
	MAE	662.88	764.06	768.08
	EVS	0.49	0.70	0.32
	MAPE	15.6022 %	18.6344 %	18.0913 %
Producto 3	MSE	167,060.14	102,220.92	125,096.20
	MAE	360.07	278.58	309.89
	EVS	0.19	0.25	0.06
	MAPE	30.9164 %	23.8550 %	26.6300 %
Producto 4	MSE	6,264,234.47	13,409,955.73	3,842,565.50
	MAE	1,766.39	2,064.52	1,331.01
	EVS	0.89	0.80	0.95
	MAPE	13.6376 %	11.5832 %	8.8597 %
Producto 5	MSE	2,148,470.83	1,264,742.43	3,472,338.80
	MAE	1,349.16	1,069.71	1,823.09
	EVS	0.83	0.94	0.92
	MAPE	8.7578 %	7.0108 %	12.0724 %

Tabla 4.30: Resultados de la mejora del modelo CNN-LSTM.

Conjunto de datos	Métrica	Búsqueda Aleatoria	Recocido Simulado	Estrategias Evolutivas
Producto 1	MSE	265,883.83	344,678.95	165,494.66
	MAE	459.65	544.81	337.61
	EVS	0.98	0.97	0.99
	MAPE	1.2977 %	1.5888 %	0.9379 %
Producto 2	MSE	628,330.02	580,364.34	387,273.57
	MAE	697.53	557.94	404.84
	EVS	0.67	0.38	0.49
	MAPE	16.8402 %	12.7002 %	8.8747 %
Producto 3	MSE	94,677.66	70,604.25	144,372.54
	MAE	273.28	237.58	339.22
	EVS	0.10	0.13	0.37
	MAPE	23.9497 %	21.3909 %	29.2949 %
Producto 4	MSE	6,300,507.66	10,232,195.98	5,497,752.09
	MAE	1,556.30	1,958.58	1,537.23
	EVS	0.91	0.87	0.93
	MAPE	9.5215 %	11.4738 %	9.8692 %
Producto 5	MSE	975,613.44	1,302,463.80	2,045,466.42
	MAE	884.23	1,047.81	1,380.22
	EVS	0.90	0.89	0.93
	MAPE	5.6985 %	6.7953 %	9.0884 %

Tabla 4.31: Resultados de la mejora del modelo Conv-LSTM.

4.5. Análisis estadístico de los resultados

Con el fin de determinar las condiciones (factores) bajo cuáles se obtienen mejores predicciones de venta para los cinco productos estudiados, los resultados de las cuatro métricas de desempeño serán analizados mediante modelos estadísticos. Los factores a analizar para cada producto fueron: Algoritmo de predicción (9), tipo de optimización de hiperparámetros (4). La siguiente lista detalla los nombres de cada uno de estos factores y sirve como referencia para la codificación empleada en el análisis posterior.

- Modelo de ML
    1. Soporte de Regresión Vectorial.
    2. Red neuronal de una neurona y una capa oculta.
    3. Red neuronal de dos capas ocultas con 50 neuronas.
    4. Red LSTM de 50 neuronas.
    5. Red LSTM apilado.
    6. Red LSTM bidireccional.
    7. Modelo XGBoost.
    8. CNN LSTM.

9. Conv LSTM.

- Metaheurística para fijación de hiperparámetros

1. Parámetros por default.
2. Búsqueda Aleatoria.
3. Recocido Simulado.
4. Estrategias Evolutivas.

En el Apéndice A, se documenta el proceso seguido para el análisis de los resultados ajustados a la distribución Beta.

4.5.1. Distribución probabilística de los resultados

Con el fin de determinar si el tipo de algoritmo utilizado para los pronósticos y, si la aplicación de metaheurísticas dentro de estos, mejoran los resultados de los pronósticos, se analizarán los resultados de las cuatro métricas. Como era de suponerse (véase 4.28 y 4.29), los resultados de cada métrica no siguen una distribución Normal ni de forma multivariada ni univariada (véase Anexo 1), por lo que se descarta el Análisis de Varianza Multivariante (MANOVA) como la técnica a utilizar para determinar si los factores (algoritmo y metaheurística) influyen en los resultados. En el Anexo 1 también se muestran los resultados de las pruebas de bondad de ajuste y transformaciones realizadas cuando se intentó la aplicación del MANOVA y ANOVA. Aunado a lo anterior, también se observó que, aunque se encontrase una transformación adecuada para que los resultados de las métricas sigan una distribución Normal univariada cada una, seguía sin cumplirse el supuesto de Normalidad Multivariada y el supuesto de varianza constante. Por lo anterior, y con el fin de conocer la distribución de probabilidad que mejor describa los resultados de cada métrica, se realizaron más pruebas de Bondad de Ajuste. Ninguna de las distribuciones de la familia Exponencial se ajustaba de manera adecuada a los resultados de la métricas, por lo que se descartaron los Modelos de Lineales Generalizados. Después de un análisis más intensivo, se observó que al transformar los resultados en un rango de 0 - 1, las cuatro métricas seguían una distribución Beta. Por lo anterior, para analizar los resultados, se utilizó el modelo de Regresión Beta propuesto en [68], [69], [70].

Como se mencionó anteriormente, los resultados de los algoritmos de ML/AI por cada métrica se escalaron en el rango de $0-1$ para la realización de la prueba de Bondad de Ajuste de la distribución Beta. La expresión utilizada para dicho

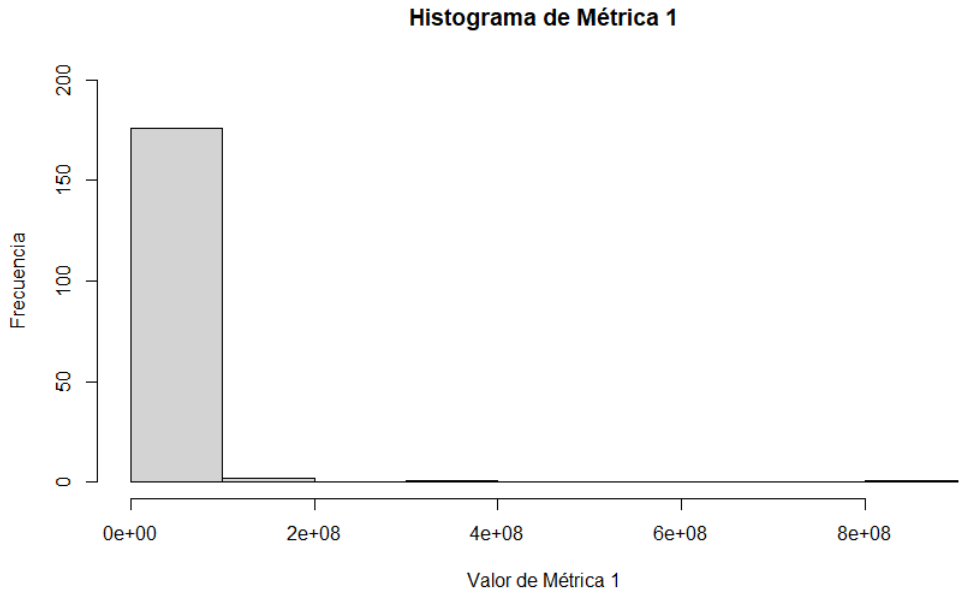


Figura 4.28: Histograma de frecuencias para la métrica 1.

escalamiento es la siguiente:

$$x_i = \frac{x_i - \text{mín}(X)}{\text{máx}(X) - \text{mín}(X)}$$

en donde x_i es el valor actual de la observación y X hace referencia al cálculo del operador sobre todas las observaciones, esta transformación se aplicó a todas las métricas por producto. La codificación ingresada al análisis de la Regresión Beta es el mismo usado al inicio de la sección 4.5, en donde se listan los algoritmos de predicción (1-9) y el tipo de optimización (1-4). Para determinar la significancia estadística de cada uno de los factores estudiados y analizados en la Regresión Beta, se fijó un nivel de α de 0.05.

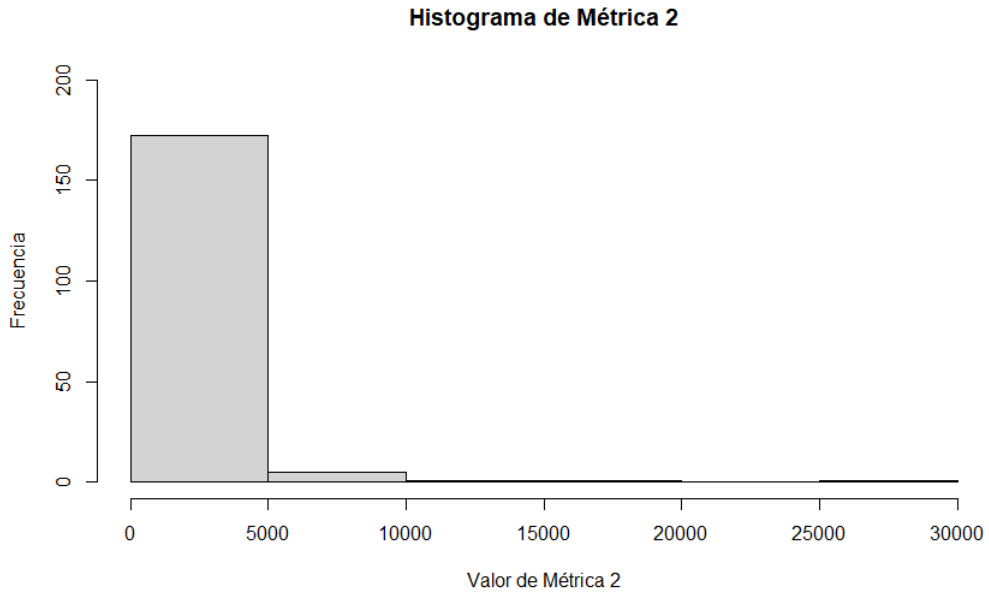


Figura 4.29: Histograma de frecuencias para la métrica 2.

Producto 1

En la tabla 4.32 se muestran los parámetros (α, β) de la función beta obtenidos mediante la maximización de la estimación de bondad de ajuste, aplicados al producto 1. De igual manera, se realizó la prueba Kolmogorov-Smirnov para validar el ajuste a la distribución beta.

Métrica	$\boldsymbol{\alpha}$	$\boldsymbol{\beta}$	**Valor P prueba K-S**
1	0.3709	46.4530	0.6805
2	0.6221	10.6468	0.8047
3	1.3710	0.3254	0.363
4	0.5971	9.7803	0.6337

Tabla 4.32: Resultados de la obtención de parámetros y prueba Kolmogorov-Smirnov para las métricas del producto 1.

Con los resultados obtenidos y gracias a que los valores p de la prueba Kolmogorov-Smirnov son mayores al nivel de significancia de 0.05, se puede confirmar que las métricas siguen una distribución Beta. Para dimensionar el efecto que cada uno de los factores (modelo de ML y algoritmo metaheurístico) tiene en las métricas de evaluación, se hizo uso de la Regresión Beta.

Se empleó la librería `betareg` del lenguaje de programación `R` para analizar la Regresión Beta a los datos transformados con la intención de comprender el efecto de cada uno de los factores en el producto 1. El resultado obtenido se encuentra en la tabla 4.33, se destaca que la referencia para el conjunto de modelos de ML fue el SVR codificado como ML 1, mientras que del lado de los algoritmos metaheurísticos, la referencia fue "sin metaheurística", codificado como Meta0.

	Métrica 1		**Métrica 2**		**Métrica 3**		**Métrica 4**	
	Coef.	Valor P	Coef.	Valor P	Coef.	Valor P	Coef.	Valor P
Intercepto	-1.4774	0.0237	-1.0979	0.05049	1.4843	0.00189	-1.1395	0.0432
ML2	1.3437	0.0872	1.7613	0.0086	-1.9	0.00047	1.78	0.00808
ML3	0.1707	0.8232	0.244	0.72192	-1.5839	0.00345	0.2711	0.69307
ML4	-0.4059	0.5914	-0.5654	0.42023	1.0578	0.09574	-0.5648	0.42175
ML5	-0.6115	0.4172	-1.0464	0.13907	1.257	0.01894	-1.0241	0.14828
ML6	-0.5057	0.503	-0.6805	0.33313	1.0267	0.10526	-0.6584	0.34985
ML7	0.3507	0.6474	0.5429	0.42413	-2.6554	2.27E-06	0.6673	0.32548
ML8	-0.4048	0.5924	-0.5602	0.4244	0.6922	0.26338	-0.5413	0.44108
ML9	-0.5378	0.476	-0.7657	0.27697	1.109	0.08169	-0.7471	0.28962
Meta1	-0.6462	0.2065	-0.8284	0.06978	-0.2006	0.60057	-0.797	0.08117
Meta2	-0.7581	0.1381	-1.0802	0.01927	0.6009	0.12333	-1.0562	0.02218
Meta3	-0.8575	0.0933	-1.199	0.00973	0.817	0.03766	-1.1342	0.01434

Tabla 4.33: Resultados de la regresión Beta para el producto 1.

Se observa que los valores p más significativos se encuentran en el intercepto, el algoritmo de ML 2 (Red Neuronal de una neurona y una capa oculta) junto con la metaheurística 3 (Estrategias Evolutivas). Debe de considerarse el signo del coeficiente, ya que indica el sentido de la significancia. En este caso, el algoritmo de ML 2 aporta al incremento de la métrica. Por otro lado, la metaheurística 3 contribuye significativamente al decrecimiento de la métrica. Al buscar reducirla, entonces se concluye que para el producto 1, la metaheurística 3 es la que más contribuye a la reducción del error cuadrático medio, que es la métrica en cuestión.

Similar a lo ocurrido con la métrica 1, el algoritmo ML 2 es el que más contribuye al incremento de la métrica 2 de error absoluto medio. De igual manera, los valores p para los tres algoritmos metaheurísticos confirman que son estadísticamente significativos de manera decreciente para la métrica en cuestión. Por lo tanto, se concluye que los tres contribuyen a la reducción de la métrica 2.

El caso de la métrica 3 llama la atención, en especial porque lo que está midiendo es la puntuación de la varianza explicada. En este sentido, se busca maximizar dicha métrica como criterio de decisión. Con lo obtenido del análisis de la Regresión Beta, se observa que la metaheurística 3 junto con los algoritmos de ML 9 (Conv-LSTM), 5 (Red LSTM Apilado) y 4 (Red LSTM de 50 neuronas), fueron los que contribuyeron, significativamente, al incremento de dicha métrica.

Recordando el objetivo de la métrica 4, ésta busca reducir el Error Medio Absoluto Porcentual. Por lo que las factores que contribuyen significativamente a la reducción de dicha métrica fueron los algoritmos metaheurísticos. Los tres, fueron estadísticamente significativos debido al valor p y al signo del coeficiente. Por lo tanto, se concluye que los algoritmos metaheurísticos usados tienen importancia para la reducción de la métrica 4.

Producto 2

En la tabla 4.34 se muestran los parámetros (α, β) de la función beta obtenidos mediante la maximización de la estimación de bondad de ajuste, aplicados al producto 2. De igual manera, se realizó la prueba Kolmogorov-Smirnov para validar el ajuste de la distribución beta.

Métrica	$\boldsymbol{\alpha}$	$\boldsymbol{\beta}$	**Valor P prueba K-S**
1	1.2070	23.7399	0.7328
2	2.3416	14.0810	0.8258
3	2.2347	0.9390	0.8223
4	2.5104	15.2298	0.7225

Tabla 4.34: Resultados de la obtención de parámetros y prueba Kolmogorov-Smirnov para las métricas del producto 2.

Con los resultados obtenidos y gracias a que los valores p de la prueba Kolmogorov-Smirnov son mayores al nivel de significancia de 0.05, se confirma que las métricas se ajustan a una distribución beta. Al igual que con el producto 1, se hizo el análisis de la regresión beta mediante la librería `betareg` en `R`. Los resultados obtenidos se documentan en la tabla 4.35.

El objetivo para la métrica 1 es minimizar su valor (error cuadrático medio), y de acuerdo a los resultados logrados, las metaheurísticas fueron significativas para la reducción de dicha métrica gracias a sus valores p. Por lo que se concluye que son estadísticamente significativas para la mejora de la métrica.

De acuerdo a la tabla referida, los tres algoritmos metaheurísticos tienen influencia de manera reductiva en la métrica 2 (error absoluto medio). Además, el algoritmo ML6 (Red LSTM Bidireccional) mostró ser el más efectivo en la reducción de la métrica de acuerdo a su valor p y el signo del coeficiente. Por lo que se concluye que para la evaluación de la métrica 2, los algoritmos metaheurísticos y el modelo ML8 (CNN-LSTM) son estadísticamente significativos.

El análisis de la anterior regresión beta, muestra que todos los algoritmos de ML son estadísticamente significativos en el incremento de la métrica 3 (puntua-

	Métrica 1		Métrica 2		Métrica 3		Métrica 4	
	Coef	Valor P	Coef	Valor P	Coef	Valor P	Coef	Valor P
Intercepto	-0.6117	0.2913	-0.0362	0.94267	-1.403	0.00013	-0.0763	0.8799
ML2	0.7265	0.3026	0.6212	0.311	1.5691	0.00018	0.6651	0.2791
ML3	-0.4815	0.5031	-0.6904	0.28296	1.5908	0.00015	-0.7069	0.2736
ML4	-0.9605	0.1845	-1.0776	0.10046	3.1437	6.98E-11	-1.0307	0.1164
ML5	-0.8558	0.2365	-0.8328	0.19859	4.6085	5.27E-15	-0.767	0.2363
ML6	-0.9741	0.1784	-1.1879	0.07172	3.2654	2.82E-11	-1.1205	0.0892
ML7	0.0992	0.8891	0.1971	0.7499	1.4686	0.00044	0.1945	0.7538
ML8	-0.3696	0.6065	-0.3947	0.53312	2.4344	3.64E-08	-0.3771	0.5525
ML9	-0.5988	0.4058	-0.7011	0.27582	2.8092	1.10E-09	-0.6816	0.2905
Meta1	-1.0262	0.0333	-0.9363	0.02747	-0.3528	0.22417	-0.9023	0.0335
Meta2	-0.9939	0.039	-1.0266	0.01626	0.621	0.04151	-1.0191	0.0172
Meta3	-1.1029	0.0225	-1.3797	0.00163	0.149	0.61389	-1.4039	0.0014

Tabla 4.35: Resultados de la regresión beta para el producto 2.

ción de la explicación de la varianza). Debe recordarse que el objetivo de mejora para esta métrica es incrementarla, por lo que todos los algoritmos de ML fueron estadísticamente significativos para la mejora de la métrica. Además, fue especialmente significativa la metaheurística 2 (Recocido Simulado) para ayudar a la mejora de dicha métrica.

El objetivo de mejora para la métrica 4 es hacia la reducción, por lo que del análisis anterior se puede recuperar que las tres metaheurísticas son estadísticamente significativas para tal efecto. Además el algoritmo de ML 6 (Red LSTM bidireccional) también es estadísticamente significativo para la mejora de la métrica.

Producto 3

En la tabla 4.36 se muestran los parámetros (α, β) de la función beta obtenidos mediante la maximización de la estimación de bondad de ajuste, aplicados al producto 3. De la misma manera, se realizó la prueba Kolmogorov-Smirnov para validar el ajuste de la distribución beta. Al tener todas las métricas valores p superiores al nivel de significancia de 0.05, se confirma que los datos se ajustan a la distribución beta.

El análisis de la regresión beta para el producto 3, se muestra en la tabla 4.37. De acuerdo a los resultados obtenidos de dicho análisis de regresión, se observa significancia estadística para las tres metaheurísticas, contribuyendo de manera importante a la reducción de la métrica 1. Adicional a esto, el algoritmo de ML 6 (Red LSTM bidireccional), también contribuye significativamente a la reducción y mejora de la métrica.

Métrica	α	β	**Valor P prueba K-S**
1	2.7023	6.5389	0.9926
2	3.8524	6.2262	0.9937
3	4.1838	3.1284	0.8408
4	3.9335	7.1011	0.9956

Tabla 4.36: Resultados de la obtención de parámetros y prueba Kolmogorov-Smirnov para las métricas del producto 3.

	Métrica 1		Métrica 2		Métrica 3		Métrica 4	
	Coef	Valor P	Coef	Valor P	Coef	Valor P	Coef	Valor P
Intercepto	0.4981	0.25361	0.875	0.03953	0.8526	0.0733	0.7225	0.07552
ML2	0.7126	0.1801	0.4984	0.33221	-2.9136	3.57E-06	0.7171	0.14677
ML3	-0.3666	0.50286	-0.6158	0.24012	-2.8371	5.51E-06	-0.5647	0.26441
ML4	-0.6626	0.23447	-0.6503	0.21565	-0.3402	0.5576	-0.7484	0.14394
ML5	-0.2878	0.5974	-0.2819	0.58526	-0.4615	0.4237	-0.3538	0.47932
ML6	-1.3697	0.01974	-1.5318	0.00631	-0.5512	0.3375	-1.5336	0.00515
ML7	0.0496	0.92643	-0.0816	0.8738	-1.0126	0.0749	0.1552	0.75255
ML8	0.0144	0.97859	-0.0054	0.99158	-0.9293	0.1024	-0.0334	0.94603
ML9	-0.2764	0.61179	-0.2452	0.63465	-0.8387	0.1411	-0.2172	0.66223
Meta1	-1.3712	0.00018	-1.2606	0.00035	0.5929	0.1198	-1.3124	0.00012
Meta2	-1.5682	2.54E-05	-1.4272	6.26E-05	0.5885	0.1226	-1.4115	3.94E-05
Meta3	-1.3727	0.00018	-1.3485	0.00014	0.2486	0.5097	-1.3003	0.00013

Tabla 4.37: Resultados de la regresión beta para el producto 3.

Al igual que con la métrica anterior, las tres metaheurísticas son estadísticamente significativas para la reducción del valor de la métrica 2. Lo mismo sucede con el algoritmo de ML6 (Red LSTM bidireccional) para la mejora y reducción de la métrica, recordando que dicha métrica busca medir y reducir el error absoluto medio.

La métrica 3 mide y evalúa qué tan bien el modelo en cuestión cuantifica la varianza, por lo que el objetivo es incrementarla. De acuerdo a los resultados obtenidos, los algoritmos de ML 2 y 3 (Red Neuronal de una neurona y una capa oculta; Red Neuronal de dos capas ocultas con 50 neuronas) son estadísticamente significativos para el decremento de dicha métrica. Además, el resto de algoritmos de ML y metaheurísticos, no mostraron relevancia estadística para la mejora de la métrica.

Del análisis anterior, se observa que las tres metaheurísticas son estadísticamente significativas para la reducción de la métrica 4. Y, como en las primeras dos métricas, el algoritmo de ML 6 (Red LSTM bidireccional) también es estadísticamente significativo para la reducción de dicha métrica.

Producto 4

En la tabla 4.38 se muestran los parámetros (α, β) de la función beta obtenidos mediante la maximización de la estimación de bondad de ajuste, aplicados al producto 4. Del mismo modo, se realizó la prueba Kolmogorov-Smirnov para validar el ajuste de la distribución beta. Al tener todas las métricas valores p superiores al nivel de significancia de 0.05, se confirma que los datos se ajustan a la distribución beta.

Métrica	$\boldsymbol{\alpha}$	$\boldsymbol{\beta}$	**Valor P prueba K-S**
1	0.4818	1.9084	0.8755
2	0.8154	1.7762	0.4973
3	1.1896	0.4122	0.7239
4	0.6765	1.7545	0.7836

Tabla 4.38: Resultados de la obtención de parámetros y prueba Kolmogorov-Smirnov para las métricas del producto 4.

El análisis de la regresión beta para el producto 4, se muestra en la tabla 4.39. De acuerdo a lo anterior, y tomando en cuenta el sentido de minimización de la métrica 1, se observa la importancia estadística de la metaheurística 3 (Estrategias Evolutivas) y de los algoritmos de ML 4, 5, 6, 8 y 9 (Red LSTM de 50 neuronas, Red LSTM apilado, Red LSTM Bidireccional, CNN-LSTM y

	Métrica 1		Métrica 2		Métrica 3		Métrica 4	
	Coef	Valor P	Coef	Valor P	Coef	Valor P	Coef	Valor P
Intercepto	-0.8155	0.03157	-0.379	0.24435	0.161	0.627	-0.8617	0.03139
ML2	0.4706	0.29269	0.659	0.08398	-0.3234	0.40849	0.7716	0.09407
ML3	0.6545	0.13967	0.7209	0.0588	-6201	0.11021	0.972	0.03559
ML4	-1.0743	0.04147	-0.8283	0.04982	1.4969	0.0017	-0.9507	0.06479
ML5	-2.4462	5.18E-05	-2.6558	1.66E-06	3.1722	5.69E-08	-2.3574	8.64E-05
ML6	-1.3296	0.01459	-1.5145	0.00124	1.8301	0.00026	-1.8045	0.00157
ML7	2.9985	1.36E-09	2.7122	5.21E-09	-2.9362	3.18E-10	2.3158	7.51E-06
ML8	-0.8488	0.0971	-0.6967	0.09331	1.1932	0.0088	-0.9379	0.06806
ML9	-0.969	6.22E-02	-0.8124	0.05388	1.428	0.00248	-1.1235	0.03264
Meta1	-0.2927	0.37625	-0.1403	0.62903	0.9921	0.00164	0.3801	0.2832
Meta2	-0.3971	2.34E-01	-0.2076	0.47606	0.3559	0.2363	0.0425	0.90527
Meta3	-1.3375	0.00022	-0.7716	0.01079	1.4063	1.76E-05	0.2025	0.56898

Tabla 4.39: Resultados de la regresión beta para el producto 4.

Conv-LSTM; respectivamente). Es importante hacer notar que el algoritmo de ML 7 (XGBoost), tiene importancia estadística pero en el sentido de incremento para la métrica en cuestión.

Los resultados obtenidos para la métrica 2 son similares a los encontrados para la métrica anterior. Los algoritmos de ML más significativos para el objetivo de reducción de la métrica fueron ML 4, 5, 6, 8 y 9. Además, la metaheurística 3 (Estrategias Evolutivas) también es estadísticamente significativa para la reducción de la métrica.

En el análisis de la métrica 3, el cual se busca maximizar, se encontró que los algoritmos de ML 4, 5, 6, 8 y 9 (Red LSTM de 50 neuronas, red LSTM apilado, red LSTM bidireccional, CNN LSTM y Conv LSTM) son significativos estadísticamente para el incremento de dicha métrica. Además, las metaheurísticas 1 y 3 (Búsqueda Aleatoria y Estrategias Evolutivas) también lo son en el mismo sentido de maximización de la métrica.

Para la métrica 4 que representa el error porcentual medio absoluto, se obtuvo significancia estadística para los algoritmos de ML 4, 5, 6, 8 y 9, al igual que para las anteriores métricas. Lo destacable es la ausencia de efecto estadístico para las metaheurísticas en la minimización de la métrica actual.

Producto 5

En la tabla 4.40 se muestran los parámetros (α, β) de la función beta obtenidos mediante la maximización de la estimación de bondad de ajuste, aplicados al producto 5. De igual manera, se realizó la prueba Kolmogorov-Smirnov para validar el ajuste de la distribución beta. Al tener todas las métricas valores p

superiores al nivel de significancia de 0.05, se confirma que los datos se ajustan a la distribución beta.

Métrica	$\boldsymbol{\alpha}$	$\boldsymbol{\beta}$	**Valor P prueba K-S**
1	0.6905	20.8889	0.9616
2	1.5764	10.3245	0.8965
3	0.9170	0.3054	0.5027
4	1.6043	10.7995	0.9273

Tabla 4.40: Resultados de la obtención de parámetros y prueba Kolmogorov-Smirnov para las métricas del producto 5.

	Métrica 1		Métrica 2		Métrica 3		Métrica 4	
	Coef	Valor P	Coef	Valor P	Coef	Valor P	Coef	Valor P
Intercepto	-1.6659	0.00525	-0.9173	0.05067	-0.1615	0.74025	-0.9256	0.05046
ML2	0.0268	0.96986	-0.0468	0.93523	-1.7779	0.00334	-0.0896	0.87748
ML3	-0.122	0.86405	-0.2074	0.72163	-1.0769	0.06553	-0.2323	0.69229
ML4	-0.7291	0.31031	-0.884	0.15006	2.0958	0.00156	-0.8893	0.15006
ML5	-1.0798	0.13386	-1.8232	0.00505	2.2603	0.0007	-1.8288	0.00509
ML6	-0.8281	0.24962	-1.0685	0.08621	2.0245	0.00218	-1.0554	0.0915
ML7	1.9177	0.00613	1.8212	0.00113	-1.4348	0.01561	1.7855	0.00152
ML8	-0.187	0.76322	-0.1839	0.75168	1.4337	0.02507	-0.2043	0.72734
ML9	-0.4321	0.54625	-0.5006	0.40071	1.6588	0.01054	-0.5224	0.38426
Meta1	0.238	0.61208	0.0914	0.81368	1.1715	0.00696	0.1122	0.77431
Meta2	-0.7226	0.12896	-0.7761	0.05742	1.0845	0.01228	-0.7577	0.06558
Meta3	-0.6458	0.17432	-0.6775	0.09498	1.0987	0.01121	-0.6512	0.11107

Tabla 4.41: Resultados de la regresión beta para el producto 5.

El análisis de la regresión beta para el producto 5, se muestra en la tabla 4.41. En el análisis de la regresión beta para la métrica 1, se observa que únicamente el algoritmo de ML 7 (XGBoost) es estadísticamente significativo para la reducción de dicha métrica. No se observa reelavancia estadística por parte de ninguna de las metaheurísticas.

De acuerdo con la información anterior, se aprecia que los algoritmos de ML 5 y 6 (Red LSTM apilado y Red LSTM Bidireccional), además de las metaheurísticas 2 y 3 (Recocido Simulado y Estrategias Evolutivas), son estadísticamente significativos para la reducción de la métrica 2 (error absoluto medio).

En el caso de la métrica 3, se busca maximizar dicha métrica y, de acuerdo al análisis de la regresión beta, los algoritmos de ML 4, 5, 6, 8, y 9 (Red LSTM de 50 neuronas, red LSTM apilado, Red LSTM bidireccional, CNN-LSTM y Conv-LSTM) son los que contribuyen significativamente a dicho objetivo. Además, las

tres metaheurísticas empleadas son igualmente significativas para maximizar la métrica 3.

Para la reducción de la métrica 4, los algoritmos de ML 5 y 6 (Red LSTM apilado y Red LSTM bidireccional), son los que más contribuyen a dicho objetivo. Además, se destaca el la metaheurística 2 (Recocido Simulado) para la minimización de la métrica 4.

Capítulo 5

Conclusiones y trabajos futuros

En este trabajo se presentó una propuesta para la solución del problema de pronósticos para series temporales cortas, aplicado a un problema industrial real. En especial, se hizo uso de diversas herramientas tanto de análisis de series temporales como el ARIMA, así como propuestas novedosas del campo del aprendizaje de máquina. La necesidad de tener pronósticos asertivos para la organización en cuestión, se ampara en la reducción de costos de almacén, operativos y logísticos. Además, dichos pronósticos permitirán a la empresa realizar planes de producción adecuados y una mejor planeación de la capacidad.

Con la poca cantidad disponible de información, se evaluaron diversas técnicas como el ARIMA y modelos de aprendizaje de máquina. Estos resultados permitieron establecer un punto de partida para dar respuesta a la hipótesis planteada. Una vez realizado el análisis exploratorio de los datos, se observó que cada uno de los cinco productos de análisis presentó un comportamiento singular. Dicho esto, no fue posible encontrar un modelo único que se ajustara a los cinco productos.

Partiendo de la idea anterior, y con una aplicación de modelos de ML con un ajuste empírico de hiperparámetros, se continuó con la experimentación empleando metaheurísticas para el ajuste de hiperparámetros, con la intención de hallar mejoras en la precisión de los pronósticos. Recapitulando, los algoritmos y metaheurísticas empleadas fueron:

- Algoritmo
    1. Soporte de Regresión Vectorial.
    2. Red neuronal de una neurona y una capa oculta.
    3. Red neuronal de dos capas ocultas con 50 neuronas.
    4. Red LSTM de 50 neuronas.
    5. Red LSTM apilado.
    6. Red LSTM bidireccional.

7. Modelo XGBoost.
8. CNN LSTM.
9. Conv LSTM.

- Forma de fijación de hiperparámetros

1. Parámetros por default.
2. Búsqueda Aleatoria.
3. Recocido Simulado.
4. Estrategias Evolutivas.

Para comparar cada uno de los algoritmos descritos así como el efecto de cada una de las metaheurísticas en dichos algoritmos, se recopilaron los resultados de las métricas de desempeño: MSE (error cuadrático medio), MAE (error absoluto medio), EVS (puntuación de la varianza explicada) y MAPE (error porcentual medio absoluto). Con esta información recolectada en cada observación, se realizó una validación de la distribución de los datos. Inicialmente, se buscó realizar un análisis de varianza multivariado para evaluar la contribución de cada uno de los factores y sus interacciones a la varianza de un modelo multivariado. Sin embargo, las observaciones son pocas para la cantidad de grados de libertad y, además, no fue posible validar los supuestos para el análisis multivariado.

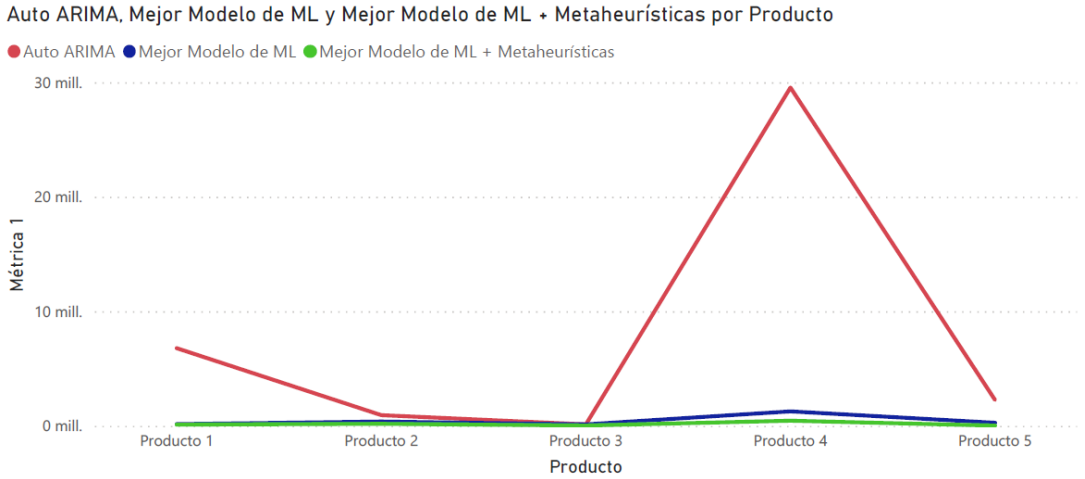
Es entonces que se analizó la distribución que siguen cada una de las métricas, determinando así, un ajuste a la distribución beta. Se determinaron entonces los parámetros distributivos mediante la estimación de bondad de ajuste para cada una de las métricas. Adicionalmente, se realizaron pruebas Kologorov-Smirnov para confirmar el ajuste de los datos a la distribución beta. Mediante un análisis de regresión beta, se encontraron los factores qué resultaron estadísticamente significativos para el incremento o decremento de la variable respuesta en cada uno de los productos. Se validó el efecto significativo de las metaheurísticas para cada uno de los productos, por lo que la hipótesis se confirma. Los resultados de la mejor combinación en cada uno de los productos se muestra en la tabla 5.1.

Producto	**ML**	**Metaheurística**	**Métrica 1**	**Métrica 2**	**Métrica 3**	**Métrica 4**
Producto 1	LSTM apilado	Recocido Simulado	112763.63	307.40	0.99	0.8495 %
Producto 2	LSTM bidireccional	Estrategias Evolutivas	183338.00	235.40	0.69	4.8724 %
Producto 3	LSTM bidireccional	Estrategias Evolutivas	32684.84	151.68	0.40	14.5786 %
Producto 4	LSTM apilado	Estrategias Evolutivas	456353.20	534.09	0.99	5.3517 %
Producto 5	LSTM apilado	Búsqueda Aleatoria	25319.64	145.09	1.00	1.0156 %

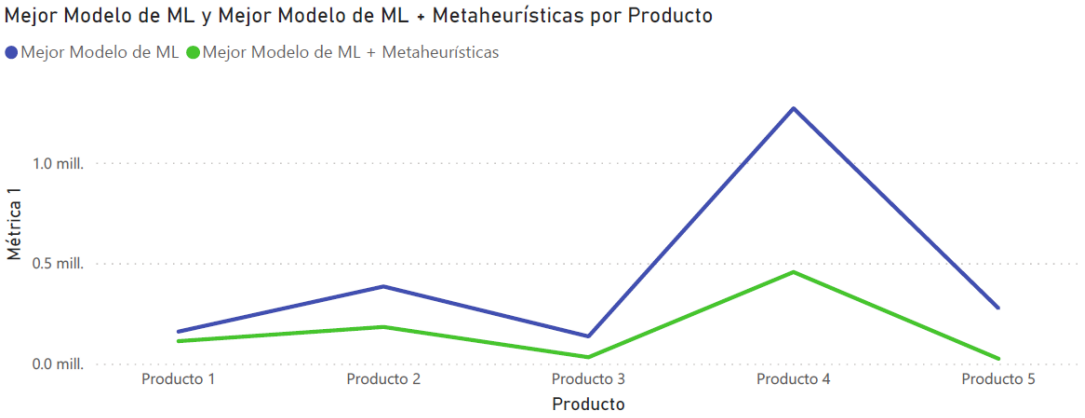
Tabla 5.1: Mejor combinación de algoritmos de ML y Metaheurísticos para cada producto.

El producto 3 es el único que no estuvo dentro del objetivo de precisión, sin embargo se quedó muy cerca del mismo. Esto se debe al comportamiento de los datos de entrenamiento, ya que para el primer año no se tiene una tendencia, sino hasta el segundo año.

Por último, la gráfica 5.1 muestra el comparativo del comportamiento de la función AutoARIMA, el mejor resultado de los modelos de ML y el mejor resultado de los modelos de ML mejorados por metaheurísticas en términos de la métrica 1.



(a) AutoARIMA vs Mejor modelo ML vs Mejor modelo de ML + Metaheurísticas



(b) Mejor modelo ML vs Mejor modelo de ML + Metaheurísticas

Figura 5.1: Comparación entre las funciones y modelos de predicción para la métrica 1.

En la figura 5.1(a) se observa que la función AutoARIMA tuvo valores altos para casi todos los productos. Además de un valor en la métrica 1 cercano a

los 30 millones para el producto 4. Por tal razón, se removió en la figura 5.1(b) con la intención de visualizar, en una mejor escala, la diferencia entre el mejor modelo de ML y la mejor combinación de ML y metaheurísticas. La diferencia entre ambas técnicas de pronóstico es considerable, tomando en consideración las unidades de la métrica. El mejor desempeño lo tuvo el modelo de ML mejorado por metaheurísticas para todos los productos.

En el caso de la métrica 2, la figura 5.2 muestra la comparación entre las tres categorías de técnicas predictivas mencionadas. Al igual que con la métrica anterior, la mejor combinación de modelos de ML mejorados por metaheurísticas, obtuvo el mejor desempeño al estar por debajo de las otras dos técnicas.

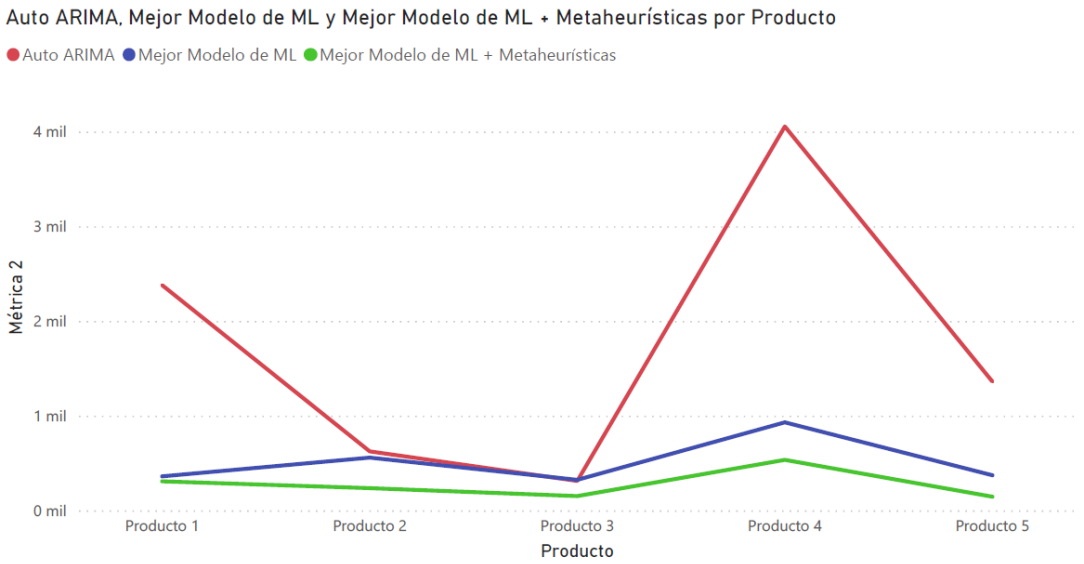
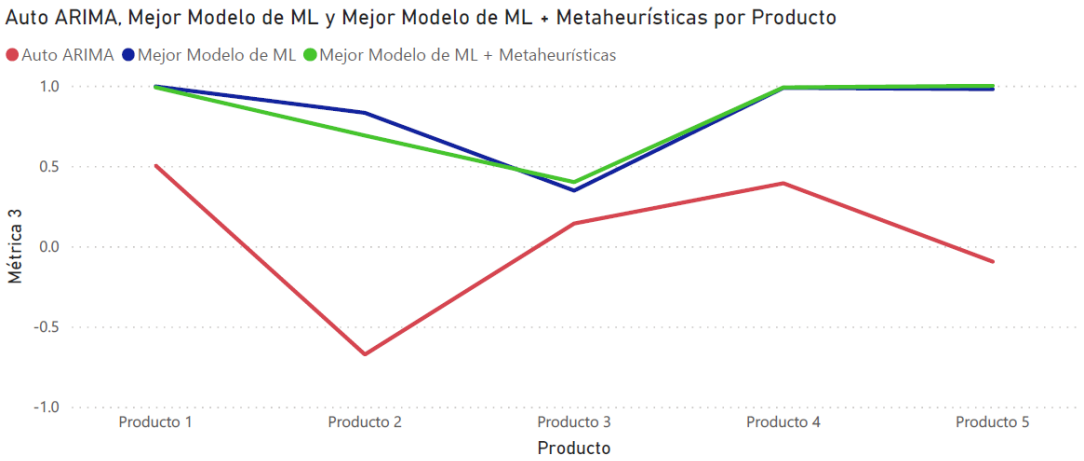


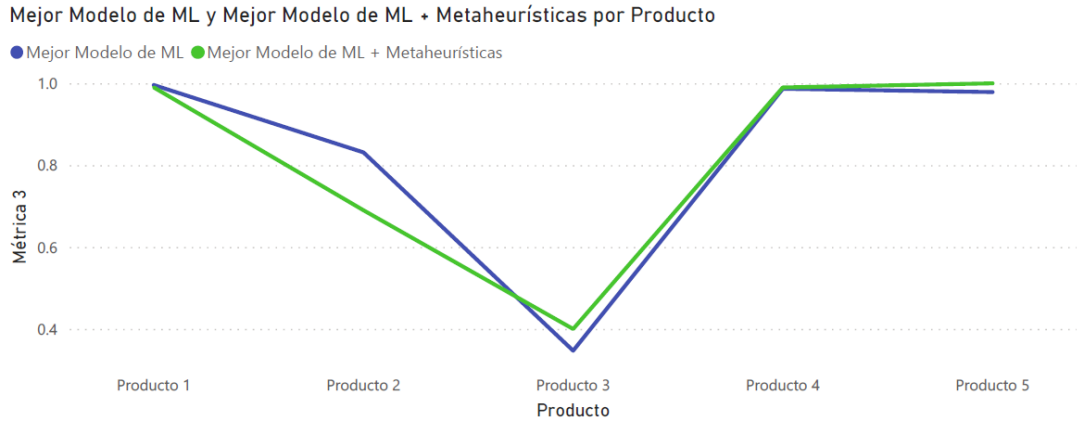
Figura 5.2: Comparación entre las funciones y modelos de predicción para la métrica 2.

Para la comparación de la métrica 3, la figura 5.3 muestra gráficamente los resultados de la comparación realizada. Debe tomarse en cuenta que el objetivo de evaluación de la métrica 3 es maximizar. Se observa en la figura 5.3(a) que la función AutoARIMA estuvo por debajo de los otros dos modelos para todos los productos. Para observar mejor la diferencia entre el mejor modelo de ML y el mejor modelo de ML mejorado por metaheurísticas, la figura 5.3(b) remueve la función AutoARIMA y permite visualizar que ambos modelos están muy cercanos entre sí.

Para terminar, el análisis comparativo para la métrica 4 se muestra en la figura 5.4. El objetivo de mejora para esta métrica es minimizar, por lo que la función AutoARIMA no tiene un buen comportamiento para los productos 1, 4



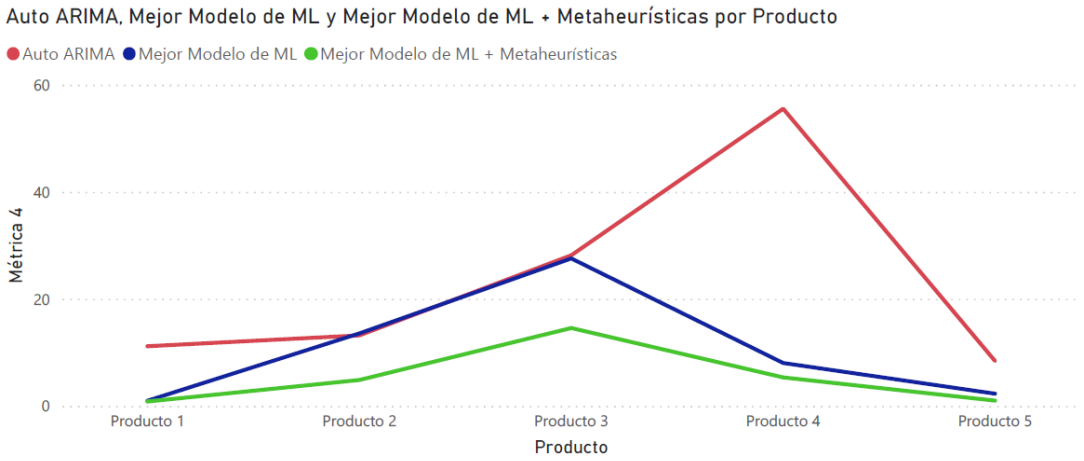
(a) AutoARIMA vs Mejor modelo ML vs Mejor modelo de ML + Metaheurísticas



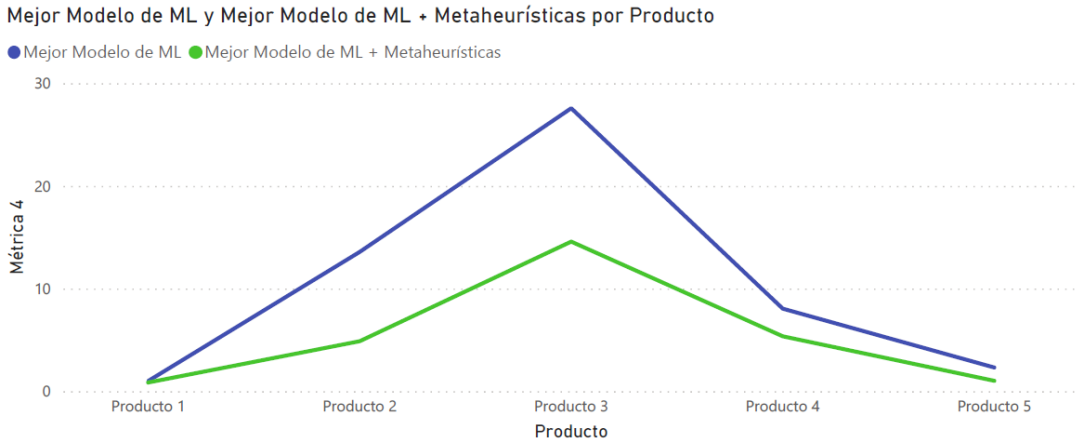
(b) Mejor modelo ML vs Mejor modelo de ML + Metaheurísticas

Figura 5.3: Comparación entre las funciones y modelos de predicción para la métrica 3.

y 5, de acuerdo a la figura 5.4(a). En la figura 5.4(b) se elimina dicha función dejando solo el mejor modelo de ML y el mejor modelo de ML mejorado por metaheurísticas. Aquí, de nueva cuenta, la combinación de ML y metaheurísticas obtuvo los mejores resultados.



(a) AutoARIMA vs Mejor modelo ML vs Mejor modelo de ML + Metaheurísticas



(b) Mejor modelo ML vs Mejor modelo de ML + Metaheurísticas

Figura 5.4: Comparación entre las funciones y modelos de predicción para la métrica 3.

5.1. Trabajo futuro

Si bien se cumplieron todos los objetivos planteados al inicio de la investigación y se validó la hipótesis, aún resta desplegar los modelos seleccionados en un entorno productivo. De igual manera es importante realizar un análisis de costos antes y después de la implementación de las soluciones, con la intención de validar la eficacia de la investigación así como determinar posibles mejoras para la organización.

Además, explorar alternativas para realizar el entrenamiento de los modelos al

estudiar series temporales cortas mediante ML. Esto con la finalidad de encontrar mejoras en los pronósticos a realizar y prevenir el sobreajuste en las predicciones obtenidas.

Por último, investigar el efecto de la metodología empleada en esta tesina aplicada a series temporales cortas multivariadas. Determinar si existe alguna mejora en los pronósticos a obtener y cómo realizar el entrenamiento de los modelos.

Bibliografía

[1] I. T. Christou, "Forecasting," in *Quantitative Methods in Supply Chain Management*. London: Springer London, 2012, pp. 139–202. [Online]. Available: http://link.springer.com/10.1007/978-0-85729-766-2_2

[2] S. M. A. Burney and S. M. Ali, "Sales Forecasting for Supply Chain Demand Management - A Novel Fuzzy Time Series Approach," in *2019 13th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS)*. IEEE, dec 2019, pp. 1–4. [Online]. Available: https://ieeexplore.ieee.org/document/9024810/

[3] D. A. A. Gnana Singh, E. J. Leavline, S. Muthukrishnan, and R. Yuvaraj, "Machine Learning based Business Forecasting," *International Journal of Information Engineering and Electronic Business*, vol. 10, no. 6, pp. 40–51, nov 2018. [Online]. Available: http://www.mecs-press.org/ijieeb/ijieeb-v10-n6/v10n6-5.html

[4] M. Florance, M.M. & Sawicz, "Positioning sales forecasting for better results," *The Journal of Business Forecasting Methods & Systems*, vol. 12, no. 4, p. 27, 1994. [Online]. Available: http://0-search.proquest.com.millenium.itesm.mx/docview/226923005?accountid=11643

[5] M. Gilliland, L. Tashman, and U. Sglavo, Eds., *Business Forecasting: Practical Problems and Solutions*. Hoboken, NJ, USA: John Wiley & Sons, Inc., dec 2015. [Online]. Available: http://doi.wiley.com/10.1002/9781119244592

[6] G. Sankaran, F. Sasso, R. Kepczynski, and A. Chiaraviglio, *Improving Forecasts with Integrated Business Planning*, ser. Management for Professionals. Cham, Switzerland: Springer International Publishing, 2019. [Online]. Available: http://link.springer.com/10.1007/978-3-030-05381-9

[7] D. Bunn and G. Wright, "Interaction of Judgemental and Statistical Forecasting Methods: Issues and Analysis," *Management Science*, vol. 37, no. 5, pp. 501–518, may 1991. [Online]. Available: http://pubsonline.informs.org/doi/abs/10.1287/mnsc.37.5.501

[8] J. T. Mentzer and J. E. Cox, "Familiarity, application, and performance of sales forecasting techniques," *Journal of Forecasting*, vol. 3, no. 1, pp. 27–36, jan 1984. [Online]. Available: http://doi.wiley.com/10.1002/for.3980030104

[9] S.-I. Ao, *Applied Time Series Analysis and Innovative Computing*, ser. Lecture Notes in Electrical Engineering. Dordrecht: Springer Netherlands, 2010, vol. 59. [Online]. Available: http://link.springer.com/10.1007/978-90-481-8768-3

[10] I. da Silva, M. Moura, I. Lins, E. Lopez Droguett, and E. Braga, "Non-Stationary Demand Forecasting Based on Empirical Mode Decomposition and Support Vector Machines," *IEEE Latin America Transactions*, vol. 15, no. 9, pp. 1785–1792, 2017. [Online]. Available: https://ieeexplore.ieee.org/document/8015086/

[11] M. Gurnani, Y. Korke, P. Shah, S. Udmale, V. Sambhe, and S. Bhirud, "Forecasting of sales by using fusion of machine learning techniques," in *2017 International Conference on Data Management, Analytics and Innovation (ICDMAI)*. IEEE, feb 2017, pp. 93–101. [Online]. Available: https://ieeexplore.ieee.org/document/8073492/

[12] D. Chen, "Chinese automobile demand prediction based on ARIMA model," in *2011 4th International Conference on Biomedical Engineering and Informatics (BMEI)*, vol. 4, no. 1. IEEE, oct 2011, pp. 2197–2201. [Online]. Available: http://ieeexplore.ieee.org/document/6098744/

[13] R. G. Hiranya Pemathilake, S. P. Karunathilake, J. L. Achira Jeewaka Shamal, and G. U. Ganegoda, "Sales Forecasting Based on AutoRegressive Integrated Moving Average and Recurrent Neural Network Hybrid Model," in *2018 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*. IEEE, jul 2018, pp. 27–33. [Online]. Available: https://ieeexplore.ieee.org/document/8686936/

[14] Xiangsheng Xie, Jiajun Ding, and Gang Hu, "Forecasting the retail sales of China's catering industry using support vector machines," in *2008 7th World Congress on Intelligent Control and Automation*, no. November 2005. IEEE, 2008, pp. 4458–4462. [Online]. Available: http://ieeexplore.ieee.org/document/4593641/

[15] C. C. Aggarwal, *Neural Networks and Deep Learning*. Cham: Springer International Publishing, 2018. [Online]. Available: http://link.springer.com/10.1007/978-3-319-94463-0

[16] F. Chen and T. Ou, "Sales forecasting system based on Gray extreme learning machine with Taguchi method in retail industry," *Expert Systems with Applications*, vol. 38, no. 3, pp. 1336–1345, mar 2011. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0957417410006317

[17] T. Chen and C. Guestrin, "XGBoost," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 42, no. 8. New York, NY, USA: ACM, aug 2016, pp. 785–794. [Online]. Available: https://dl.acm.org/doi/10.1145/2939672.2939785

[18] Y. Wang and Y. Guo, "Forecasting method of stock market volatility in time series data based on mixed model of ARIMA and XGBoost," *China Communications*, vol. 17, no. 3, pp. 205–221, mar 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9058617/

[19] B. M. Pavlyshenko, "Linear, machine learning and probabilistic approaches for time series analysis," in *2016 IEEE First International Conference on Data Stream Mining & Processing (DSMP)*, no. August. IEEE, aug 2016, pp. 377–381. [Online]. Available: http://ieeexplore.ieee.org/document/7583582/

[20] S. Sarafrazi, R. Choudhari, H. Mehta, C. Mehta, O. Japalaghi, K. Mehta, H. Han, and P. Francis-Lyon, "Cracking the "Sepsis" Code: Assessing Time Series Nature of EHR Data, and Using Deep Learning for Early Sepsis Prediction," in *2019 Computing in Cardiology Conference (CinC)*, vol. 45. Creative Commons, dec 2019, pp. 3–6. [Online]. Available: http://www.cinc.org/archives/2019/pdf/CinC2019-411.pdf

[21] Y. Zhou and S. Zheng, "Climate adaptive optimal design of an aerogel glazing system with the integration of a heuristic teaching-learning-based algorithm in machine learning-based optimization," *Renewable Energy*, vol. 153, pp. 375–391, jun 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0960148120301580

[22] Z.-F. Liu, L.-L. Li, M.-L. Tseng, and M. K. Lim, "Prediction short-term photovoltaic power using improved chicken swarm optimizer - Extreme learning machine model," *Journal of Cleaner Production*, vol. 248, p. 119272, mar 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0959652619341423

[23] V. Ayumi, L. M. R. Rere, M. I. Fanany, and A. M. Arymurthy, "Optimization of convolutional neural network using microcanonical annealing algorithm," in *2016 International Conference on Advanced Computer Science and*

Information Systems (ICACSIS). IEEE, oct 2016, pp. 506–511. [Online]. Available: http://ieeexplore.ieee.org/document/7872787/

[24] Y. Jiang, G. Tong, H. Yin, and N. Xiong, "A Pedestrian Detection Method Based on Genetic Algorithm for Optimize XGBoost Training Parameters," *IEEE Access*, vol. 7, pp. 118 310–118 321, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8807116/

[25] J. L. R. García and J. A. S. Gómez, "Descomposición de series de tiempo cortas: un enfoque bayesiano (1)," *Estadística española*, vol. 52, no. 173, pp. 127–154, 2010. [Online]. Available: http://www.eco.uva.es/tono/Enlaces_archivos/967_354_173_5.pdf

[26] A. Kirshners and A. Borisov, "A Comparative Analysis of Short Time Series Processing Methods," *Information Technology and Management Science*, vol. 15, no. 1, jan 2012. [Online]. Available: http://content.sciendo.com/view/journals/itms/15/1/article-p65.xml

[27] J. R. Bence, "Analysis of Short Time Series: Correcting for Autocorrelation," *Ecology*, vol. 76, no. 2, pp. 628–639, mar 1995. [Online]. Available: http://doi.wiley.com/10.2307/1941218

[28] A. Alonso, A. García Sipols, and C. Simón de Blas, "Forecasting financial short time series," *Electronic Journal of Applied Statistical Analysis*, vol. 11, no. 1, pp. 42–57, 2018.

[29] O. Hryniewicz and K. Kaczmarek, *Strengthening Links Between Data Analysis and Soft Computing*, ser. Advances in Intelligent Systems and Computing, P. Grzegorzewski, M. Gagolewski, O. Hryniewicz, and M. Á. Gil, Eds. Cham: Springer International Publishing, 2015, vol. 315, no. January. [Online]. Available: https://link.springer.com/10.1007/978-3-319-10765-3

[30] B. Brijs, *Business Analysis for Business Intelligence*. Auerbach Publications, apr 2016. [Online]. Available: https://www.taylorfrancis.com/books/9781439858370

[31] P. Pittman and J. Atwater, *APICS Dictionary*, 16th ed. APICS: American Production and Inventory Control Society, 2019. [Online]. Available: http://www.apics.org/apics-for-individuals/publications-and-research/apics-dictionary

[32] S. Feng, "Research on Demand Control of Business Intelligence," in *2019 International Conference on Machine Learning, Big Data and Business*

Intelligence (MLBDBI). IEEE, nov 2019, pp. 292–295. [Online]. Available: https://ieeexplore.ieee.org/document/8945799/

[33] C. W. Chase, *Demand-Driven Forecasting: A structured approach to Forecasting*, 2nd ed. New Jersey: John Wiley & Sons, Inc., 2013.

[34] N. Stefanovic, "Collaborative predictive business intelligence model for spare parts inventory replenishment," *Computer Science and Information Systems*, vol. 12, no. 3, pp. 911–930, 2015. [Online]. Available: http://www.doiserbia.nb.rs/Article.aspx?ID=1820-021415000034S

[35] W. Benton and H. Shin, "Manufacturing planning and control: The evolution of MRP and JIT integration," *European Journal of Operational Research*, vol. 110, no. 3, pp. 411–440, nov 1998. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0377221798000800

[36] A. Krishnamurthy, R. Suri, and M. Vernon, "Re-Examining the Performance of MRP and Kanban Material Control Strategies for Multi-Product Flexible Manufacturing Systems," *International Journal of Flexible Manufacturing Systems*, vol. 16, no. 2, pp. 123–150, apr 2004. [Online]. Available: http://link.springer.com/10.1023/B:FLEX.0000044837.86194.19

[37] R. H. Shumway and D. S. Stoffer, *Time Series Analysis and Its Applications*, 4th ed., ser. Springer Texts in Statistics. Cham: Springer International Publishing, 2017. [Online]. Available: http://link.springer.com/10.1007/978-3-319-52452-8

[38] M. Swamynathan, *Mastering Machine Learning with Python in Six Steps*. Berkeley, CA: Apress, 2017. [Online]. Available: http://link.springer.com/10.1007/978-1-4842-2866-1

[39] C. Narendra Babu and B. Eswara Reddy, "Predictive data mining on average global temperature using variants of arima models," in *IEEE-International Conference On Advances In Engineering, Science And Management (ICAESM -2012)*, 2012, pp. 256–260.

[40] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. J. Terpenning, "Stl: A seasonal-trend decomposition procedure based on loess," *Journal of Official Statistics*, vol. 6, pp. 3–33, 1990. [Online]. Available: https://www.scb.se/contentassets/ca21efb41fee47d293bbee5bf7be7fb3/stl-a-seasonal-trend-decomposition-procedure-based-on-loess.pdf

[41] Y. Kodratoff, *Introduction to machine learning*. Elsevier, 2014.

[42] E. Bisong, *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. Berkeley, CA: Apress, 2019. [Online]. Available: http://link.springer.com/10.1007/978-1-4842-4470-8

[43] T. A. Runkler, *Data Analytics*. Wiesbaden: Springer Fachmedien Wiesbaden, 2016, vol. 42, no. 6. [Online]. Available: http://link.springer.com/10.1007/978-3-658-14075-5

[44] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*, ser. Springer Texts in Statistics. New York, NY: Springer New York, 2013, vol. 103. [Online]. Available: http://link.springer.com/10.1007/978-1-4614-7138-7

[45] F.-C. Yuan, "Parameters Optimization Using Genetic Algorithms in Support Vector Regression for Sales Volume Forecasting," *Applied Mathematics*, vol. 03, no. 10, pp. 1480–1486, 2012. [Online]. Available: http://www.scirp.org/journal/doi.aspx?DOI=10.4236/am.2012.330207

[46] V. Vapnik, S. Golowich, A. Smola, M. Mozer, M. Jordan, and T. Petsche, "Support vector method for function approximation, regression, estimation, and signal processing," *Advances in Neural Information Processing System*, vol. 9, pp. 281–287, 1997.

[47] H. Chen, "Machine learning for information retrieval: Neural networks, symbolic learning, and genetic algorithms," *Journal of the American Society for Information Science*, vol. 46, no. 3, pp. 194–216, apr 1995. [Online]. Available: https://doi.org/10.1002/(SICI)1097-4571(199504)46:3%3C194::AID-ASI4%3E3.0.CO;2-S

[48] I. N. da Silva, D. Hernane Spatti, R. Andrade Flauzino, L. H. B. Liboni, and S. F. dos Reis Alves, *Artificial Neural Networks*, 1st ed. Switzerland: Springer International Publishing, 2017. [Online]. Available: http://link.springer.com/10.1007/978-3-319-43162-8

[49] Y.-C. Hu and J.-F. Tsai, "Evaluating classification performances of single-layer perceptron with a Choquet fuzzy integral-based neuron," *Expert Systems with Applications*, vol. 36, no. 2, pp. 1793–1800, mar 2009. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0957417407006483

[50] M. A. Istiake Sunny, M. M. S. Maswood, and A. G. Alharbi, "Deep Learning-Based Stock Price Prediction Using LSTM and Bi-Directional LSTM Model," in *2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES)*. IEEE, oct 2020, pp. 87–92. [Online]. Available: https://ieeexplore.ieee.org/document/9257950/

[51] D. Sarkar, R. Bali, and T. Sharma, *Practical Machine Learning with Python*. Berkeley, CA: Apress, 2018. [Online]. Available: http://link.springer.com/10.1007/978-1-4842-3207-1

[52] X. Liao, N. Cao, M. Li, and X. Kang, "Research on Short-Term Load Forecasting Using XGBoost Based on Similar Days," in *2019 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*. IEEE, jan 2019, pp. 675–678. [Online]. Available: https://ieeexplore.ieee.org/document/8669635/

[53] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting," jun 2015. [Online]. Available: http://arxiv.org/abs/1506.04214

[54] "ConvLSTM2D layer." [Online]. Available: https://keras.io/api/layers/recurrent_layers/conv_lstm2d/

[55] T.-Y. Kim and S.-B. Cho, "Predicting residential energy consumption using CNN-LSTM neural networks," *Energy*, vol. 182, pp. 72–81, sep 2019. [Online]. Available: https://doi.org/10.1016/j.energy.2019.05.230https://linkinghub.elsevier.com/retrieve/pii/S0360544219311223

[56] I. E. Livieris, E. Pintelas, and P. Pintelas, "A CNN–LSTM model for gold price time-series forecasting," *Neural Computing and Applications*, vol. 32, no. 23, pp. 17 351–17 360, dec 2020. [Online]. Available: https://link.springer.com/10.1007/s00521-020-04867-x

[57] "Conv1D layer." [Online]. Available: https://keras.io/api/layers/convolution_layers/convolution1d/

[58] "LSTM layer." [Online]. Available: https://keras.io/api/layers/recurrent_layers/lstm/

[59] A. Duarte, M. Laguna, and R. Marti, *Metaheuristics for Business Analytics*, ser. EURO Advanced Tutorials on Operational Research. Cham: Springer International Publishing, 2018. [Online]. Available: http://link.springer.com/10.1007/978-3-319-68119-1

[60] F. Herrera, "Introducción a los Algoritmos Metaheurísticos," in *XIII Conferencia de la Asociación Española para la Inteligencia Artificial*. Universidad de Granada, 2009, pp. 2–7. [Online]. Available: https://sci2s.ugr.es/sites/default/files/files/Teaching/OtherPostGraduateCourses/Metaheuristicas/Int-Metaheuristicas-CAEPIA-2009.pdf

[61] R. Hernández Torres, M. Irizar Mesa, O. Llanes Santiago, L. D. T. Câmara, A. J. da Silva Neto, and L. M. Zumalacárregui de Cárdenas, "Comparación de diferentes algoritmos metaheurísticos en la estimación de parámetros del modelo relacional general de cromatografía líquida en columna," *Ingeniare. Revista chilena de ingeniería*, vol. 22, no. 1, pp. 14–25, jan 2014. [Online]. Available: https://scielo.conicyt.cl/scielo.php?script=sci_arttext&pid=S0718-33052014000100003

[62] B. Chopard and M. Tomassini, *An Introduction to Metaheuristics for Optimization*, ser. Natural Computing Series. Cham: Springer International Publishing, 2018. [Online]. Available: http://link.springer.com/10.1007/978-3-319-93073-2

[63] M. Gendreau and J.-Y. Potvin, "Metaheuristics in Combinatorial Optimization," *Annals of Operations Research*, vol. 140, no. 1, pp. 189–213, nov 2005. [Online]. Available: http://link.springer.com/10.1007/s10479-005-3971-7

[64] C. Blum and G. R. Raidl, *Hybrid Metaheuristics*, ser. Artificial Intelligence: Foundations, Theory, and Algorithms. Cham: Springer International Publishing, 2016. [Online]. Available: http://link.springer.com/10.1007/978-3-319-30883-8

[65] R. Kumar, P. Kumar, and Y. Kumar, "Time Series Data Prediction using IoT and Machine Learning Technique," *Procedia Computer Science*, vol. 167, pp. 373–381, 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1877050920307067

[66] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[67] T. G. Smith *et al.*, "pmdarima: Arima estimators for Python," 2017–, [Online; accessed December 20, 2020]. [Online]. Available: http://www.alkaline-ml.com/pmdarima

[68] S. Ferrari and F. Cribari-Neto, "Beta regression for modelling rates and proportions," *Journal of Applied Statistics*, vol. 31, no. 7, pp. 799–815, 2004. [Online]. Available: https://doi.org/10.1080/0266476042000214501

[69] M. Smithson and J. Verkuilen, "A better lemon squeezer? maximum-likelihood regression with beta-distributed dependent variables." *Psychological methods*, vol. 11, no. 1, p. 54, 2006.

[70] A. B. Simas, W. Barreto-Souza, and A. V. Rocha, "Improved estimators for a general class of beta regression models," *Computational Statistics & Data Analysis*, vol. 54, no. 2, pp. 348–366, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167947309003107

Apéndice A

Análisis de varianza multivariado y univariado

Similar a lo ocurrido con el análisis de la varianza univariado (ANOVA), el MANOVA requiere variables de respuesta continuas y predictores categóricos. Al hacer uso del MANOVA en lugar de un ANOVA individual por cada una de las métricas, se reduce la probabilidad de rechazar erróneamente la hipótesis nula. Adicionalmente, el MANOVA puede ayudar a identificar pequeñas diferencias que no serían posibles de detectar mediante el ANOVA individual.

A.1. Validación de supuestos MANOVA

En el MANOVA se contrasta la hipótesis nula (H_0) de que las medias de todos los grupos son iguales o provienen de la misma población $H_0 : \mu_1 = \mu_2 = \mu_3 = ... = \mu_n$. Para poder hacer uso del MANOVA, se requiere el cumplimiento de tres supuestos estadísticos:

1. Los datos deben proceder de muestras aleatorias simples.
2. Las variables dependientes deben de seguir una distribución Normal Multivariada.
3. Las varianzas de las subpoblaciones deben ser iguales.

En la tabla A.1 se muestran los resultados de las pruebas de bondad de ajuste para la prueba de normalidad multivariada. De dicha tabla se puede observar que las métricas no siguen una distribución Normal Multivariada ya que cada valor-p es muy pequeño. Por lo anterior, se buscaron transformaciones a cada una de estas para que el supuesto se cumpla. A continuación se muestran las transformaciones realizadas mediante la transformación de Johnson.

- Producto 1
  1. Métrica 1 (MSE): $w = -1.59959 + 0.339646 \arcsin\left(\dfrac{x - 168922}{32376.2}\right)$.
  2. Métrica 2 (MAE): $w = -4.83697 + 0.728670 \ln(x - 278.101)$.
  3. Métrica 3 (EVS): Se ajusta a la normalidad.
  4. Métrica 4 (MAPE): $w = -0.526679 + 0.645302 \ln(x - 0.778051)$.
- Producto 2
  1. Métrica 1 (MSE): $w = -1.25536 + 0.701413 \arcsin\left(\dfrac{x - 308400}{104385}\right)$.
  2. Métrica 2 (MAE): $w = -0.550765 + 0.747332 \arcsin\left(\dfrac{x - 534.984}{140.918}\right)$.
  3. Métrica 3 (EVS): $w = -0.908548 + 0.72695 \ln\left(\dfrac{x + 1.02117}{0.872701 - x}\right)$.
  4. Métrica 4 (MAPE):$w = -0.354028 + 0.629234 \arcsin\left(\dfrac{x - 13.8605}{2.84693}\right)$.
- Producto 3
  1. Métrica 1 (MSE): $w = -0.502007 + 1.38193 \arcsin\left(\dfrac{x - 100223}{60434.9}\right)$.
  2. Métrica 2 (MAE): Se ajusta a la normalidad.
  3. Métrica 3 (EVS): $w = 1.04827 + 1.45173 \arcsin\left(\dfrac{x - 0.473045}{0.352906}\right)$.
  4. Métrica 4 (MAPE): Se ajusta a la normalidad.
- Producto 4
  1. Métrica 1 (MSE): $-1.19212 + 0.596211 \ln\left(\dfrac{x + 209771}{129980957 - x}\right)$.
  2. Métrica 2 (MAE): $w = 0.870188 + 0.892084 \ln\left(\dfrac{x - 192.081}{9167.9 - x}\right)$.
  3. Métrica 3 (EVS): $w = -0.787594 + 0.508068 \ln\left(\dfrac{x + 0.441738}{0.997484 - x}\right)$.
  4. Métrica 4 (MAPE): $w = 1.05682 + 0.756023 \ln\left(\dfrac{x - 4.59460}{65.4407 - x}\right)$.
- Producto 5

1. Métrica 1 (MSE): $w = -11.8097 + 0.829097 \ln(x + 120198)$.
2. Métrica 2 (MAE): $w = -0.830756 + 1.02568 \arcsin\left(\frac{x - 710.631}{619.108}\right)$.
3. Métrica 3 (EVS): $w = -0.780080 + 0.364620 \ln\left(\frac{x + 1.65089}{0.997181 - x}\right)$.
4. Métrica 4 (MAPE): $w = -1.07383 + 1.18989 \arcsin\left(\frac{x - 3.64489}{4.67462}\right)$.

Conjunto de datos	Prueba	Estadístico de prueba	Valor p	Normalidad
Producto 1	E-statistic	9.077762	0.0	No
Producto 2	E-statistic	3.769667	0.0	No
Producto 3	E-statistic	3.53614	0.0	No
Producto 4	E-statistic	4.696569	0.0	No
Producto 5	E-statistic	3.567861	0.0	No

Tabla A.1: Resultados de la prueba de normalidad multivariada mediante la prueba Doornik-Hansen.

Tras realizar la transformación necesaria en cada una de las métricas, se realizó de nuevo una prueba de normalidad multivariada, los resultados se muestran en la tabla A.2. Con la información obtenida se concluye que los datos no se ajustan de manera multivariada a la normalidad, por lo que no es posible realizar el análisis MANOVA.

Conjunto de datos	Prueba	Estadístico de prueba	Valor p	Normalidad
Producto 1	E-statistic	2.542589	0.0	No
Producto 2	E-statistic	2.618254	0.0	No
Producto 3	E-statistic	3.109278	0.0	No
Producto 4	E-statistic	2.804982	0.0	No
Producto 5	E-statistic	1.732091	0.0	No

Tabla A.2: Resultados de la prueba de normalidad multivariada tras la transformación de las métricas, mediante la prueba Doornik-Hansen.

Las métricas de evaluación empleadas son diferentes entre sí y evalúan diferentes características de cada una de las interacciones entre modelos de ML y algoritmos metaheurísticos. Por lo tanto, se puede evaluar estadísticamente cada uno de ellos de manera independiente, con el objetivo de entender y determinar su comportamiento mediante el ANOVA de dos vías.

En este punto es importante resaltar que no es posible realizar el ANOVA de dos vías con interacción, ya que se tienen 36 observaciones por producto y no

serían suficientes los grados de libertad para integrar las interacciones de los factores. Por lo tanto se realizó un análisis ANOVA de dos vías sin interacción para cada uno de los productos. Para poder realizar un análisis ANOVA se requiere el cumplimiento de tres supuestos:

1. Las poblaciones son normales.
2. Las muestras son independientes.
3. Las poblaciones tienen igual varianza.

En la siguiente sección se detalla la validación de los supuestos del análisis ANOVA.

A.2. Validación de supuestos ANOVA

Para simplificar el análisis, se estudió la métrica 1 que es el error cuadrático medio como variable dependiente. Para cada uno de los productos se realizó la prueba de normalidad Anderson-Darling, en donde la hipótesis nula H_0 : es que la muestra proviene de una distribución normal, mientras que la hipótesis alternativa H_a : es que la muestra no proviene de una distribución normal. El criterio de decisión está basado en el nivel de significancia $\alpha = 0.05$ Si $p < \alpha$, entonces se rechaza H_0 y, en caso contrario, no es posible rechazar H_0.

La tabla A.3 muestra el resultado de la prueba de normalidad para cada uno de los productos.

Conjunto de datos	**Valor p**
Producto 1	0.2264
Producto 2	0.6342
Producto 3	0.9077
Producto 4	0.4790
Producto 5	0.2143

Tabla A.3: Resultados de la prueba de normalidad Anderson-Darling.

Con la información anterior se concluye que la métrica en cada uno de los productos, se ajusta a la normalidad. Por lo que el primer supuesto se valida.

Para validar el supuesto de igualdad de varianza, se hizo uso de la prueba de Levene. Para dicha prueba, la hipótesis nula H_0 es: las varianzas poblacionales son iguales, mientras que para la hipótesis nula H_a : establece que existe diferencia

entre las varianzas. El criterio de decisión es que si el valor p es menor al nivel de significancia ($\alpha = 0.05$), entonces se rechaza H_0.

La tabla A.4 muestra los resultados obtenidos tras evaluar la igualdad de varianzas para la métrica 1 en cada uno de los productos. Se divide en dos grupos: Modelo de ML y Algoritmo Metaheurístico. Se observa que para el producto 4 y 5 se tiene un valor p menor a 0.05 en el grupo del modelo de ML, por lo que no se cumple con el supuesto de igualdad de varianza.

Conjunto de datos	**Valor p (ML)**	**Valor p (Metaheurístico)**
Producto 1	0.5239	0.9957
Producto 2	0.5161	0.7146
Producto 3	0.8116	0.9388
Producto 4	0.01988	0.9949
Producto 5	6.45e-05	0.8016

Tabla A.4: Resultados de la prueba de igualdad de varianzas de Levene.