UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS

FACULTAD DE CONTADURÍA, ADMINISTRACIÓN E INFORMÁTICA

MAESTRÍA EN OPTIMIZACIÓN Y CÓMPUTO APLICADO

Estudio de técnicas de Machine Learning para estimar la producción de aguacate en la zona norte del Estado de Morelos

T E S I S

QUE PARA OBTENER EL GRADO DE

MAESTRÍA EN OPTIMIZACIÓN Y CÓMPUTO APLICADO

PRESENTA

PAUL PARIS ARIZMENDI PERALTA

DIRECTOR DE TESIS

DR. JOSÉ ALBERTO HERNÁNDEZ AGUILAR

CODIRECTOR

DR. PEDRO MORENO BERNAL

REVISORES:

DR. JOSÉ ALBERTO HERNÁNDEZ AGUILAR
DR. PEDRO MORENO BERNAL
DR. SERGIO NESMACHNOW
DR. OUTMANE OUBRAM
DRA. LORENA DÍAZ GONZÁLEZ

CUERNAVACA, MORELOS. MARZO, 2024

Para C.C

«Yo no sé por qué la nieve es blanca. Sin embargo, pienso que la nieve blanca es hermosa. No la odio...»

DICTAMEN COMITÉ REVISOR

Dra. Lorena Díaz González

Dr. Outmane Oubram

Dr. Sergio Nesmachnow

Dr. Pedro Moreno Bernal

Dr. José Alberto Hernández Aguilar

Agradecimientos

Permítanme contarles el comienzo:

«En el principio creó Dios los cielos y la tierra. Y la tierra estaba desordenada y vacía, y las tinieblas estaban sobre la faz del abismo, y el Espíritu de Dios se movía sobre la faz de las aguas. Y dijo Dios: Sea la luz; y fue la luz. Y vio Dios que la luz era buena; y separó Dios la luz de las tinieblas. Y llamó Dios a la luz Día, y a las tinieblas llamó Noche. Y fue la tarde y la mañana un día.»

Gavillas han pasado desde ese momento, y ahora estamos aquí de nuevo, en la segunda tesis que me permito elaborar, puedo aseguraros de que esta trilogía estará completa, estando yo tan maravillado de mi creación, así como Dios lo está de la suya.

Agradezco a mis padres, Salomón Paul y Carmen Peralta, por animarme e impulsarme en esta segunda etapa de mi educación profesional, si bien ellos han sido quienes me dieron el impulso y yo quién he volado, a ellos les debo más que mi vida.

Agradezco a mi hermano Alan Salomón por ser tan tosco como siempre, ambos sabemos que, bajo esa capa de seriedad suya, se esconde una persona noble y de un gran corazón.

Agradezco a los doctores José Alberto y Pedro Moreno por ser los guías en esta etapa de la maestría, tal vez no fui el alumno que esperaban, pero sí el que necesitaban

Y nuevamente a ti, si has seguido conmigo hasta el final.

CONTENIDO

CONTENIDO v
ÍNDICE DE FIGURAS vii
RESUMEN 1
ABSTRACT 2
CAPÍTULO 1 INTRODUCCIÓN 3
1.1 Marco contextual 3
1.2 Planteamiento del problema 6
1.3 Hipótesis 6
1.4 Pregunta de investigación 6
1.5 Objetivos generales y específicos 6
1.5.1 General 6
1.5.2 Específicos 7
1.6 Justificación 7
1.7 Alcances y limitaciones 8
1.8 Estructura del documento. 10
CAPÍTULO 2 ESTADO DEL ARTE 11
2.1 Machine Learning en la agricultura 11
2.2 Internet de las cosas (IoT) 20
2.3 Área de Estudio 22
CAPÍTULO 3 METODOLOGÍA DE INVESTIGACIÓN 24
3.1 Estación IOT 25
3.2 Componentes de software. 26
3.3 Pipeline /Esquema de trabajo para la realización de Machine Learning 29
3.4 Obtención de los datos 30
3.5 Datos de Origen 32
3.6 Importar los datos. 33
3.7 Preprocesamiento 33
3.8 Limpieza 33
3.9 Preparación de datos 35
3.10 Recorte 36
3.11 Resultados de la extracción de datos 36
3.12 Modelos/Técnicas usadas 39

3.12.1 Algoritmos de regresión 39
3.12.2 Perceptrón multicapa 40
3.12.3 Máquinas de soporte vectorial 40
3.12.4 Árboles de decisión 41
3.12.5 Bosques aleatorios 42
3.12.6 M5 Prime 42
Capítulo 4. Resultados y discusión 46
4.1 Gráfica de correlación 46
4.2 Métricas de evaluación de datos 47
- **4.2.1 R2 o Coeficiente de determinación** 47
- **4.2.2 RMSE (Root Mean Square Error)** 48

4.3 Datos de entrenamiento y datos de prueba 49
4.4 Resultados de los de los modelos de ML sin sintonizar 50
4.5 Grid Search 51
4.6 Resultados optimizados de los modelos de ML. 52
4 Conclusiones y trabajos futuros 67
BIBLIOGRAFÍA 69
Anexos 73
Código utilizado 73

ÍNDICE DE FIGURAS

Figura 1 Aplicación de Machine Learning en el campo. (miRiego, 2019) 5
Figura 2 Ubicación de la UPA de estudio (Alphabet Inc., 2023)..... 8
Figura 3 Vista previa de la interfaz de usuario del sistema ALMANACMEX..... 20
Figura 4 ubicación geográfica de la huerta de aguacates (Alphabet Inc., 2023) 23
Figura 5 Arquitectura IoT propuesta..... 27
Figura 6 Plataforma IoT propuesta..... 28
Figura 7 Estación IoT y sus componentes 29
Figura 8 Esquema de trabajo basado en (Nvidia, 2022) 30
Figura 13 Producción kilos de aguacate..... 38
Figura 14a Resultados al aplicar Decision Tree en los datos de entrenamiento (Train)..... 53
Figura 14b Resultados al aplicar Decision Tree en los datos de prueba (Test)..... 54
Figura 15a Resultados al aplicar Multilayer Regressor en los datos de entrenamiento (Train) 55
Figura 15b Resultados al aplicar Multilayer Regressor en los datos de prueba (Test) 56
Figura 16a Resultados al aplicar Support Vector Regressor en los datos de entrenamiento (Train) 57
Figura 16b Resultados al aplicar Support Vector Regressor en los datos de prueba (Test) 58
Figura 18 M5 Prime..... 62
Figura 19 Gráficos de precisión de los modelos Decision Tree Train y Test 63
Figura 20 Gráficos de precisión de los modelos Multilayer Regressor Train y Test..... 64
Figura 21 Gráficos de precisión de los modelos Support Vector Regressor Train y Test..... 64
Figura 22 Gráficos de precisión de los modelos Random Forest Train y Test..... 65

RESUMEN

En este estudio de investigación, se realizó una comparación entre cinco modelos de Machine Learning (ML) utilizando datos provenientes del sustrato y del entorno de una Unidad Productora de Aguacate (UPA) en la zona norte del Estado de Morelos, específicamente en el municipio de Huitzilac. Los datos recopilados incluyeron temperaturas, humedades, luminosidad, así como los niveles de Nitrógeno (N), Fósforo (P) y Potasio (K). Estos datos fueron monitoreados durante un período de un año mediante sensores especializados.

La metodología empleada para el procesamiento de datos consistió en preprocesarlos, limpiarlos, prepararlos, visualizarlos y, posteriormente, someterlos a pruebas utilizando cinco modelos seleccionados: Decisión Tree, Random Forest, Support Vector Machine, Multilayer Perceptron y M5 Prime. La evaluación inicial de estos modelos se realizó sin modificar sus hiperparámetros (parámetros base), seguida de una fase de ajuste utilizando una grid search (sintonización de parámetros).

Una vez obtenidos los resultados optimizados de los cinco modelos, se procedió a compararlos, utilizando dos métricas como guía: el coeficiente de determinación (R cuadrada, R2) y el error cuadrático medio (Root Mean Square Error, RMSE). El objetivo principal fue identificar el mejor modelo capaz de predecir la producción de aguacate en el futuro.

En resumen, este trabajo culminó con la identificación de un modelo óptimo que contribuye a la determinación precisa de la producción futura de aguacate, proporcionando una importante herramienta que ayuda en la toma de decisiones dentro de la Unidad Productora de Aguacate (UPA).

ABSTRACT

In this research study, a comparison was made among five Machine Learning (ML) models using data obtained from the substrate and environment of an Avocado Production Unit (UPA) located in the northern zone of the State of Morelos, specifically in the municipality of Huitzilac. The collected data included temperatures, humidities, luminosity, as well as Nitrogen (N), Phosphorus (P), and Potassium (K) levels. These data were monitored over a period of one year using specialized sensors.

The methodology employed for data processing involved preprocessing, cleaning, preparation, visualization, and subsequent testing using five selected models: Decision Tree, Random Forest, Support Vector Machine, Multilayer Perceptron, and M5 Prime. The initial evaluation of these models was conducted without altering their hyperparameters (base parameters), followed by an adjustment phase using grid search (parameter tuning).

Once the optimized results of the five models were obtained, a comparison was made using two metrics as guidance: the coefficient of determination (R squared, R2) and the Root Mean Square Error (RMSE). The primary objective was to identify the best model capable of predicting future avocado production.

In summary, this work concluded with the identification of an optimal model that contributes to the precise determination of future avocado production, providing a valuable tool for decision-making in the Avocado Production Unit (UPA).

CAPÍTULO 1 INTRODUCCIÓN

1.1 Marco contextual

La población mundial está experimentando un crecimiento acelerado y se proyecta que alcance los 9.700 millones de habitantes para el año 2050, en comparación con los 7.700 millones registrados en 2020 (Cohen, 2001), este incremento demográfico conlleva un aumento en la demanda de alimentos en los próximos años. Para satisfacer esta creciente demanda, se requerirá un incremento significativo en la productividad agrícola.

Sin embargo, el desafío radica en aumentar la productividad alimentaria sin comprometer los recursos naturales y el medio ambiente. La producción intensiva de cultivos conlleva problemas como la degradación del suelo debido al uso excesivo de productos químicos y pesticidas, la escasez de agua, la deforestación y la emisión de gases de efecto invernadero. Es evidente que se necesitan soluciones tecnológicas para lograr un equilibrio entre el aumento de la producción de alimentos y la preservación del medio ambiente.

En este contexto, la industria 4.0 está emergiendo como una fuerza transformadora en el mundo. Tecnologías como la realidad virtual y aumentada, el Internet de las cosas (IoT), los asistentes virtuales, el Big Data, la computación en la nube, el Machine Learning, las redes neuronales artificiales y muchas otras, están revolucionando diversos sectores.

En el ámbito agrícola, el uso de sensores, dispositivos portátiles, análisis de datos y robótica ofrece nuevas oportunidades para mejorar la producción de alimentos de manera más eficiente y sostenible. Estas tecnologías permiten desde la creación de prototipos y pruebas hasta la optimización de los recursos y la conectividad en la cadena de suministro.

Hoy en día, la vida cotidiana se apoya en la capacidad de las máquinas para aprender y llevar a cabo ciertas tareas de alto nivel. La inteligencia artificial y el aprendizaje automático (ML) se han convertido en componentes esenciales en una variedad de aplicaciones, incluyendo, pero no limitándose a, las recomendaciones de películas en plataformas digitales, el reconocimiento de voz de los asistentes virtuales y la capacidad de los vehículos autónomos para detectar y navegar por las carreteras. (Sánchez Romero, 2021). Estos avances en el campo de la inteligencia artificial tienen sus raíces en años de investigación y desarrollo.

En este contexto, la agricultura debe aportar soluciones prácticas para beneficiar la demanda de consumo de alimentos mediante la incorporación de tecnología basada en la Industria 4.0 de manera sostenible. La agricultura sostenible se refiere al ciclo de producción, cosecha y distribución de todo lo relacionado con la agricultura sin desperdicio (Ojeda-Beltrán, 2022).

En un principio, las máquinas se programaban para llevar a cabo las actividades o acciones necesarias. Sin embargo, en la actualidad, la inteligencia artificial posibilita que las máquinas aprendan sin necesidad de ser programadas específicamente para ello.

Hoy en día, la agricultura utiliza tecnología avanzada para aumentar la productividad mediante la optimización de recursos de forma controlada y medible. De esta forma, la agricultura inteligente combina la agricultura sostenible y las Tecnologías de la Información y la Comunicación. En particular, el Internet de las cosas (IoT) y la tecnología de sensores son vitales para aumentar la productividad en la agricultura inteligente.

El “Machine Learning” (ML) es una rama de la inteligencia artificial que posee sistemas que son capaces de identificar patrones entre los datos para hacer predicciones mediante el uso de algoritmos, los cuales simulan la inteligencia humana mediante el aprendizaje del ambiente que los rodea (El Naqa & Murphy, 2015). El ML surge como un subcampo de la inteligencia artificial que capacita a las computadoras para aprender sobre algo sin necesidad de ser programadas de forma explícita para ello.

Debido a su versatilidad, el ML tiene muchas aplicaciones en diversos campos de la investigación: en las cadenas de suministros la inteligencia artificial (IA) ha sido aplicada en diferentes etapas del suministro haciendo más eficiente este proceso (Icarte Ahumada, 2016), otro campo de aplicación es en el área de mantenimiento donde las técnicas de Machine Learning se emplean para determinar cuándo pueden ocurrir las fallas en las máquinas y equipos a través del análisis de los datos de los procesos y productos (Guerrero Cano, Luque Sendra, Lama Ruiz, & Córdoba Roldán, 2019) (Lázaro Enguita, 2018).

De igual manera, el ML se utiliza para la predicción de parámetros del suelo como el carbono orgánico y el contenido de humedad, la predicción del rendimiento de los cultivos, la detección de enfermedades y malezas en los cultivos y la detección de especies (El-Bendary, El Hariri, Hassanien, & Badr, 2015). También es una herramienta para afrontar los retos de la agricultura sostenible, que junto con el IOT (Internet de las cosas) habilita la maquinaria para la próxima revolución agrícola (agricultura digital) (Sharma, Jain, Gupta, & Chowdary, 2021).

Un ejemplo de lo que se puede hacer con el Machine Learning en la agricultura se puede apreciar en la Figura 1, en la cual se muestra un hombre que utilizando una tableta puede determinar algunos parámetros como los son humedad, pH, temperatura y nutrientes.

Figura 1 Aplicación de Machine Learning en el campo. (miRiego, 2019)

1.2 Planteamiento del problema

Uno de los principales problemas que se tienen en el campo mexicano y particularmente en el Estado de Morelos, es que la tecnología no ha sido aplicada de manera que ayude a mejorar su productividad, por ejemplo, las técnicas de trabajo para una Unidad Productiva de Aguacate (UPA) se hacen a través de métodos empíricos transmitidos de generación en generación y no consideran la información técnica y capacitación existente. Así mismo, no se le da la importancia requerida a la relación que existe entre las variables meteorológicas y nutricionales y la producción de aguacate. pero no se ha considerado todo el entorno ambiental que afecta la producción de aguacate.

En esta investigación se busca conocer de qué manera la aplicación de técnicas de ML puedan ayudar en el desarrollo de la producción de aguacate y de ser así, determinar cuáles son las técnicas más adecuadas para predecir la producción de aguacate en la zona norte del estado de Morelos, para su posterior replicación.

1.3 Hipótesis

H1. Aplicando técnicas de Machine Learning y considerando variables meteorológicas y nutrimentales se podrá pronosticar el rendimiento en la producción de aguacate de una UPA.

1.4 Pregunta de investigación

¿Cómo se puede aplicar ML para predecir la producción de aguacate de una huerta experimental de aguacate en el municipio de Huitzilac, Morelos, a partir de datos climáticos y datos nutricionales de la planta?

1.5 Objetivos generales y específicos

1.5.1 General

Predecir la producción de aguacate en la zona norte del estado de Morelos mediante técnicas de ML usando datos climáticos y propiedades de los nutrientes (abono).

1.5.2 Específicos

1. Determinar el rendimiento a través de la simulación de condiciones meteorológicas y nutricionales de la UPA.
2. Contrastar los resultados del modelado con datos de campo.
3. Generar una base de datos con información meteorológica y nutrimental de la huerta piloto utilizando una estación de monitoreo local.
4. Aplicar fundamentos de Machine Learning para desarrollar una metodología que permita predecir la producción de aguacate considerando factores meteorológicos y nutricionales del terreno (área de trabajo).
5. Aplicar cinco modelos de Machine Learning (Linear Regression, Multilayer Perceptron, Support Vector Machine, Random Forest y M5 Prime) para estimar la producción de aguacate con condiciones climáticas y datos nutrimentales de Unidad Productora de Aguacate (UPA).
6. Identificar la mejor técnica de Machine Learning que permita pronosticar la producción de una unidad productora de aguacate (UPA) en base a los datos climáticos y nutricionales.

1.6 Justificación

La razón detrás del desarrollo de este proyecto surge de la necesidad de mejorar la gestión de los recursos agrícolas disponibles, como nutrientes, plaguicidas y abonos, así como de la preocupación por no comprometer los recursos para las generaciones futuras. Esto implica la preservación de los elementos vitales del entorno, como el suelo, el agua y el aire, para garantizar la continuidad de la productividad de la tierra sin contaminación.

El ML se revela como una herramienta de gran utilidad para cumplir nuestros objetivos, ya que nos proporciona técnicas y modelos de gran potencia. Estos recursos nos permiten realizar análisis de manera más eficiente y efectiva, facilitando la determinación de resultados en nuestros estudios.

1.7 Alcances y limitaciones

El estudio se realizó en el periodo de tiempo de agosto del 2022 al 31 de marzo del 2023, en la temporada de lluvias correspondiente al verano del hemisferio norte, siendo el ambiente una zona de transición entre el ecosistema boscoso y la selva baja caducifolia.

Este trabajo de investigación está considerado para una huerta piloto ubicada en el municipio de Huitzilac, Morelos, con condiciones bien definidas de la edafología de los árboles. La Figura 2 muestra una vista satelital de la UPA de estudio.

Figura 2 Ubicación de la UPA de estudio (Alphabet Inc., 2023)

Como limitaciones se tienen los siguientes puntos:

- Conectividad de red de nivel medio.

Considerando la ubicación geográfica en una zona boscosa con una deficiente cobertura telefónica, se debe tener en cuenta que la conectividad disponible se encuentra restringida al estándar de tercera generación (3G) como máximo.

- Cortes intermitentes de energía.

No obstante, la disponibilidad de energía eléctrica para mantener en funcionamiento la estación de monitoreo, resulta relevante señalar que la distancia entre la Unidad de Procesamiento de Datos (UPA) y la red eléctrica principal ocasiona una conexión intermitente. Además, durante la temporada de lluvias, que abarca típicamente los meses de mayo a octubre, se producen interrupciones prolongadas del suministro eléctrico y variaciones en la tensión.

- Tiempo limitado de respaldo operativo.

Se dispone de un supresor de picos con sistema de alimentación ininterrumpida (UPS) para salvaguardar la estación autónoma ante fluctuaciones de voltaje y descargas eléctricas, además de proporcionar energía de respaldo en caso de interrupciones eléctricas. No obstante, es importante tener en cuenta que la duración de esta energía de respaldo es limitada. En caso de producirse un corte prolongado de suministro eléctrico, la estación se volverá inoperativa hasta que se restablezca el servicio eléctrico.

- Objetos de estudio

Dado que la mayoría de los árboles son de la misma variedad, Hass Flor de María, se seleccionaron únicamente dos ejemplares para llevar a cabo las pruebas. Uno de ellos se utilizó como grupo de control, en el cual se mantuvo un crecimiento natural sin la aplicación de ningún tipo de nutriente o fertilizante adicional. El otro árbol fue sometido a un régimen de nutrición de acuerdo con el calendario establecido por el huertero.

1.8 Estructura del documento.

En el capítulo 1 de este documento se presenta la introducción, así como el marco contextual de este trabajo.

El capítulo 2 nos plantea el estado del arte, el cómo el Machine Learning es aplicado en el área de estudio y en trabajos similares.

El capítulo 3 presenta el proceso del trabajo, desde las pruebas preliminares en la estación de monitoreo autónoma, los datos a recolectar, el proceso de limpieza y clasificación de datos, y el cómo fueron aplicados en los diversos modelos estudiados.

El capítulo 4 nos muestra los resultados y los trabajos futuros a realizar.

Posteriormente se encuentran, las conclusiones y trabajos futuros y finalmente, las referencias bibliográficas.

CAPÍTULO 2 ESTADO DEL ARTE

El concepto de Industria 4.0 se refiere a la cuarta revolución industrial, la cual, impulsada por la transformación digital, implica un cambio significativo en la manera en que las empresas gestionan su cadena de valor. Originariamente, la Industria 4.0 es una iniciativa estratégica presentada por el gobierno alemán en la Feria de Hannover de 2011, con el propósito de transformar la industria mediante la digitalización y la utilización del potencial ofrecido por las nuevas tecnologías. (Maisueche Cuadrado, 2019)

La Inteligencia Artificial se refiere a la capacidad de una computadora para exhibir habilidades similares a las humanas en términos de procesamiento de información, aprendizaje y toma de decisiones. Su propósito radica en abordar problemas complejos imitando el razonamiento humano a través de algoritmos. Para implementar estos algoritmos, se utiliza una herramienta llamada Aprendizaje Automático (Machine Learning), la cual, respaldada por técnicas estadísticas, permite que las máquinas aprendan a partir de la experiencia.

Los productores, los gobiernos y la academia están explorando formas innovadoras para aumentar la producción agrícola. Recientes trabajos relacionados proponen diferentes metodologías utilizando tecnologías IoT para procesos agrícolas eficientes (Diaz, Mazza, Combarro, Giménez, & Gaiad, 2017). A continuación, se presenta una breve reseña de trabajos relacionados a los campos de agricultura inteligente, Machine Learning y otros métodos de aprendizaje automático.

2.1 Machine Learning en la agricultura

Gómez et al. (Gomes Alves, y otros, 2019) propusieron un sistema para recopilar datos de una sonda de suelo utilizando una plataforma IoT para la gestión del agua agrícola. El dispositivo IoT recolectó datos y mostró información en un panel de

monitoreo. Se conectaron varios nodos de sensores a la plataforma IoT, incluido el sensor de temperatura y humedad DHT22, el sensor de luz ambiental BH1750, el dispositivo GPS Venus de posición geoespacial, el sensor de temperatura del suelo DS18B20 y el sensor de humedad del suelo CSMv1.2.

Un módulo Raspberry Pi-2 recibió señales de los nodos sensores a través de la interfaz I2C y el bus serie GPIO. Las aplicaciones en la nube utilizadas fueron Fiware IoT, Fiware Orion, MongoDB, Draco, MySQL y Grafana. Los resultados mostraron que la plataforma IoT analizó datos en la nube. Luego, los datos procesados fueron enviados como entrada a un sistema físico basado en un Controlador Lógico Programable para el sistema de riego, disminuyendo el impacto en la gestión del agua.

La agricultura desempeña un papel crucial en la economía global y enfrenta una creciente presión debido a la expansión continua de la población humana. En respuesta a este desafío, han surgido nuevos campos científicos como la agrotecnología y la agricultura de precisión, también conocida como agricultura digital. Estos campos emplean enfoques intensivos en datos para aumentar la productividad agrícola y reducir su impacto ambiental (Liakos, Busato, Moshou, Pearson, & Bochtis, 2018).

Los datos generados en las operaciones agrícolas modernas provienen de una amplia gama de sensores que permiten una comprensión más completa del entorno operativo. Esto incluye una interacción de las condiciones dinámicas del cultivo, el suelo y el clima, así como datos generados por la maquinaria utilizada en la operación. Esta abundancia de datos conduce a una mayor precisión en la toma de decisiones y a una capacidad mejorada para actuar rápidamente en respuesta a las condiciones cambiantes.

Serikul et al. (Serikul, Nakpong, & Nakjuatong, 2019) propusieron un prototipo de cápsula inteligente para monitorear la humedad en bolsas de arroz almacenadas en un almacén. El prototipo utilizó un microcontrolador ESP8266 y un sensor de humedad SHT21. Los datos recopilados se enviaron a un servidor Blynk a través de una red Wi-Fi y una aplicación móvil.

La inversión en el prototipo de hardware fue de 300 USD. Los resultados mostraron un monitoreo efectivo de la humedad en cada cápsula inteligente en tiempo real. La aplicación de cápsulas inteligentes fue adecuada para controlar la humedad en bolsas de arroz ayudando a prevenir la humedad excesiva.

En el artículo **ML de agricultura precisa**, los autores (Sharma, Jain, Gupta, & Chowdary, 2021) presentan una revisión sistemática de las aplicaciones de ML en el campo de la agricultura. Las áreas en las que se enfocan son la predicción de parámetros del suelo como el contenido de humedad y carbono orgánico, la predicción del rendimiento de los cultivos, el desarrollo de enfermedades, malezas y la detección de especies nocivas.

Para resolver los problemas de humedad en el suelo se utilizaron técnicas de modelo de regresión basado en máquina de aprendizaje extremo (ELM). El algoritmo se probó con 5 funciones diferentes y la predicción se validó utilizando la técnica de validación cruzada de dejar uno fuera.

Así mismo, se utilizaron varios otros algoritmos de ML tales como lo son Bosques aleatorios (Random Forest), kNN (vecinos más cercanos), RNN (Redes neuronales artificiales), obteniendo como resultado de un 99% en eficacia, comprobaron que el riego controlado con técnicas de ML en el cultivo de trigo es mejor que un riego periódico y manual/tradicional.

Trilles et al. (Trilles, González-Pérez, & Huerta, 2020) propusieron una plataforma IoT para monitorear la producción de vino en un contexto de agricultura inteligente. La plataforma IoT estudiada se evaluó en cuanto a escalabilidad, estabilidad, interoperabilidad y reutilización. Los paradigmas informáticos utilizados fueron microservicios y computación sin servidor.

Se validó la arquitectura tecnológica propuesta (SEnviro Connect) para la producción de vino en España. Los resultados experimentales mostraron que, en cuanto a la escalabilidad, la arquitectura de microservicios evaluada requiere solo un servidor para garantizar la escalabilidad y la estabilidad con 0% de pérdidas de tasas de rendimiento superiores a 2400 msg/seg.

El consumo de CPU y memoria fue menor y estable. La arquitectura propuesta basada en contenedores Docker ofrecía escalabilidad horizontal y reutilización. La validación de la plataforma se realizó en un periodo de 140 días durante la temporada de viñedo en 2018, monitoreando cinco nodos SEnviro.

Los autores (Diaz, Mazza, Combarro, Giménez, & Gaiad, 2017) en su investigación **Machine Learning aplicado a la predicción de la producción de cítricos** analizaron datos de cítricos (limón, mandarina y naranja) en la provincia de Corrientes, Argentina. Su objetivo era predecir la producción de dichos frutos mediante el algoritmo M5 Prime, un modelo constructor de árboles de regresión que produce una clasificación basada en funciones lineales por partes.

Los datos disponibles consisten en información sobre 946 huertos de cítricos pertenecientes a variedades tales como limón (*citrus limon burman*), mandarina (*citrus reticulata blanco*) y naranja (*citrus sinensis osbeck*). Se utilizaron datos tales como posición global (latitud, longitud, grados, minutos y segundos), temperaturas mínimas, promedio y máximas, precipitaciones anuales totales y días sin heladas, así como otros datos del área de cultivo, dígase distancia entre plantas, existencia de fauna local, etc.

La información se analizó con estadística descriptiva, coeficientes de correlación, análisis de componentes principales y Biplot. La producción fue estimada utilizando el algoritmo M5-Prime, que es un constructor de árboles de regresión que genera una clasificación basada en funciones lineales por partes. Para todas las especies analizadas, la variable más relevante resultó ser la edad de los árboles. En las plantaciones de mandarina y naranja, la edad fue seguida por las distancias entre y dentro de las hileras, y además el riego también tuvo un impacto en la producción de mandarina.

Como resultados se obtuvo que la producción por árbol está fuertemente asociada de una manera positiva a los árboles de una juventud promedio y relacionada negativamente a las precipitaciones pluviales en el caso del limón, en caso de la mandarina la relación viene asociada a la distancia entre plantas tanto en filas como en hileras, finalmente, en el caso de la naranja, la relación va asociada

positivamente con la edad y de manera negativa con las precipitaciones, algo similar al limón.

(García Cañón, 2019) participó en la implementación de técnicas de ML para predecir variables meteorológicas y del suelo que afectan la agricultura. El objetivo era establecer modelos de ML capaces de predecir variables del suelo, como la humedad, los nutrientes y las temperaturas, para que los agricultores pudieran tener un horizonte de predicción y tomar decisiones adecuadas para sus cultivos.

Utilizó la metodología de bosques aleatorios, conocida como Random Forest (RF). En un árbol individual de este modelo, el nodo raíz es el punto de decisión que divide el conjunto de datos utilizando la variable que proporciona la mejor métrica de división para cada subconjunto.

Además, mediante análisis correspondientes, se pudo determinar la importancia de otras variables. Por ejemplo, si se conoce de antemano que la radiación solar tiene un fuerte impacto en la temperatura, se puede ejercer un control aún mayor sobre los cultivos. Esto se logra mediante el análisis de la importancia de las características (Feature importance), que cuantifica la relevancia de cada variable en la predicción.

Como resultado, se ha demostrado que algoritmos robustos como Random Forest Regression pueden predecir con un alto nivel de precisión variables en horizontes de tiempo amplios, siempre que se disponga de los datos necesarios para entrenar los modelos.

El ML resultó útil en otros aspectos, no solamente para la predicción de frutos, sino también para revisar aspectos tales como las enfermedades de la fruta.

La antracnosis representa una de las enfermedades más significativas que afectan al fruto del aguacate, y su presencia puede limitar considerablemente las posibilidades de exportación de este producto. En el trabajo de (Morales García, Rodríguez Guzmán, Azpíroz Rivero, & Pedraza Santos, 2009) **Modelo para la estimación del área del fruto en la evaluación de la antracnosis en aguacate**, se desarrolló un método no destructivo e indirecto para estimar el área del fruto de

aguacate Hass y evaluar la severidad de la antracnosis. Este método consistió en medir el largo y ancho de 175 frutos, a los cuales se les extrajo la pulpa y la semilla, y se trazó el contorno de la cáscara sobre papel. Posteriormente, se recortó este contorno y se midió con un integrador de lámina foliar. Se estableció una relación entre el área de la cáscara, el largo y el ancho del fruto mediante una regresión lineal múltiple, considerando el largo y el ancho de los frutos como variables independientes y el área como la variable dependiente a predecir.

Al comparar los valores observados con los estimados, se encontró una estrecha coincidencia entre el área real del fruto medida con el integrador y el área del fruto estimada con el modelo. La validación del modelo mediante un análisis de regresión arrojó un buen ajuste, lo que permitió su aplicación en la evaluación de la severidad de la antracnosis en aguacates Hass de Michoacán, México. (Morales García, Rodríguez Guzmán, Azpiroz Rivero, & Pedraza Santos, 2009)

Las manchas ocasionadas por el patógeno fueron medidas y el área enferma estimada se comparó con el área total del fruto calculada utilizando el modelo propuesto. Se encontró que la evaluación de la severidad por antracnosis fue más precisa en comparación con el uso de una escala arbitraria basada en el porcentaje de daño del área total del fruto. Estos hallazgos respaldan la validez del modelo para calcular el área del fruto de aguacate.

Machine Learning aplicado a estimar la producción en el sector agrícola.

Los investigadores (Nosratabadi, Ardabili, Lakner, Mako, & Mosavi, 2021) buscando encontrar una predicción tanto de la producción agrícola como ganadera de la República Islámica de Irán trabajaron con una **Predicción de la producción de alimentos utilizando algoritmos de Machine Learning, perceptrón multicapa y ANFIS,** para poder determinar la cantidad de suministros con los que contará el país en los próximos 10 años.

En su trabajo se utilizan técnicas de ML y redes neuronales artificiales (ANN) para poder llevar a cabo entrenamientos de modelos con los datos históricos de una variedad de animales y plantas/granos producidos a nivel nacional. En el caso del

trigo, desarrollaron un algoritmo híbrido GWO-ANN (Grey Wolf Optimizer Artificial Neural Network) el cual consiste en trabajar de primera mano con una red de neuronas que analice los datos de entrada para pasar a un análisis individual de los resultados dependiendo de las entradas y concentrarlos en una salida final.

Los resultados obtenidos muestran un claro aumento en la producción tanto agrícola como ganadera del país, con el detalle de que los datos utilizados son los datos finales proporcionados por instancias locales, esto significa que son datos que ya consideran pérdidas y/o mermas posibles.

En el trabajo **Uso de técnicas de aprendizaje automático para evaluar la madurez del tomate** desarrollado por (El-Bendary, El Hariri, Hassanien, & Badr, 2015) se usa ML para determinar la madurez del fruto del jitomate en cultivos para lograr obtener la mayor calidad del fruto cosechado.

Se presentan acercamientos de clasificación automática en cuanto medidas y evaluación del fruto obtenido y sus diferentes etapas de maduración. Para ello, los autores utilizaron algoritmos de Componentes de Análisis Principales (PCA) en adición de Maquinas de Soporte Vectorial (SVM) y Análisis Linear Discriminatorio (LDA).

Los datos usados incluyeron un total de 250 imágenes para entrenamiento y prueba de los conjuntos de datos con una validación cruzada de 10 veces. Los resultados obtenidos indican una precisión del 90.80% usando algoritmos de SVM en su modalidad uno-contra-uno, una precisión del 84.80% utilizando algoritmos de SVM en su modalidad de uno-contra-todos con una función de kernel lineal y, finalmente una clasificación de madurez del 84% usando el algoritmo LDA.

La Asignación Latente de Dirichlet (ALD), también conocida como Latent Dirichlet Allocation (LDA), es un modelo generativo que permite explicar conjuntos de observaciones a través de grupos no observados (Blei, Ng, & Jordan, 2003) Estos grupos explican las similitudes presentes en ciertas partes de los datos. Por ejemplo, si las observaciones son palabras en documentos, el modelo supone que cada documento es una combinación de un número limitado de categorías

(llamadas también tópicos), y la presencia de cada palabra en un documento se atribuye a una de las categorías a las que pertenece el documento.

En (Nyalala, y otros, 2019) se introdujo un método de predicción de la masa y el volumen de tomates Cherry basado en un sistema de visión por computadora y algoritmos de aprendizaje automático. Se estableció una relación entre la masa y el volumen de los tomates, y se utilizó para estimar la masa en un conjunto de datos de prueba con un R2 de 0.9824 y un RMSE de 15.84 g. Se adquirieron imágenes de profundidad de los tomates en diferentes orientaciones y se extrajeron características mediante técnicas de procesamiento de imágenes.

Se desarrollaron cinco modelos de predicción de regresión basados en características de imagen 2D y 3D. El modelo Radial Basis Function – Support Vector Machine (RBF-SVM) superó a todos los modelos explorados con una precisión del 0.9706 (solo características 2D) y 0.9694 (todas las características) en la estimación de masa y volumen, respectivamente. La masa o el volumen predicho por el modelo luego se puede aplicar a la función de potencia masa-volumen establecida. Este sistema introducido puede aplicarse como una técnica precisa, consistente y no destructiva para la separación y clasificación en línea de tomates Cherry basada en masa, volumen o densidad.

En el artículo **Estimación de las ventas de aguacate mediante algoritmos de aprendizaje automático y datos meteorológicos.** (Rincón Patiño, Lasso, & Corrales, 2019), presentan un enfoque de ML para estimar la cantidad de aguacate Hass vendido en diversas ciudades de los Estados Unidos de América con granularidad mensual, utilizando datos climatológicos y registros históricos de ventas.

Para ello, los autores llevaron a cabo la evaluación de cuatro algoritmos: Regresión Lineal, Perceptrón Multicapa, Máquina de Vectores Soporte para Regresión y Modelo de Predicción de Regresión Multivariada

Mediante el uso del modelo de regresión multivariante, se logró crear una aplicación que permite a los productores y vendedores de aguacate el planificar las ventas a

través de la estimación de las ganancias en dólares y la cantidad de aguacates que se podrían vender en diversas ciudades del ya mencionado país.

Este modelo es escalable ya que permite aplicarse a cualquier ciudad de los Estados Unidos de América siempre y cuando se cuente con los datos tanto climáticos y de ventas en su modalidad histórica.

El ALMANACMEX (Baez-Gonzalez, Kiniry, & Williams, 2016) es una versión del modelo de simulación dinámica ALMANAC (Agricultural Land Management Alternatives with Numerical Assessment Criteria Model) presentado en la Figura 3, que tiene una interfaz que permite el uso de datos climáticos, de suelo, genotipo y manejo en la simulación de cultivos en México. Fue desarrollado para facilitar el uso del modelo por investigadores y técnicos del sector agropecuario en México.

Este paquete de software contiene una base de datos climáticos con información histórica de parámetros mensuales y datos diarios de más de 3000 estaciones climáticas en México. La base de datos de suelo está formada con información sobre tipos de suelo en áreas agrícolas, pecuarias y forestales de México y sus características fisicoquímicas. Similar a la de otras versiones del ALMANAC, la base de datos contiene información de más de 80 cultivos existentes en México

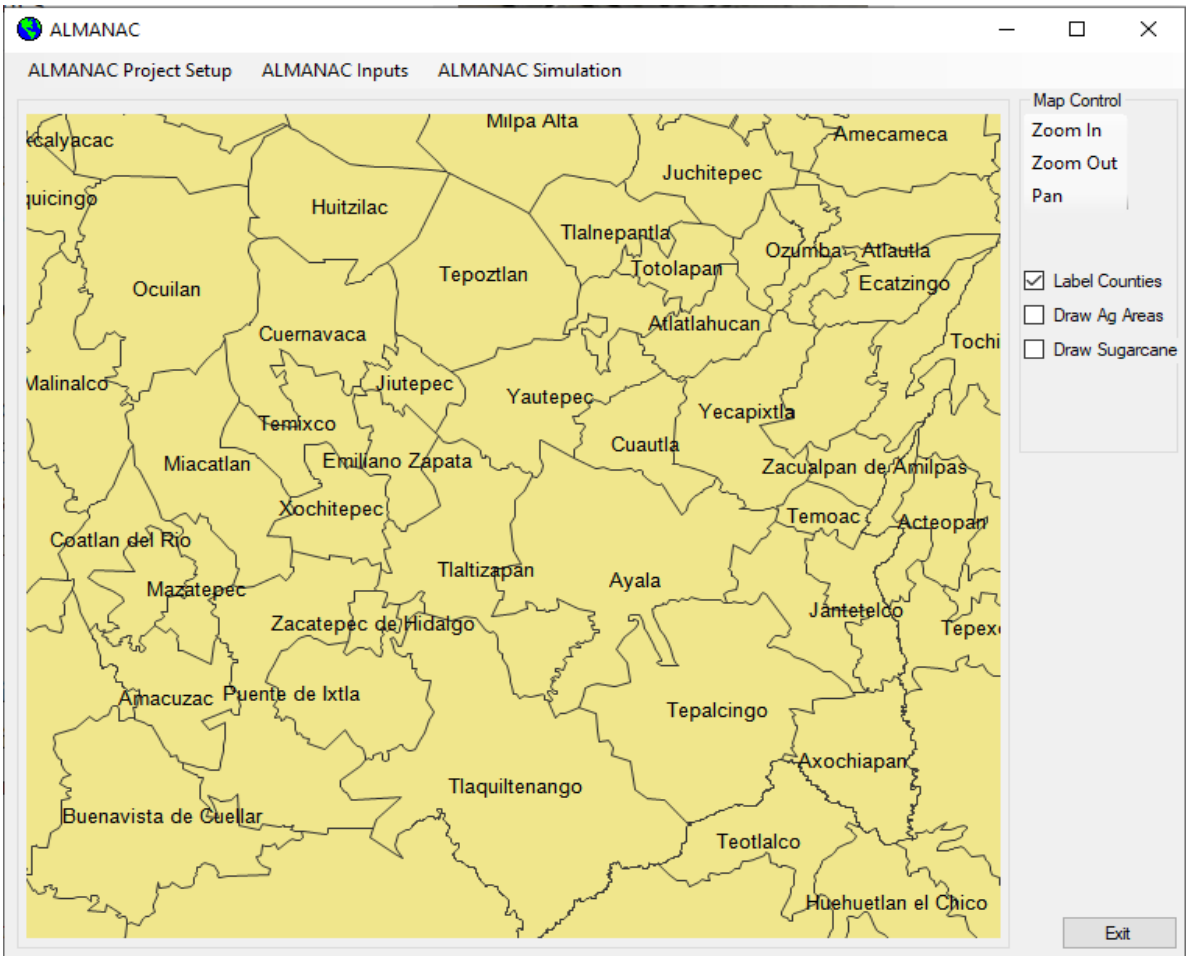

Figura 3 Vista previa de la interfaz de usuario del sistema ALMANACMEX

2.2 Internet de las cosas (IoT)

El paradigma IoT se ha desarrollado como un componente esencial de la era transformadora de los cambios tecnológicos en las ciudades inteligentes y rurales. IoT se define como la interacción de los mundos físico y digital a través de varios sensores y actuadores. IoT es un paradigma de capacidades informáticas y de redes integradas en cualquier objeto concebible con capacidades para modificar su estado.

En este contexto, IoT se refiere a dispositivos y periféricos conectados a una red de forma colaborativa para ejecutar una tarea con alta inteligencia a través de sensores, actuadores, procesadores y transceptores integrados.

La arquitectura básica para IoT considera tres capas: capa de percepción, capa de red y capa de aplicación. Sin embargo, se requieren capas adicionales para considerar adecuadamente todos los aspectos involucrados en aplicaciones realistas y de investigación.

La arquitectura de cinco capas incluye capas adicionales: la capa de red se separa en capas de transporte y procesamiento, y se incluye una nueva capa comercial para administrar la plataforma IoT implementada.

Las cinco capas son de acuerdo con (Espinosa, Ponte, Gibeaux, & González, 2021):

1. Capa de percepción. La capa de percepción es la capa física y proporciona la función de convertir señales analógicas en forma digital y viceversa.

Además, esta capa es responsable de detectar y recopilar información sobre el entorno a través de sensores y actuadores. Los sensores se utilizan para detectar cambios ambientales por parámetros físicos como la humedad o la temperatura para transformarlos en señales eléctricas. Los actuadores permiten transformar una señal eléctrica en acciones físicas a través de una máquina o dispositivo.

2. Capa de Transporte. La capa de transporte es responsable de las comunicaciones de transporte entre los componentes inteligentes de la red IoT, incluidos los dispositivos de red y las computadoras. Esta capa transmite datos de sensores recopilados desde la capa de percepción a la capa de procesamiento a través de redes de sensores inalámbricas como 3G/4G/5G, RFID, NFC, LAN y Bluetooth,.

3. Capa de procesamiento. La capa de procesamiento almacena, analiza y procesa un gran volumen de datos de la capa de transporte. Las tecnologías utilizadas en la capa de procesamiento son computación en la nube, computación distribuida, bases de datos y Big data.

4. Capa de aplicación. La capa de aplicación proporciona servicios al usuario final. Las aplicaciones de IoT se implementan en ciudades inteligentes, hogares inteligentes, salud y agricultura inteligentes.

5. Capa empresarial. La capa empresarial administra las aplicaciones de servicio basadas en IoT a través de modelos comerciales y de ganancias.

2.3 Área de Estudio

A continuación, se describe el área considerada para el estudio de caso.

La huerta de aguacate “La Ceiba” está ubicada en el municipio de Huitzilac, en el Noreste del estado de Morelos, México (coordenadas UTM 19.008174, -99.268800).

El terreno tiene una altitud de 2 273 metros sobre el nivel del mar. El clima tiene una temperatura promedio entre 14 °C y 22 °C durante todo el año. La huerta de aguacate está dividida en cuatro zonas, y la zona en producción se utilizó para la validación de la plataforma IoT propuesta.

La zona de producción contiene árboles de aguacate de diez años. Las demás zonas tienen árboles jóvenes sin producción.

El área de estudio se fertiliza cada cuatro meses. De esta manera, se estudian dos árboles de aguacate a través de los nutrientes del suelo. Un árbol de aguacate se fertiliza con un promedio de 21,77 gramos por árbol (g/t) de N, 4,7 g/t de P y 39,16 g/t de K. El otro árbol de aguacate estudiado no se fertiliza.

La Fig. 4 muestra la ubicación geográfica de la huerta de aguacates estudiada. La imagen fue tomada de Google Earth y utilizada solo con fines académicos, de acuerdo con el copyright de "uso justo".

Figura 4 ubicación geográfica de la huerta de aguacates (Alphabet Inc., 2023)

Se cuenta con apoyo del propietario de la UPA en facilidades como disponibilidad total para el uso de esta, así mismo, se cuenta con un módulo de Arduino mega para poder realizar el monitoreo de nutrientes y temperaturas.

Selección de la huerta piloto.

La decisión de seleccionar nuestra huerta piloto como objeto de estudio UPA responde a varias necesidades del proyecto.

1.- Contar con una UPA dentro de los límites territoriales establecidos, dígase esto, dentro de la zona norte del Estado de Morelos.

2.-La UPA debía de encontrarse dentro de una zona productora de aguacate o cuyo clima respondiera a las exigencias de la misma planta.

3.-La UPA debe de encontrarse en producción, esto con el fin de tener datos reales en cuanto a producción se refiere, para poder ser utilizado en futuros modelos y comparaciones.

CAPÍTULO 3 METODOLOGÍA DE INVESTIGACIÓN

En este apartado se presenta la metodología bajo la cual se desarrolló esta investigación. Se inició con la ubicación del área de estudio y se seleccionó la UPA con las mejores condiciones para su estudio; esto es una altura adecuada, variando entre los 2000 y 2300 metros sobre el nivel del mar, una temperatura templada que oscile entre los 15 y los 25 grados Celsius y suelos con un pH neutro, ni muy ácidos ni muy básicos.

Dicha UPA se encuentra dividida en 4 zonas, de las cuales sólo una es la productiva. Se cuenta con plantas de diversas edades, lo que permitirá predecir la producción de las zonas con plantas más jóvenes con relación a las ya productoras. De esa zona, se tomará una muestra de 2 plantas para el desarrollo del estudio. La edad de dichas plantas oscila entre los 8 y 10 años, lo cual las sitúa en un rango de jóvenes tardías, pero con un nivel de producción de fruto adecuado.

Posteriormente se seleccionaron los árboles que iban a ser utilizados de acuerdo con las características antes mencionadas, con el objetivo de escoger los ejemplares más convenientes para la colocación de sensores de monitoreo.

La estación propuesta para monitorear la información del cultivo de aguacate se implementa utilizando MQTT y Arduino, de esta manera se logra conseguir datos para su posterior limpieza, análisis y clasificación. El protocolo MQTT es un protocolo de conectividad que se utiliza para Máquina a Máquina (M2M) e Internet de las cosas (IoT). Así mismo es un protocolo de mensajería liviano que funciona con un mecanismo de publicación-suscripción basado en un broker y se ejecuta sobre el Protocolo de Control de Transmisión/Protocolo de Internet (TCP/IP) (Hillar, 2017). La estación fue montada en la huerta designada, utilizando los árboles previamente seleccionados, ya instalada junto a los sensores, se puso en operación y se realizaron pruebas de conexión.

Una vez realizada esta acción, se procedió a la obtención de datos, siendo esta mediante lecturas de un dispositivo IoT elaborado en base Arduino. Los datos son recolectados mediante sensores análogos

3.1 Estación IOT

Estructura general del sistema. Basada en las primeras tres capas de la arquitectura de cinco capas para IoT, la arquitectura de IoT propuesta se divide en tres etapas (Moreno-Bernal, y otros, 2022).

La primera etapa corresponde a la recolección de datos de sensores de nutrientes del suelo y condiciones climáticas a través del dispositivo IoT como estación de monitoreo. La información recopilada contiene los siguientes datos: fecha/hora, temperatura del suelo, temperatura del área, humedad del suelo, lúmenes, nitrógeno (N), fósforo (P) y potasio (K) de dos árboles de aguacate. Un árbol de aguacate se fertiliza y el otro consume los nutrientes naturales del suelo. Los sensores utilizados son:

- Módulo GSM SIM808. La finalidad de este elemento es para poder conectarse a la red 3G de datos móviles con la finalidad de enviar los datos recolectados a un servidor utilizando el protocolo MQTT.
- Sensor fotorresistor que convierte la señal análoga de lecturas de luz a lúmenes.
- Sonda de temperatura climática (ambiente) con sensor modelo HiLetgo DS18B20
- Sensor de humedad del tipo HD-38 para registrar la humedad del suelo.
- Módulo convertidor RS-485 para transformar lecturas analógicas a digitales compatibles con Arduino.
- Sensor de nutrientes NPK Soil para registrar las lecturas del suelo.
- Tarjeta de almacenamiento de memoria extraíble.

Todos estos componentes están integrados en una plataforma fácil de implementar, adaptada para ser utilizada en cualquier instalación agrícola. La plataforma de IoT propuesta es energéticamente eficiente: el consumo de energía total estimado de la infraestructura implementada es de solo ~365 mA (0,0438 kWh). El costo total de la plataforma construida es de aproximadamente 250 USD.

La segunda etapa envía la información recopilada del dispositivo IoT cada intervalo de tiempo a un servidor MQTT. Se utiliza un dispositivo de almacenamiento local para registrar datos como copia de seguridad.

En la última etapa, la limpieza de datos se realiza mediante un script de Python para futuros análisis.

3.2 Componentes de software.

La plataforma IoT propuesta para monitorear la información del cultivo de aguacate se implementa utilizando las bibliotecas MQTT y Arduino. El agente MQTT está disponible mediante un servidor Mosquitto en un host en la nube con una dirección IP pública. La función principal del corredor MQTT es filtrar los mensajes entrantes del cliente del dispositivo IoT de MQTT en la zona de Huitzilac. Luego, se ejecuta un script de Python para guardar los datos en un archivo CSV en un disco de almacenamiento local. Una vez que se guardan los datos, se realiza un proceso de limpieza de datos para procesar los datos recopilados, con el fin de verificar la coherencia y la redundancia de la información almacenada para futuros análisis.

La Fig. 5 describe la arquitectura IoT propuesta para monitorear los nutrientes y las condiciones climáticas de la producción de aguacate. En esta figura se identifican los principales componentes de hardware y productos de software utilizados.

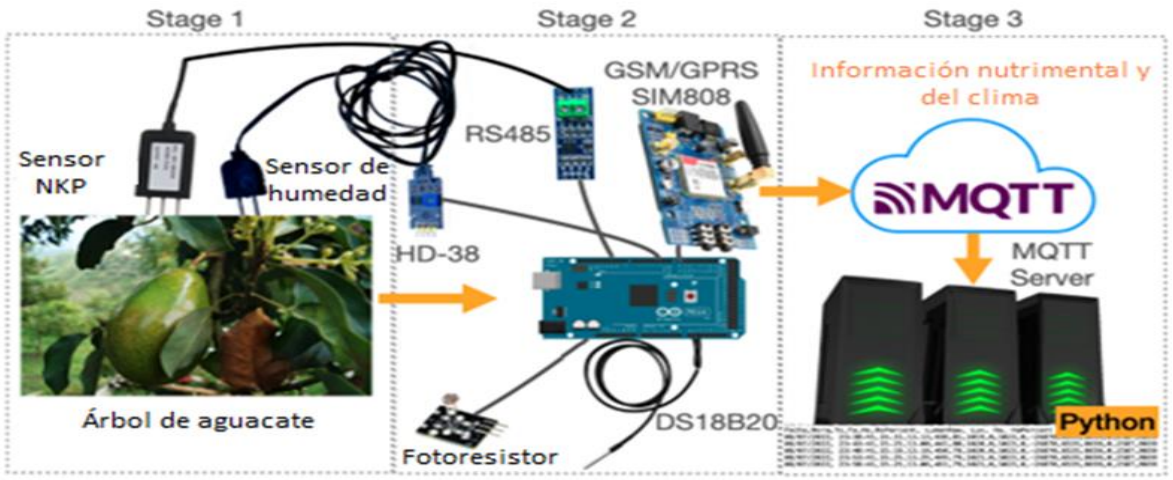

Figura 5 Arquitectura IoT propuesta

Las fotografías de la Fig. 6 presenta fotografías del despliegue real (en producción) de la plataforma IoT propuesta para el monitoreo de información del cultivo de aguacate en Huitzilac, Morelos.

La fotografía en la Fig. 6(a) muestra el sensor NPK del suelo en uno de los árboles de aguacate que se están monitoreando. La fotografía de la Fig. 6(b) presenta el dispositivo IoT. La fotografía en la Fig. 6(c) muestra la protección del escudo de metal del dispositivo IoT colocado en el huerto de aguacates. Finalmente, la fotografía en la Fig. 6(d) muestra el sensor Soil NPK envuelto en un escudo de plástico para protegerlo de roedores y factores externos.

La plataforma IoT desplegada permitió recopilar datos relevantes sobre el cultivo de aguacate.

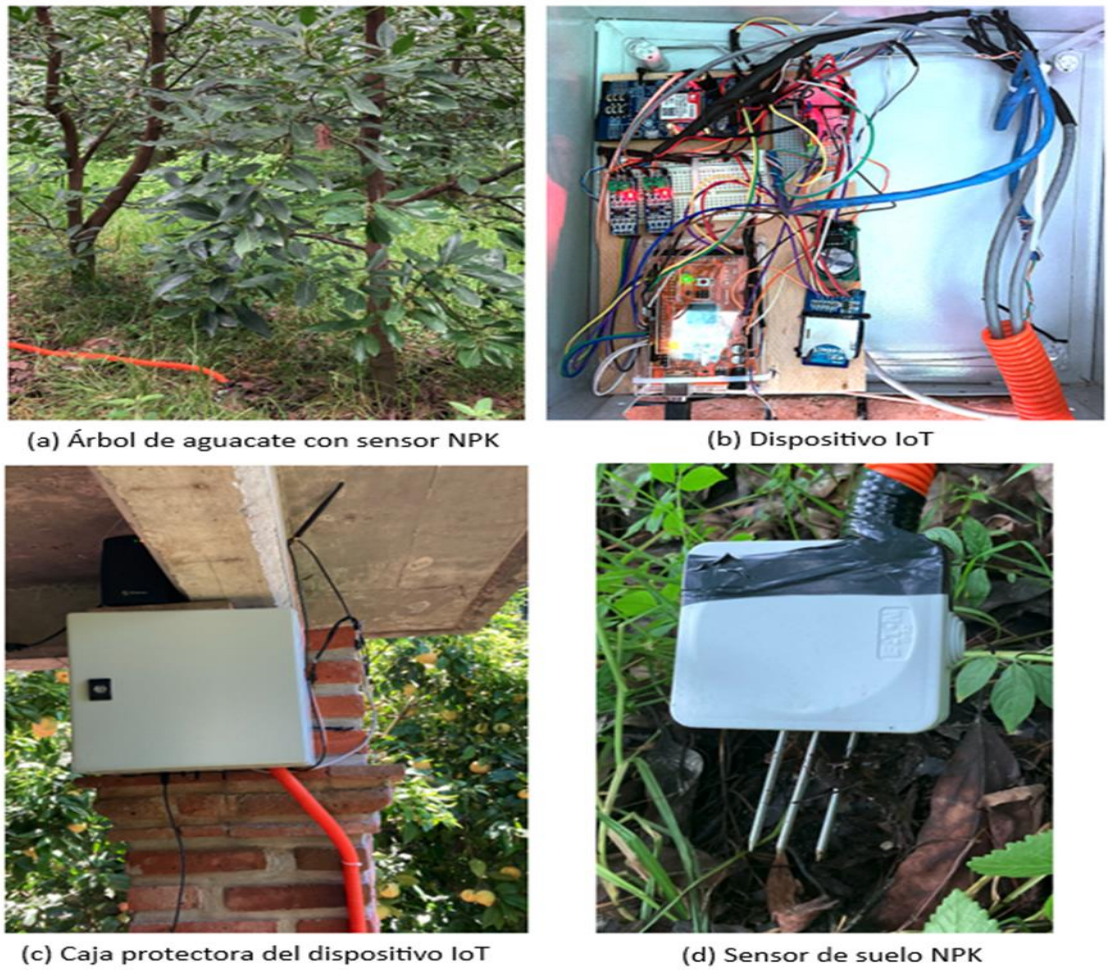

Figura 6 Plataforma IoT propuesta

La Fig. 7 presenta una fotografía de la estación IoT y sus componentes.

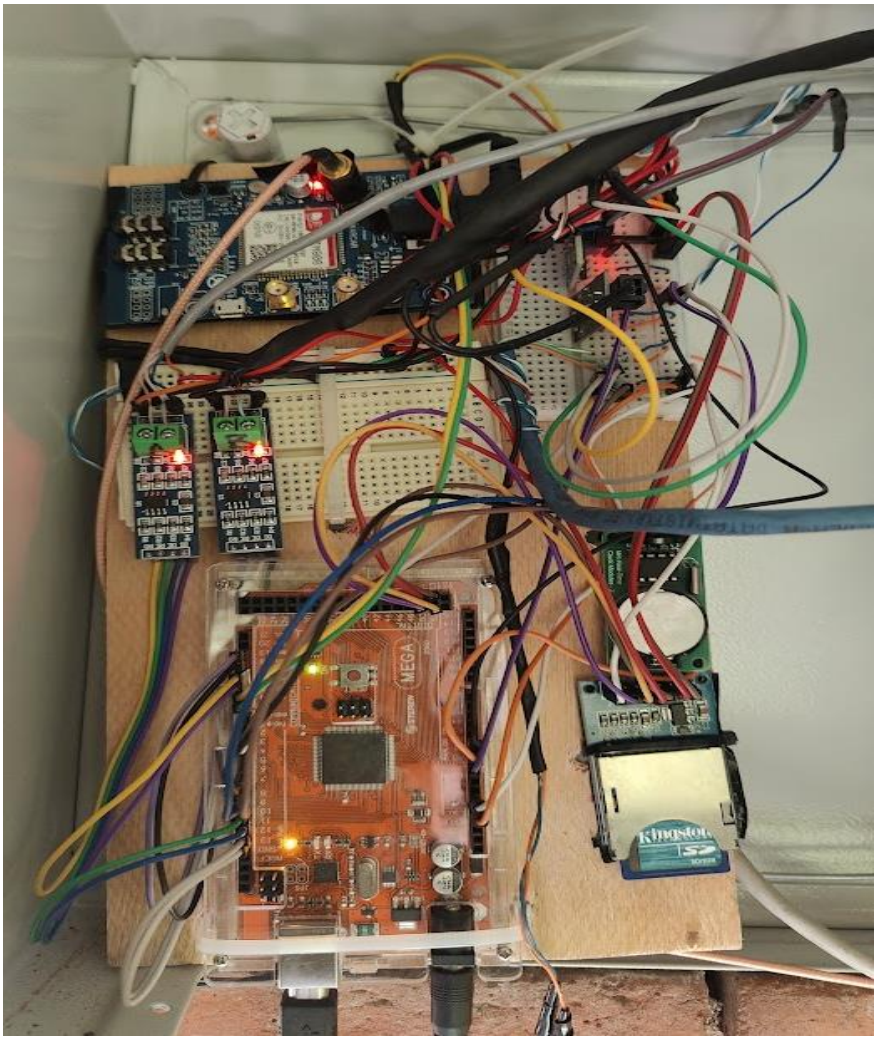

Figura 7 Estación IoT y sus componentes

3.3 Pipeline /Esquema de trabajo para la realización de Machine Learning

La Fig.8, muestra el esquema de cómo se han trabajado y procesado los datos, desde su obtención hasta que se generan resultados con ellos mismos

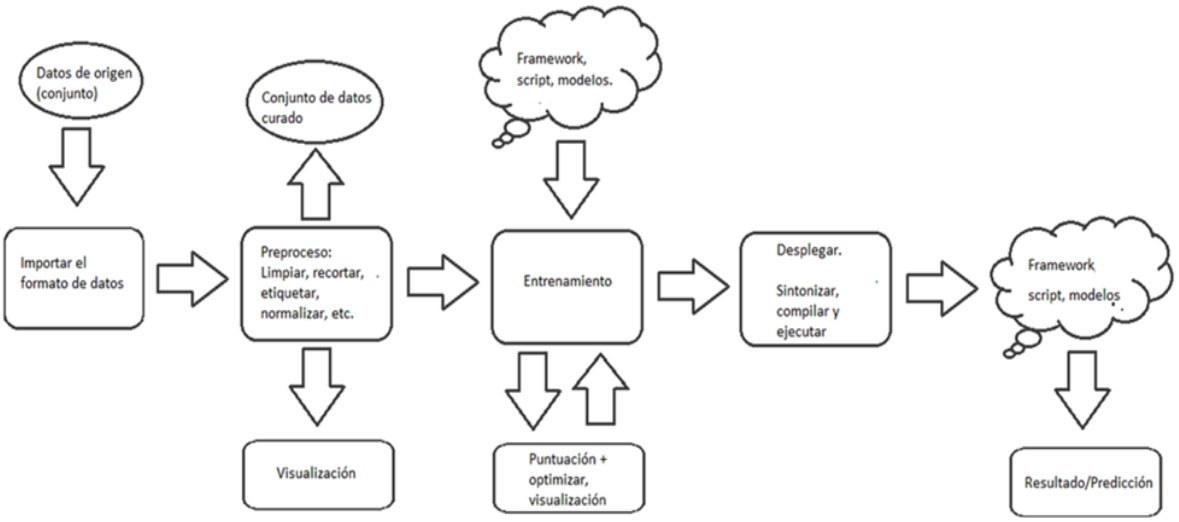

Figura 8 Esquema de trabajo basado en (Nvidia, 2022)

3.4 Obtención de los datos

Los datos son registrados cada cinco minutos mediante la estación de monitoreo montada base Arduino, los cuáles son almacenados de dos maneras: local mediante una memoria de respaldo extraíble y en un servidor privado vía MQTT. Dichos datos son variables como las temperaturas del suelo y del ambiente, datos como el nitrógeno, el fósforo y el potasio de los dos árboles bajo estudio.

A continuación, se muestra la Tabla 1 donde se incluyen datos a obtener, dichos datos son utilizados en este estudio, tales como las temperaturas, las humedades, luminosidad y ciertos nutrientes del suelo. Dichos datos han sido utilizados antes en trabajos de investigaciones que han sido utilizadas como referencia.

Tabla 1 Variables de investigación

Temperatura del suelo, temperatura ambiental, humedad relativa, humedad del suelo, viento	De acuerdo con los autores (Diaz, Mazza, Combarro, Giménez, & Gaiad, 2017), las temperaturas máxima y media, la evapotranspiración de referencia, la velocidad del viento y la humedad relativa, son las variables meteorológicas de mayor influencia en la masa fresca y el diámetro ecuatorial de los frutos. En huertos de cítricos de clima templado, las lluvias otoñales mejoran el tamaño final de los frutos y el contenido de jugo, y reducen la concentración de azúcares y ácidos libres.
Temperaturas, humedades y precipitación.	En (Rincón Patiño, Lasso, & Corrales, 2019) se utilizaron datos de la plataforma Weather Underground como fuente de datos meteorológicos, más específicamente, se utilizaron datos de temperatura (máxima, mínima y media), humedad (máxima, mínima y media) y precipitación.
Humedades, pH, nutrientes del suelo	Las propiedades del suelo están directamente relacionadas con las condiciones geográficas y climáticas de la tierra en uso y, por lo tanto, es un factor importante para tener en cuenta. La predicción de las propiedades del suelo consiste principalmente en predecir los nutrientes en el suelo, la humedad de la superficie del suelo, las condiciones climáticas durante el ciclo de vida del cultivo. Un análisis científico de los nutrientes del suelo, la humedad del suelo, el pH es importante para determinar las propiedades del suelo. (Sharma, Jain, Gupta, & Chowdary, 2021)

Temperatura, humedad del suelo y ambiental,	La estimación de la temperatura y la humedad del suelo también se encuentran entre las aplicaciones de los modelos de aprendizaje automático para la gestión del suelo. Además, se ha puesto de moda el uso de modelos de aprendizaje automático para resolver problemas relacionados con el manejo del ganado y las cose-chas. (Nosratabadi, Ardabili, Lakner, Mako, & Mosavi, 2021)

3.5 Datos de Origen

La primera etapa corresponde a la recolección de datos de sensores de nutrientes del suelo y condiciones climáticas a través del dispositivo IoT. Un árbol de aguacate se fertiliza y el otro consume los nutrientes naturales del suelo.

El envío de los datos a un servidor MQTT se hace vía internet, para esto, se utiliza una SIM instalada en la estación, la cual tiene una conectividad a internet del tipo 3G, dadas las condiciones geográficas de la UPA.

Los datos sin procesar son emitidos por el programa de control Arduino del dispositivo IoT. Estos datos están codificados en ASCII, almacenados en un archivo, con campos delimitados por comas (formato CSV).

Dichos datos son recolectados cada 300 segundos, o lo que es lo mismo, en intervalos de cinco minutos. De esta manera los cambios que pudiesen llegar a ocurrir serían más perceptibles y notables para las técnicas a utilizar.

Así mismo, ciertos datos preliminares, como los valores de cobre, zinc, sodio, hierro, entre otros elementos, se obtuvieron de análisis de uso de suelo previos, provistos por el dueño de la UPA.

3.6 Importar los datos.

Los datos son importados desde el servidor a un equipo de cómputo donde puedan ser preprocesados.

Dichos datos son extraídos del servidor MQTT y se procede a su posterior análisis y lectura. Los datos obtenidos Inicialmente se descargan en un documento CSV delimitado por comas

Esta etapa empieza con la extracción de datos del archivo en formato .txt de la memoria SD de la estación censora. Los datos obtenidos deben de ser preprocesados. Inicialmente las lecturas se vacían en un documento CSV delimitado por comas, y son tratados con un lector de hojas de datos (tales como Microsoft Exel o Lector de hojas de LibreOffice) para tenerlos en un formato más legible y amigable para el usuario.

3.7 Preprocesamiento

Los datos se obtienen en bruto, y tienen que ser analizados de manera manual para detectar errores, tales como datos faltantes, duplicados, erróneos, corruptos o cualquier otra eventualidad que impida que dichos sean procesados por algún lector de hojas de cálculo o bases de datos.

3.8 Limpieza

La limpieza de datos implica identificar datos duplicados, corregir errores de ortografía y sintaxis, y corregir campos vacíos y valores negativos o nulos. Los errores de datos ocurren cuando el dispositivo IoT tiene una interrupción en el suministro de energía por un factor externo en el área rural o cuando las condiciones ambientales afectan los sensores produciendo una información de lectura errónea.

Dicho proceso prepara los datos recopilados por el dispositivo IoT propuesto para su análisis mediante la eliminación o modificación de registros erróneos. La Figura 9 muestra los valores que se colectan de la estación de monitoreo.

Figura 9 Valores colectados, obtenidos directamente de la estación de monitoreo

```
Fecha,Hora,Ts,Ta,Hs,HsPercent, LumenRaw, Lux, Ha, HaPercent, NSA, PSA, KSA, NSB, PSB, KSB
08/05/2022, 11:11:53,13.44,17.50,392,87,44,1113,1022,0,28673,16158,8703,151,2050,8703
08/05/2022, 11:11:59,13.44,17.50,418,83,40,1230,1017,0,28673,16158,8703,151,2050,8703
08/05/2022, 11:12:06,13.44,17.50,408,85,43,1140,1020,0,28673,16158,8703,151,2050,8703
08/05/2022, 11:12:13,13.38,17.56,424,83,43,1140,1020,0,28673,16158,8703,151,2050,8703

Fecha,Hora,Ts,Ta,Hs,HsPercent, LumenRaw, Lux, Ha, HaPercent, NSA, PSA, KSA, NSB, PSB, KSB
08/05/2022, 11:12:31,13.44,17.69,324,96,42,1169,1019,0,28673,16158,8703,151,2050,8703

Fecha,Hora,Ts,Ta,Hs,HsPercent, LumenRaw, Lux, Ha, HaPercent, NSA, PSA, KSA, NSB, PSB, KSB
08/05/2022, 11:12:41,13.44,17.69,367,90,42,1169,1019,0,28673,16158,8703,151,2050,8703
08/05/2022, 11:12:47,13.50,17.75,422,83,42,1169,1019,0,28673,16158,8703,151,2050,8703
08/05/2022, 11:12:54,13.50,17.75,412,84,42,1169,1019,0,28673,16158,8703,151,2050,8703
Fecha,Hora,Ts,Ta,Hs,HsPercent, LumenRaw, Lux, Ha, HaPercent, NSA, PSA, KSA, NSB, PSB, KSB
Fecha,Hora,Ts,Ta,Hs,HsPercent, LumenRaw, Lux, Ha, HaPercent, NSA, PSA, KSA, NSB, PSB, KSB
08/05/2022, 11:15:42,13.50,17.44,412,84,40,1230,1017,0,-26870,6535,8659,0,2566,8659

Fecha,Hora,Ts,Ta,Hs,HsPercent, LumenRaw, Lux, Ha, HaPercent, NSA, PSA, KSA, NSB, PSB, KSB
08/05/2022, 11:25:44,13.63,17.87,421,83,40,1230,1019,0,-28918,28744,8659,0,2567,8659
08/05/2022, 11:30:44,13.63,17.31,419,83,40,1230,1017,0,-28918,28744,8659,0,2567,8659
08/05/2022, 11:35:44,13.63,16.94,435,81,41,1198,1021,0,-28918,28744,8659,0,2567,8659
08/05/2022, 11:40:44,13.69,17.37,438,81,42,1169,1020,0,-28918,28744,8659,0,2567,8659
```

La Figura 10, muestra los valores colectados “en bruto”, preparados como base de datos, para su tratamiento de limpieza.

Figura 10 Valores colectados con el proceso de limpieza aplicado

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Fecha	Hora	Ts	Ta	Hs	HsPercent	LumenRaw	Lux	Ha	HaPercent	Nitrum	Kalium	Phosphorus	Nitrum	Kalium	Phosphorus
2	01/10/2022	12:43:31	14.94	17.62	532	68	45	1087	1015	1	28673	16158	8703	159	2050	8703
3	01/10/2022	12:48:32	14.94	18.87	271	104	43	1140	1017	0	28673	16158	8703	159	2050	8703
4	01/10/2022	12:53:34	15	17.81	746	38	44	1113	1017	0	28673	16158	8703	159	2050	8703
5	01/10/2022	12:58:35	15.13	18.5	174	117	41	1198	1017	0	28673	16158	8703	159	2050	8703
6	01/10/2022	13:03:37	15.13	19.31	344	94	44	1113	1016	0	28673	16158	8703	159	2050	8703
7	01/10/2022	13:08:38	15.25	19.31	386	88	42	1169	1014	1	28673	16158	8703	159	2050	8703
8	01/10/2022	13:13:39	15.38	19.75	246	107	39	1262	1015	1	28673	16158	8703	159	2050	8703
9	01/10/2022	13:18:41	15.38	19	743	38	38	1297	1015	1	28673	16158	8703	159	2050	8703
10	01/10/2022	13:23:42	15.44	19.12	239	108	38	1297	1016	0	28673	16158	8703	159	2050	8703
11	01/10/2022	13:28:44	15.56	19.37	447	79	38	1297	1018	0	28673	16158	8703	159	2050	8703
12	01/10/2022	13:33:45	15.56	19.12	375	89	37	1333	1016	0	28673	16158	8703	159	2050	8703
13	01/10/2022	13:38:46	15.69	18.69	325	96	40	1230	1012	1	28673	16158	8703	159	2050	8703
14	01/10/2022	13:43:48	15.69	17.62	218	111	38	1297	1016	0	28673	16158	8703	159	2050	8703
15	01/10/2022	13:48:49	15.69	17.44	699	44	41	1198	1018	0	28673	16158	8703	159	2050	8703
16	01/10/2022	13:53:51	15.75	16.94	704	44	42	1169	1016	0	28673	16158	8703	159	2050	8703
17	01/10/2022	13:58:52	15.75	17.12	854	23	38	1297	1015	1	28673	16158	8703	159	2050	8703
18	01/10/2022	14:03:53	15.75	18.31	723	41	39	1262	1017	0	28673	16158	8703	159	2050	8703
19	01/10/2022	14:08:55	15.94	18.19	239	108	37	1333	1017	0	28673	16158	8703	159	2050	8703
20	01/10/2022	14:13:56	15.88	17.56	434	81	37	1333	1014	1	28673	16158	8703	159	2050	8703
21	01/10/2022	14:18:58	16	17.87	217	111	40	1230	1015	1	28673	16158	8703	159	2050	8703
22	01/10/2022	14:23:59	16.19	17.87	232	109	39	1262	1016	0	28673	16158	8703	159	2050	8703
23	01/10/2022	14:29:00	16.19	18.75	204	113	40	1230	1014	1	28673	16158	8703	159	2050	8703
24	01/10/2022	14:34:02	16.25	17	543	66	37	1333	1015	1	28673	16158	8703	159	2050	8703
25	01/10/2022	14:39:03	16.19	17.31	214	112	40	1230	1013	1	28673	16158	8703	159	2050	8703
26	01/10/2022	14:44:05	16.31	17.31	865	21	40	1230	1016	0	28673	16158	8703	159	2050	8703
27	01/10/2022	14:49:06	16.37	16.81	604	58	44	1113	1016	0	28673	16158	8703	159	2050	8703
28	01/10/2022	14:54:07	16.37	17.62	578	61	44	1113	1016	0	28673	16158	8703	159	2050	8703
29	01/10/2022	14:59:09	16.31	17.5	209	112	43	1140	1017	0	28673	16158	8703	159	2050	8703
30	01/10/2022	15:04:10	16.31	18.56	837	25	45	1087	1015	1	28673	16158	8703	159	2050	8703
31	01/10/2022	13:09:12	16.37	18	476	75	49	994	1017	0	28673	16158	8703	159	2050	8703
32	01/10/2022	13:10:58	16.44	18.25	576	61	46	1063	1017	0	28673	16158	8703	159	2050	8703
33	01/10/2022	15:15:59	16.44	18.19	442	80	45	1087	1015	1	28673	16158	8703	159	2050	8703
34	01/10/2022	15:21:01	16.44	18	715	42	44	1113	1017	0	28673	16158	8703	159	2050	8703
35	01/10/2022	15:26:02	16.5	17.94	370	90	47	1039	1015	1	28673	16158	8703	159	2050	8703
36	01/10/2022	15:31:03	16.56	17.62	701	44	52	934	1015	1	28673	16158	8703	159	2050	8703
37	01/10/2022	15:36:05	16.5	17.94	372	90	24	2083	1015	1	28673	16158	8703	159	2050	8703
38	01/10/2022	15:41:06	16.44	17.25	827	27	29	1715	1017	0	28673	16158	8703	159	2050	8703
39	01/10/2022	15:46:08	16.44	17.81	622	55	27	1846	1016	0	28673	16158	8703	159	2050	8703
40	01/10/2022	15:51:09	16.56	18.69	191	115	31	1601	1016	0	28673	16158	8703	159	2050	8703
41	01/10/2022	15:56:11	16.56	17.69	273	104	32	1550	1016	0	28673	16158	8703	159	2050	8703
42	01/10/2022	16:01:12	16.56	18	799	31	29	1715	1016	0	28673	16158	8703	159	2050	8703

A continuación, en el proceso de limpieza de datos, se verifica la coherencia y redundancia de los datos. Hay varias opciones para realizar acciones si falta alguno de estos, por ejemplo, puede ser que se normalice este mismo, escalando a un rango aproximando los límites superior e inferior de los datos o aproximando uniformemente en un rango. Los datos de las lecturas anteriores se pueden copiar en el lugar faltante si es que no hay una variación considerable en los valores. Dicho valor no presente se puede omitir, pero es preferible tener información diferente para los modelos posteriores. La figura 11 muestra los datos después del proceso de limpieza.

Figura 11 Datos después del proceso de limpieza

Fecha	Hora	Ts	Ta	Hs	HsPercent	LumenRaw	Lux	Ha	HaPercent	Nitrum	Kalium	Phosphorus	Nitrum	Kalium	Phosphorus
05/08/2022	11:11:07	13.38	17.44	317	97	41	1198	1019	0	28673	16158	8703	151	2050	8703
05/08/2022	11:11:13	13.44	17.44	350	93	44	1113	1020	0	28673	16158	8703	151	2050	8703
05/08/2022	11:11:20	13.44	17.37	359	92	40	1230	1020	0	28673	16158	8703	151	2050	8703
05/08/2022	11:11:27	13.38	17.37	367	90	40	1230	1020	0	28673	16158	8703	151	2050	8703
05/08/2022	11:11:33	13.44	17.44	342	94	42	1169	1022	0	28673	16158	8703	151	2050	8703
05/08/2022	11:11:40	13.38	17.44	362	91	42	1169	1020	0	28673	16158	8703	151	2050	8703
05/08/2022	11:11:46	13.38	17.44	364	91	41	1198	1019	0	28673	16158	8703	151	2050	8703
05/08/2022	11:11:53	13.44	17.5	392	87	44	1113	1022	0	28673	16158	8703	151	2050	8703
05/08/2022	11:11:59	13.44	17.5	418	83	40	1230	1017	0	28673	16158	8703	151	2050	8703
05/08/2022	11:12:06	13.44	17.5	408	85	43	1140	1020	0	28673	16158	8703	151	2050	8703
05/08/2022	11:12:13	13.38	17.56	424	83	43	1140	1020	0	28673	16158	8703	151	2050	8703
05/08/2022	11:12:31	13.44	17.69	324	96	42	1169	1019	0	28673	16158	8703	151	2050	8703
05/08/2022	11:12:41	13.44	17.69	367	90	42	1169	1019	0	28673	16158	8703	151	2050	8703
05/08/2022	11:12:47	13.5	17.75	422	83	42	1169	1019	0	28673	16158	8703	151	2050	8703
05/08/2022	11:12:54	13.5	17.75	412	84	42	1169	1019	0	28673	16158	8703	151	2050	8703

3.9 Preparación de datos

Inicialmente, los datos se obtienen con una granularidad de cinco minutos entre lectura y lectura, esto para tener una buena base de información. Se decidió tenerlo a esa distancia para un buen cronometraje de las horas totales del día, abarcando así aproximadamente 288 lecturas por día, entendiéndose un día como 24 horas.

Para un mejor procesamiento de los datos, las lecturas diarias se compilaron en una sola cada una, esto fue tomando todas las lecturas de un día para después promediarlas, con lo cual se obtuvieron un total de 436 registros ya sintetizados. El periodo de lecturas va desde el 1 de agosto del 2021 al 31 de marzo del 2023. Hay

lapsos que no se tienen registrados debido a cortes de luz prolongados, fallas en la red de datos entre otros percances fuera de nuestro control, debido a eso el número de registros es menor a l número de días totales.

Posteriormente, esos datos se empataron con los tiempos de los cortes; estos, al ser de una periodicidad diaria, se toman como base para poder llevar al mismo nivel las lecturas de la estación IoT. Por lo tanto, se procedió a promediar todas las lecturas en su día correspondiente, obteniendo un conjunto de datos apto para su posterior procesamiento.

3.10 Recorte

En este paso, se determina que datos son útiles y cuales no, en esta etapa se define el grado de importancia de cada categoría con la que se ha de trabajar y se procede a eliminar dichos datos, dejarlos en reserva o conservarlos.

Se determinó que los datos como la luminosidad no eran determinantes para la producción, ya que el sol es una constante que se mantiene durante todo el año, a pesar de los meses de lluvia, por lo cual fue eliminada en la etapa de procesamiento.

3.11 Resultados de la extracción de datos

El dispositivo IoT para monitorear árboles de cultivo de aguacate recopila datos sobre la temperatura del suelo, la temperatura del clima, la humedad del suelo y los lúmenes.

Además, recopila los datos NPK del suelo de dos árboles de aguacate.

El valor promedio de la temperatura del suelo fue de 15.57°C, y la temperatura del clima de 14.57°C, considerando una temperatura promedio en la zona entre 14°C y 22°C durante el año, lo que ayuda a mantener las raíces sanas y la floración.

El valor promedio de humedad del suelo fue de 75% (escala de 0 a 100) durante la época de lluvias. La cantidad promedio de luz visible recolectada por un sensor de fotorresistencia fue de 980 lúmenes durante la temporada de lluvias.

De los dos aguacates monitoreados, el valor promedio de N del árbol fertilizado (Sensor A) fue de 18405 mg/kg en comparación con el árbol no fertilizado (Sensor B), que tuvo un valor promedio de 129 mg/kg.

El valor medio de K para el sensor A fue de 14394 mg/kg en comparación con el sensor B de 2150 mg/kg. Finalmente, el valor de P promedio para ambos sensores fue de 8694 mg/kg.

3.12 Visualización de los datos

La Figura 12 muestra una representación visual de las lecturas recopiladas durante los primeros días del estudio

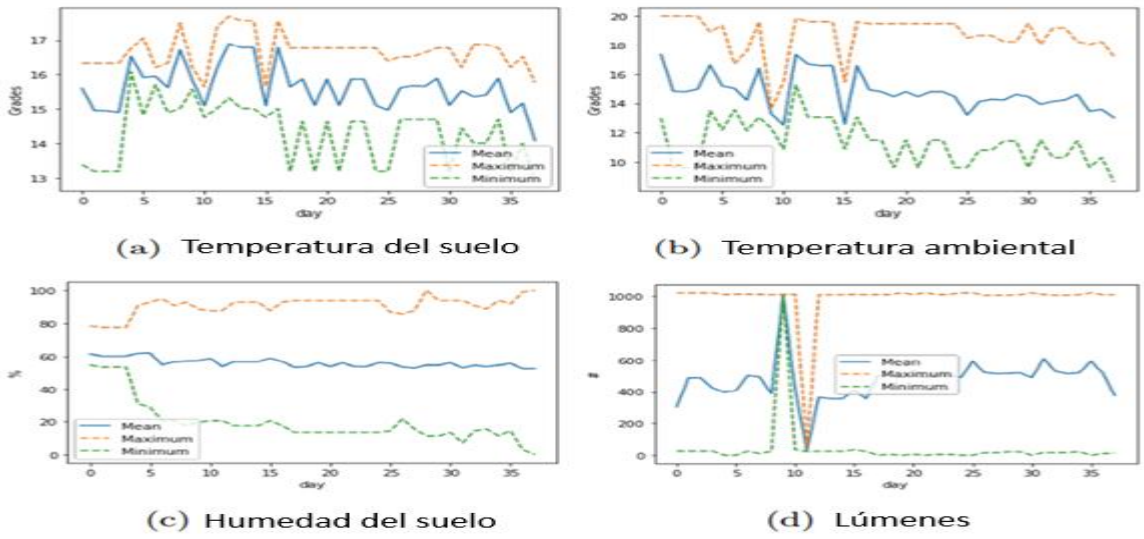

Figura 12 Visualización de las lecturas de datos

La Figura 13 muestra la producción en términos de peso y la figura 11 ilustra la producción en términos de kilos y unidades durante ese mismo lapso respectivamente.

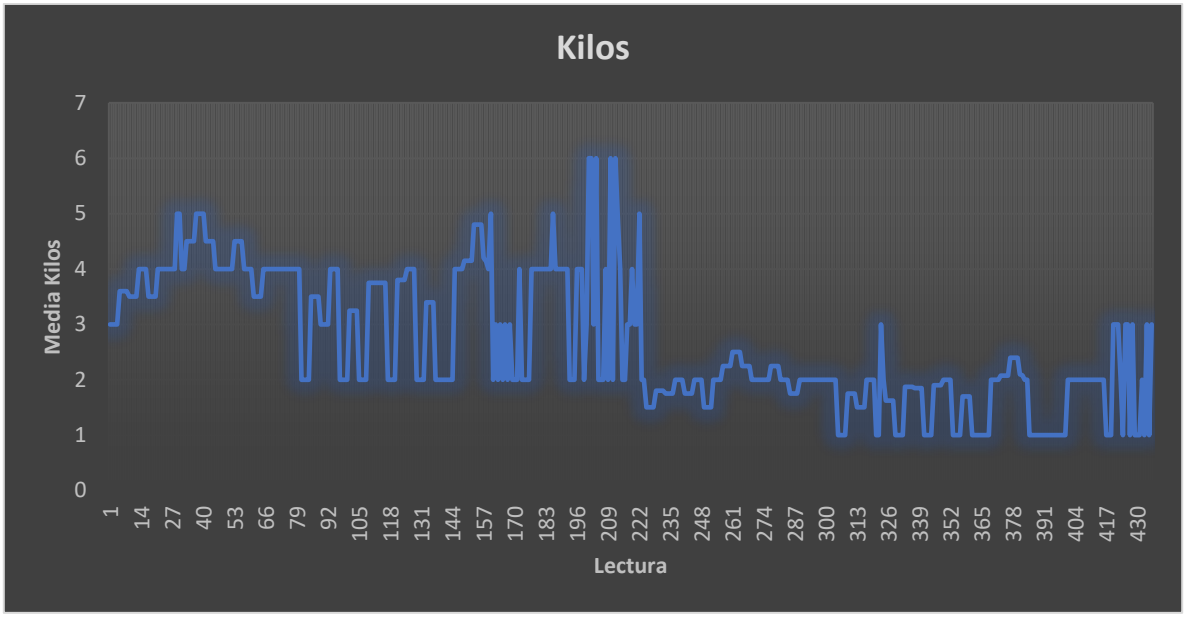

Figura 9 Producción kilos de aguacate

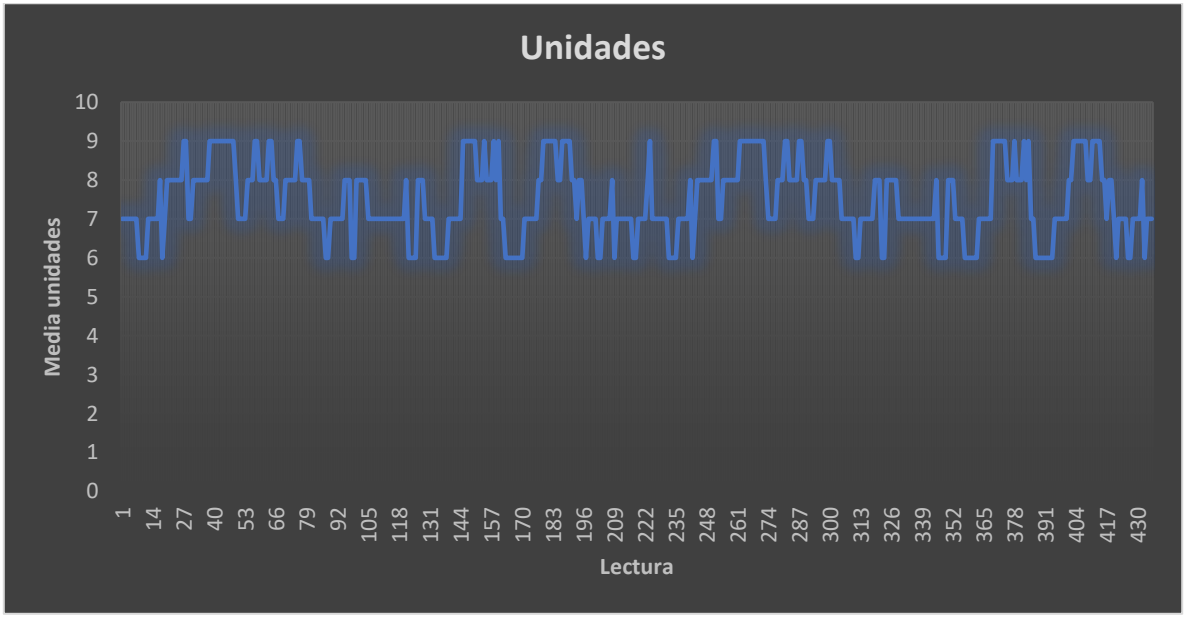

Figura 14 Unidades producidas

3.12 Modelos/Técnicas usadas

Se optó por utilizar técnicas de regresión, ya que se consideraron las más apropiadas para el tipo de investigación llevada a cabo, específicamente, técnicas supervisadas. Se descartaron los algoritmos de clasificación debido al tipo de datos manejados y los resultados esperados. Mientras que con los algoritmos de clasificación se pueden agrupar datos según ciertas características (por ejemplo, el calibre de la fruta), en este caso se busca analizar series de tiempo, siendo los algoritmos supervisados de regresión los más idóneos para esta tarea.

3.12.1 Algoritmos de regresión

En el contexto de la regresión, el proceso de aprendizaje automático implica la estimación y comprensión de las relaciones entre las variables. El análisis de regresión se centra en una variable dependiente y varias variables independientes, lo que lo convierte en una herramienta valiosa para la predicción y el pronóstico. (Martínez Heras, 2020).

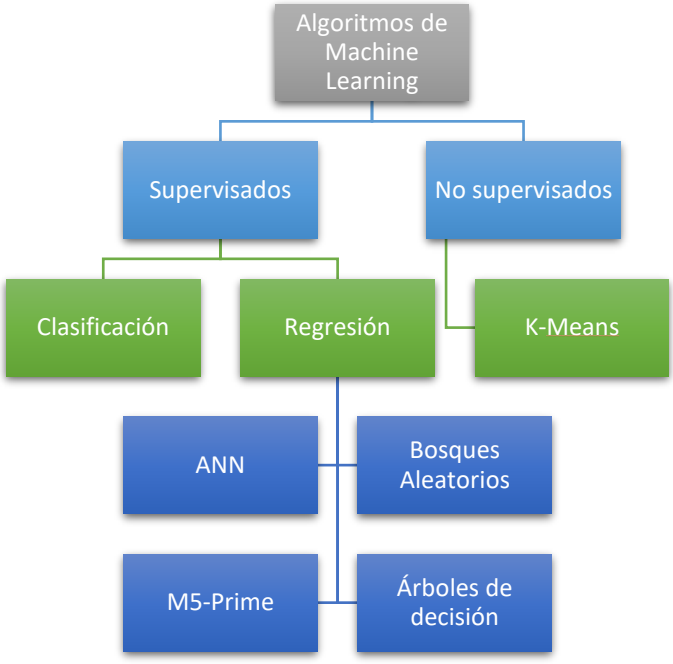

Figura 15 Algoritmos de Machine Learning utilizados.

3.12.2 Perceptrón multicapa

Se trata de una red neuronal artificial (RNA) que consta de múltiples capas, lo que le confiere la capacidad de abordar problemas que no son linealmente separables. En el contexto de las redes neuronales, esta arquitectura es una de las más ampliamente empleadas para resolver una variedad de problemas. Esto se debe a su capacidad como un aproximador universal y a su facilidad de uso. (López Herraiz, 2021).

Una red neuronal es un modelo computacional basado en el funcionamiento del cerebro humano. Consiste en una red de nodos interconectados, llamados neuronas, organizados en capas. Hay tres tipos que son principales en cuanto a las capas en una red neuronal se refiere: la capa de entrada, una, dos o más capas ocultas y la capa de salida. Así mismo, una red neuronal procesa información a través de capas interconectadas, ajusta los pesos de las conexiones durante el proceso de entrenamiento y es capaz de aprender y generalizar patrones a partir de datos. Este enfoque es particularmente poderoso para tareas como reconocimiento de patrones, clasificación y regresión en problemas complejos (Raschka & Mirjalili, 2017).

3.12.3 Máquinas de soporte vectorial

Las máquinas de vectores de soporte, también conocidas como máquinas de vector soporte, constituyen un conjunto de algoritmos de aprendizaje supervisado que se utilizan principalmente para problemas de clasificación y regresión. (Legorreta Anguiano, 2015)

Estos métodos están específicamente diseñados para abordar problemas de clasificación y regresión. Utilizando un conjunto de ejemplos de entrenamiento donde las clases están etiquetadas, podemos entrenar una SVM para construir un modelo que pueda predecir la clase de nuevas muestras. De manera intuitiva, una SVM es un modelo que representa los puntos de muestra en el espacio, separando las clases en dos espacios lo más amplios posible mediante un hiperplano de

separación. Este hiperplano se define como el vector entre los dos puntos más cercanos de las dos clases, conocidos como vectores de soporte. Cuando se asignan nuevas muestras a este modelo, en función de los espacios a los que pertenecen, pueden ser clasificadas en una u otra clase.

Las Support Vector Machines para regresión buscan encontrar un hiperplano que maximice el margen entre las predicciones y los valores reales. La elección del kernel y la configuración de los parámetros son aspectos clave en la aplicación efectiva de SVR.

3.12.4 Árboles de decisión

Un árbol de decisión es un algoritmo de aprendizaje supervisado no paramétrico que se utiliza tanto para clasificación como para regresión. Se caracteriza por su estructura jerárquica en forma de árbol, que consta de un nodo raíz, ramas, nodos internos y nodos hoja.

El proceso de construcción de un árbol de decisión comienza con un nodo raíz, el cual no tiene ramas entrantes. Las ramas que salen del nodo raíz conducen a los nodos internos, también conocidos como nodos de decisión. Estos nodos de decisión evalúan las características disponibles y dividen el conjunto de datos en subconjuntos más homogéneos. Estos subconjuntos se indican mediante nodos hoja o terminales. Los nodos hoja representan los resultados posibles dentro del conjunto de datos.

Los árboles de regresión funcionan dividiendo iterativamente los datos en subconjuntos y asignando valores numéricos a las hojas. La construcción del árbol busca minimizar el error cuadrático medio, y el modelo resultante puede ser utilizado para hacer predicciones numéricas en nuevos datos.

3.12.5 Bosques aleatorios

Los árboles de decisión son un tipo de algoritmo de Machine Learning supervisado muy utilizado debido a su precisión, simplicidad y flexibilidad. Su capacidad para realizar tanto tareas de clasificación como de regresión, junto con su naturaleza no lineal, lo convierte en una herramienta altamente adaptable para una variedad de datos y situaciones.

La razón por la que se les llama "bosques" es porque generan un conjunto de árboles de decisión. Posteriormente, los datos de estos árboles se combinan para asegurar predicciones más precisas. Mientras que un único árbol de decisión puede tener un resultado limitado y un conjunto reducido de grupos, un bosque de árboles de decisión garantiza resultados más precisos al considerar una mayor cantidad de grupos y decisiones. Además, añaden aleatoriedad al modelo al seleccionar la mejor característica entre un subconjunto aleatorio de características. Esto proporciona un beneficio adicional en términos de generalización y robustez del modelo. (Boters Pitarch, 2021)

3.12.6 M5 Prime

El árbol modelo M5 Prime es una variante de los árboles de decisión para la tarea de regresión, que es utilizado para predecir valores de la variable de respuesta numérica Y, además, es un árbol de decisión binario que tiene funciones de regresión lineal en los nodos terminales (hoja). (Diaz, Mazza, Combarro, Giménez, & Gaiad, 2017)

En la práctica, el algoritmo del modelo M5 Prime tiene dos pasos diferentes: etapas de crecimiento y etapa de poda. La etapa de crecimiento se usa para dividir nodos en función de los valores de los atributos involucrados; el objetivo principal es reducir el error de predicción de las respuestas numéricas (Y) en los nodos terminales y aumentar la profundidad del árbol.

La etapa de poda juzga cuánto contribuye cada atributo al error de predicción en un nodo y luego corta las ramas innecesarias.

3.13 Datos recolectados por la estación

El dispositivo IoT para monitorear árboles de cultivo de aguacate recopila datos sobre la temperatura del suelo, la temperatura del clima, la humedad del suelo y los lúmenes. Además, recopila los datos NPK del suelo de dos árboles de aguacate.

La figura 16 presenta los valores promedio por día de los datos recopilados por la plataforma IoT. El valor promedio de la temperatura del suelo fue de 15.90°C, y la temperatura del clima de 13.74°C, considerando una temperatura promedio en la zona entre 14°C y 22°C durante el año, lo que ayudó a mantener sanas las raíces y la floración.

El valor promedio de humedad del suelo fue de 75% (escala de 0 a 100) durante la época de lluvias. La cantidad promedio de luz visible recolectada por un sensor de fotorresistencia fue de 980 lúmenes durante la temporada de lluvias. De los dos paltos monitoreados, el valor promedio de N del árbol fertilizado (Sensor A) fue de 18405 mg/kg en comparación con el árbol no fertilizado (Sensor B), que tuvo un valor promedio de 129 mg/kg. El valor medio de K para el sensor A fue de 14394 mg/kg en comparación con el sensor B de 2150 mg/kg. Finalmente, el valor de P promedio para ambos sensores fue de 8694 mg/kg.

Figura 16 Valores máximos, mínimos y promedio de los datos recopilados por la plataforma IoT.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Lectura	Ts	Ta	Hs	HsPercent	LumenRaw	Lux	Ha	NS	PS	KS	Kilos	Unidades
2	1	13.7017993	12.6458824	868.356401	52.650519	556.930796	1002.79239	1013.69204	28673	16158	8703	3	7
3	2	13.447561	11.497561	894.947735	51.3763066	565.341463	483.703833	1018.40418	28673	16158	8703	3	7
4	3	13.8360208	12.6622145	911.228374	49.5536332	561.539792	489.747405	1018.3218	28673	16158	8703	3	7
5	4	13.9338542	13.2048611	923.569444	53.1458333	557.225694	557.736111	1017.82639	28673	16158	8703	3	7
6	5	14.3036364	13.2966434	932.346154	55.7587413	548.923077	1006.47552	1017	28673	16158	8703	3.6	7
7	6	13.3843554	11.4248084	940.881533	54.3066202	553.878049	1251.65505	1016.54355	28673	16158	8703	3.6	7
8	7	13.3778397	12.0272474	953.425087	54.2857143	555.550523	1014.74913	1015.86411	28673	16158	8703	3.6	7
9	8	13.1930314	12.1327526	962.808362	54.0836237	548.010453	538.682927	1015.3345	28673	16158	8703	3.6	6
10	9	13.4051916	11.9414983	971.034843	53.0034843	550.923345	1587.20209	1014.96167	28673	16158	8703	3.5	6
11	10	13.6718182	12.1989161	977.013986	52.1258741	552.699301	904.43007	1014.56993	28673	16158	8703	3.5	6
12	11	13.5717014	11.9213889	984.322917	50.6006944	556.017361	552.579861	1016.74653	28673	16158	8703	3.5	6
13	12	13.5954355	12.5762369	989.247387	49.7491289	552.578397	1252.61673	1018.38328	28673	16158	8703	3.5	7
14	13	13.8660839	11.9580769	993.300699	55.7937063	552.472028	1156.32867	1018.42657	28673	16158	8703	4	7
15	14	12.4070035	10.3046342	996.045296	54.9790941	555.752613	859.23345	1018.31011	28673	16158	8703	4	7
16	15	12.57223	11.101777	998.926829	54.28223	554.89547	876.641115	1018.37631	28673	16158	8703	4	7
17	16	13.5275175	12.1046154	1000.91259	54.2762238	552.678322	825.604895	1018.39511	28673	16158	8703	4	7
18	17	13.78	12.2088153	1002.66551	53.5400697	556.299652	829.926829	1018.40767	28673	16158	8703	3.5	8
19	18	13.6204181	11.5864112	1003.8223	52.815331	559.125436	766.188153	1018.37979	28673	16158	8703	3.5	6
20	19	13.6427875	12.4122648	1005.12195	51.4947735	554.369338	632.170732	1018.777	28673	16158	8703	3.5	7
21	20	13.9676655	12.5368641	1006.41463	49.9442509	557.937282	585.547038	1019.18815	28673	16158	8703	3.5	8
22	21	14.0033218	12.4104844	1007.21107	52.183391	553.865052	533.858132	1019.0519	28673	16158	8703	4	8
23	22	14.1705245	12.1741259	1007.07692	55.8636364	550.22028	993.758741	1018.42657	28673	16158	8703	4	8
24	23	14.007108	11.8386411	1007.90244	54.4808362	552.996516	984.135889	1018.44948	28673	16158	8703	4	8
25	24	13.9383217	12.0699301	1008.59091	54.4300699	553.216783	933.122378	1018.45804	28673	16158	8703	4	8
26	25	13.8662847	12.2123611	1009.36111	54.0208333	550.881944	953.659722	1018.46181	28673	16158	8703	4	8
27	26	13.6245007	11.0449618	1007.86231	53.151428	553.782631	683.204998	1018.51146	28673	16158	8703	4	8
28	27	13.5104181	11.1555052	1011.39024	52.2473868	552.571429	677.188153	1019.05575	28673	16158	8703	4	9
29	28	14.0060627	12.025993	1012.4007	50.7212544	556.512195	739.23345	1019.39373	28673	16158	8703	4	9
30	29	14.9122997	13.4170732	1013.16028	49.1149826	554.498258	744.815331	1019.38676	28673	16158	8703	5	7
31	30	15.6292308	14.1788112	1014.0979	55.972028	550.762238	747.108392	1019.36364	28673	16158	8703	5	7
32	31	15.6090592	13.5847387	1015	54.9547038	553.296167	726.43554	1019.38328	28673	16158	8703	4	8
33	32	15.0886063	12.5326481	1015.94425	54.2648084	552.02439	738.299652	1019.40418	28673	16158	8703	4	8
34	33	15.2386014	12.9353147	1016.17133	54.2307692	549.825175	821.881119	1019.25525	28673	16158	8703	4.5	8
35	34	15.5477004	12.8195819	1016.99652	53.6655052	547.947735	844.341463	1019.40767	28673	16158	8703	4.5	8
36	35	14.638007	11.3887413	1017.31818	52.9125874	548.835664	843.706294	1019.36364	28673	16158	8703	4.5	8
37	36	15.1290592	12.3437979	1017.62718	51.7770035	551.020906	905.74216	1019.35889	28673	16158	8703	4.5	8
38	37	15.0758188	12.2040767	1018.16028	50.0278746	550.212544	952.686411	1019.38328	28673	16158	8703	5	8
39	38	15.3945105	12.4482168	1018.5	51.4685315	547.41958	866.328671	1019.39511	28673	16158	8703	5	9
40	39	15.1918467	12.278676	1018.66202	55.8606272	548.209059	819.770035	1019.38676	28673	16158	8703	5	9
41	40	15.670453	12.8218467	1018.49477	54.6167247	544.923345	910.275261	1019.40767	28673	16158	8703	5	9
42	41	16.2878671	13.6693007	1018.63986	53.479021	541.615385	987.56993	1019.34615	28673	16158	8703	4.5	9

La figura 17 muestra la información obtenida de los cortes realizados, así como los valores de:

- Fecha del corte
- Kilos obtenidos en ese corte
- Unidades totales del día
- Ancho del mejor y peor ejemplar
- Alto del mejor y peor ejemplar
- Peso del mejor y peor ejemplar

Figura 17 Valores de los cortes de aguacate.

Fecha	Kilos	Unidades	Ancho_mejor	Alto_mejor	Peso_mejor	Ancho_peor	Alto_peor	Peso_Peor
24/09/2022	26.2	117	83.4	117.5	387	56.6	70.7	123
06/10/2022	21.02	93	80.3	111.5	310	55.7	73.3	138
08/10/2022	28.835	135	84.9	115.6	400	59	73	130
16/10/2022	19.802	90	80.9	114.9	357	60	77.4	143
24/10/2022	35.787	146	84.5	112	355	62.9	75.2	149
13/11/2022	32.355	137	85.3	125	428	61.9	77.1	149
16/11/2022	24.803	102	83.8	118.9	361	61	81.9	157
20/11/2022	24.875	113	87.7	113.5	418	59.2	70.5	126
27/11/2022	33.479	134	90	116.9	447	60	73.7	141
05/12/2022	27.397	120	84.6	117.5	394	60	74.5	141
08/12/2022	27.068	120	80	117.1	330	62	74.4	150
16/12/2022	25.271	105	82.7	109	334	64.3	80	160
01/01/2023	26.757	114	85.3	112.6	355	61.2	78.3	151
07/01/2023	27.675	125	83.7	108.4	357	55.1	77.5	127
15/01/2023	27.4	115	79.5	106.6	319	64.8	73.9	170
19/01/2023	26.343	112	84.3	122.3	387	62.3	78.3	161
27/01/2023	30.366	134	85.3	126.1	423	59.6	76.1	133
29/01/2023	34.932	167	80.2	110.2	301	58.6	70.8	132
08/02/2023	20.453	96	84.2	114.4	355	59.4	73.4	130
11/02/2023	30.796	119	88	118.9	421	58.7	79.6	147
25/02/2023	28.583	122	82.4	129.2	354	58.8	71	125
13/03/2023	34.156	156	83.7	117.4	365	56.8	70.5	112
21/03/2023	31.383	153	84.1	120.7	355	55.2	71.7	114
31/03/2023	23.385	128	82.9	113	352	50.5	72.4	101

Capítulo 4. Resultados y discusión

En este capítulo se tratan los resultados obtenidos después de analizar y tratar todos los datos obtenidos previamente.

4.1 Gráfica de correlación

Tras el análisis de los datos, podemos identificar las variables que tienen una mayor influencia en la producción tanto en términos de peso (kilos) como en unidades. La Figura 18 presenta una gráfica de correlación, comúnmente conocida como "gráfica de calor" o “mapa de calor”, que muestra todas las variables de control en uso. En esta representación gráfica, es importante destacar que cuanto más cerca el valor se encuentra de 1 o -1, mayor es la correlación, ya sea positiva o negativa.

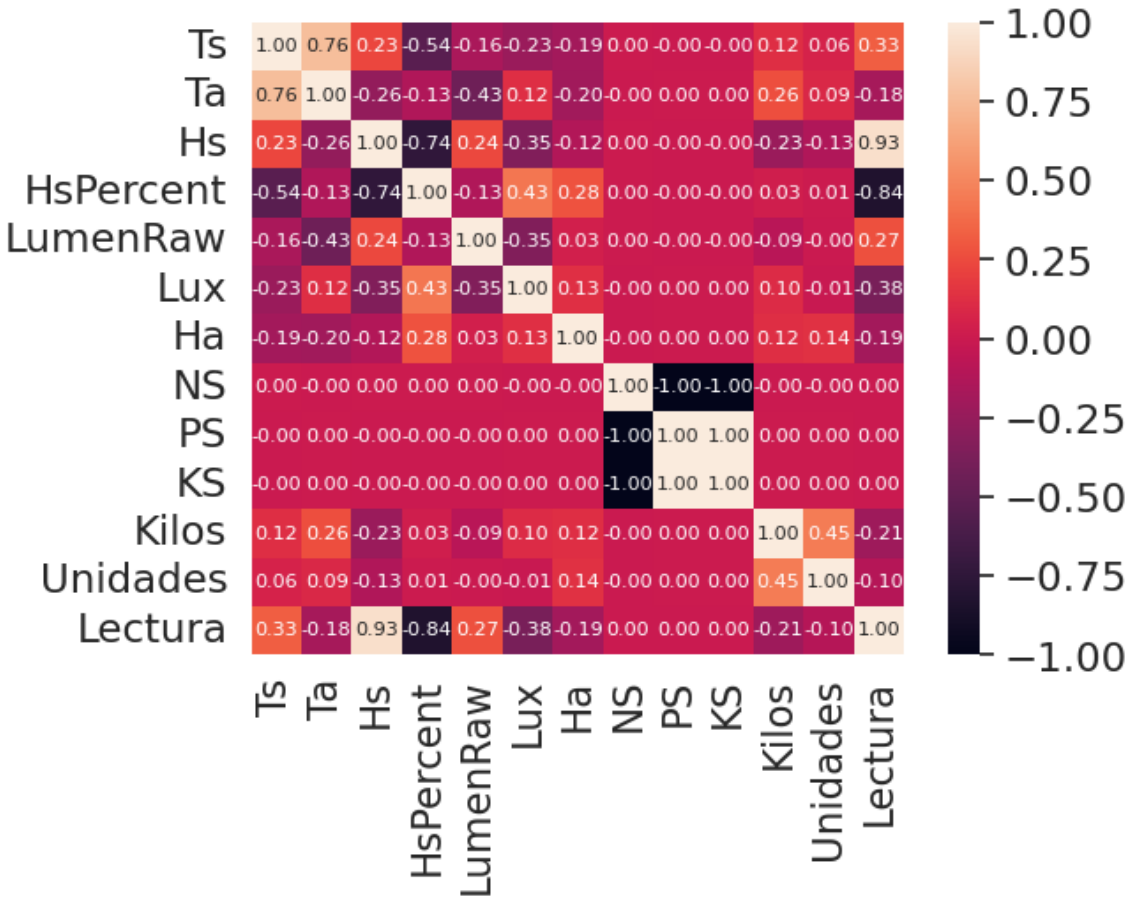

Figura 18 Gráfico de calor o correlación

Según los datos de esta gráfica, podemos concluir que las variables que ejercen la mayor influencia en la producción, en términos de peso (kilos) y unidades, son la humedad del suelo (HS) y la temperatura ambiente (Ta). Esto se debe a que, como se mencionó previamente, presentan los valores más significativos de correlación con la columna de kilos, siendo 0.26 puntos para la temperatura amiente y -0.23 puntos para la humedad ambiental.

4.2 Métricas de evaluación de datos

Para evaluar los resultados obtenidos de las distintas técnicas de ML se utilizaron las siguientes métricas.

- **4.2.1 R2 o Coeficiente de determinación**
    - Es una medida estadística fundamental que analiza en qué medida las variaciones en una variable pueden ser explicadas por las variaciones en una segunda variable al predecir el resultado de un evento específico. Esta métrica desempeña un papel crucial en la evaluación de modelos predictivos, ya que permite determinar cuánta de la variabilidad en los datos de respuesta se puede atribuir a la capacidad del modelo para realizar predicciones precisas. En esencia, el R^2 proporciona una medida de cuánto poder predictivo tiene el modelo y cuánta varianza en los datos puede ser explicada por las variables independientes incluidas en el modelo.
    - La puntuación más alta que se puede obtener en el coeficiente de determinación R^2 es 1, lo que indica una precisión total del modelo. Por otro lado, la peor puntuación posible es 0, lo que sugiere una falta total de coincidencia entre las predicciones del modelo y los datos observados. Sin embargo, es importante destacar que, en ciertos casos, esta puntuación puede ser negativa, lo que subraya la limitada capacidad del modelo para realizar predicciones precisas y su incapacidad para ajustarse adecuadamente a los datos de entrada.

- En el caso general, cuando la variable de respuesta (Y verdadera) no es constante y varía en función de las características de entrada, un modelo constante que simplemente prediga el valor promedio de Y sin considerar las características de entrada obtendría una puntuación del coeficiente de determinación R^2 de 0. Esto se debe a que, en ausencia de un modelo que explique la variabilidad en Y, el R^2 refleja la incapacidad del modelo constante para realizar predicciones precisas y se convierte en una métrica de referencia para evaluar la calidad de otros modelos en comparación con este escenario nulo. (Raschka & Mirjalili, 2017)

- **4.2.2 RMSE (Root Mean Square Error)**
  - Es una medida ampliamente utilizada para cuantificar las discrepancias entre los valores predichos por un modelo o estimador y los valores observados en una muestra o población.
  - La puntuación ideal que buscamos lograr para el Error Cuadrático Medio (RMSE) es 0, lo que señala la ausencia total de errores en las predicciones. Por otro lado, un valor cercano a 1, e incluso igual a 1, indica un alto nivel de error. La proximidad a 1 en la puntuación RMSE refleja un error significativo, y cuanto más cerca esté de 1, mayor será la magnitud de ese error. Sin embargo, es importante destacar que este error puede superar el valor 1, lo cual indica un nivel más elevado de error.
  - Cuando el Error Cuadrático Medio (RMSE) se combina con el coeficiente de determinación R^2, se obtiene un enfoque más completo en la evaluación del modelo. Esta combinación permite una comprensión más profunda de la calidad de las predicciones del modelo. Mientras que el RMSE cuantifica el tamaño promedio de los errores en las predicciones, el R^2 mide la proporción de la varianza en los datos de respuesta que es explicada por el modelo. Al considerar ambos juntos, se logra un mejor equilibrio en la evaluación

del rendimiento del modelo, lo que proporciona una visión más precisa de su capacidad para ajustarse a los datos y hacer predicciones precisas. (Scikit Learn, 2023).

4.3 Datos de entrenamiento y datos de prueba

Dentro de la metodología del proyecto, se utilizaron tanto los datos de entrenamiento como los datos de prueba como base para obtener los resultados. A pesar de mantener constantes los mejores hiperparámetros, se observaron diferencias en los resultados de R2 y RMSE.

A continuación, en las Tablas 7 y 8, se presentan los tiempos de ejecución en segundos de cada algoritmo de Machine Learning. Esto se hace tanto con los hiperparámetros estándar como con los hiperparámetros sintonizados para obtener los mejores tiempos de ejecución. El porcentaje de datos utilizado para estos experimentos son un 80% de datos de entrenamiento (train) y 20% datos de prueba (test). Las pruebas se corrieron en el ambiente en la nube Google Colab, la cual es una plataforma en línea gratuita que permite ejecutar y escribir código en Python.

Tabla 2 Tiempos de ejecución en segundos de cada uno de los algoritmos de ML

	Hiperparámetros estándar			
	R2 Train	RMSE Train	R2 Test	RMSE Test
Decision Tree	1	0.00E+00	0.18483509	1.00801476
Multilayer Regressor	-5.69864611	3.07768162	0.18483509	5.50729814
SVR	0.65325217	0.70022409	0.36148557	0.89213281
Random forest	0.95254399	0.25904535	0.50947322	0.78194375
M5 Prime	0.77643921	0.56224809	0.48048916	0.80471381

Tabla 3 Tiempos de ejecución en segundos de cada uno de los algoritmos de ML

	Hiperparámetros sintonizados			
	R2 Train	RMSE Train	R2 Test	RMSE Test
Decision Tree	1	0.00E+00	0.184835086	1.008014757
Multilayer Regressor	0.722945704	0.526359474	0.336298381	0.814678844
SVR	0.750670363	0.499329187	0.452430089	0.739979669
Random forest	0.952543986	0.25904535	0.50947322	0.781943748
M5Prime	0.721565511	0.627468152	0.488697597	0.798331135

En términos de R2 Test, Random Forest parece tener el mejor rendimiento en ambas configuraciones de hiperparámetros, mientras que en términos de RMSE Test, SVR parece ser el mejor en ambas configuraciones. Es importante tener en cuenta que la elección del mejor modelo depende de los resultados combinados de ambas pruebas.

4.4 Resultados de los de los modelos de ML sin sintonizar

Al ejecutar los modelos, en una ronda inicial, se obtienen resultados estándar, ya que se toman tal y como la bibliografía lo establece, dicho de otra manera, no se modifican los hiperparámetros con los que vienen configurados los algoritmos por defecto.

Dichos hiperparámetros no suelen ser los óptimos a la hora de obtener resultados, ya que regularmente no entregan resultados aceptables, entendiéndose esto como aquellos que tienen un alto nivel de error y un grado bajo de precisión

Tales resultados se pueden apreciar en la siguiente Tabla 9, siendo presentados a continuación:

Tabla 4 Resultados obtenidos de los modelos de ML sin sintonizar

Técnica usada	Decision Tree Regressor	Multilayer Regressor	SVR	Random forest	M5 Prime
Hiperparámetros	Max_Depth=none	hidden_layer_sizes=100	C=1.0	N_estimators=100	Max_Depth=2
	Min_simples_split=2	Max_iter=200	Epsilon=0.1	Max_Depth=none	Min_simples_leaf=1
	Min_simples_leaf=1	learning_rate_init=0.001	Kernel=rbf	Min_simples_split=2	Min_simples_Split=2
	Max_features=none	activation='relu'	learning_rate_init=0.001	Min_simples_leaf=1	
	Random_state=1	Random_state=1	Random_state=1	Random_state=1	Random_State=1
R2 train	1.0	-5.698646105	0.6532	0.9525	0.7764
R2 test	0.1848	0.184835086	0.3614	0.5094	0.4804
RMSE train	4.1838e-17	3.077681616	0.7002	0.2590	0.5622
RMSE test	1.0080	5.507298139	0.8921	0.7819	0.8047
Tiempo ejecución	<1s	1s	<1s	<1s	<1s

Aquí se observa el cómo todos los resultados tienen un tiempo de ejecución menor a un segundo, con resultados de R2 y RMSE muy variados, estando lejos de ser buenos en combinación.

4.5 Grid Search

El método tradicional de optimización de hiperparámetros es una búsqueda en cuadrícula, o “Grid Search” (GS) en inglés, que simplemente realiza una búsqueda completa en un subconjunto determinado del espacio de hiperparámetros del algoritmo de entrenamiento.

Debido a que el espacio de parámetros del algoritmo de aprendizaje automático puede incluir espacios con valores reales o ilimitados para algunos parámetros, es posible que necesitemos especificar un límite para aplicar una GS.

La GS adolece de espacios de grandes dimensiones, pero a menudo se puede paralelizar fácilmente, ya que los valores de hiperparámetros con los que trabaja el algoritmo suelen ser independientes entre sí. (Liashchynskyi & Liashchynskyi, 2019)

La GS en Python funciona por medio de la comparación: se toman dos valores de un conjunto pequeño para dos o más parámetros, se evalúan todas las combinaciones posibles y con ellas se forma una cuadrícula de valores.

4.6 Resultados optimizados de los modelos de ML.

Una vez que los modelos son ajustados de acuerdo con los mejores hiperparámetros, y aplicando otros elementos, tales como utilizar estos mismos en la GS o el fijar el random state a 1, se obtienen mejores resultados en cuanto a los indicadores R2 y RMSE. Dichos resultados mejoran en el aspecto de acercarse más al 1 y al cero respectivamente, véase la Tabla 10.

En dicha tabla se hace una comparación de los 5 modelos utilizados, mostrando datos como lo son sus hiperparámetros, los resultados para los datos de entrenamiento y prueba que abarcan tanto el error cuadrático (R2) como Root Mean Square Error (RMSE) y el tiempo de ejecución de cada uno de ellos.

Tabla 5 Resultados con hiperparámetros optimizados de los modelos ML

Técnica usada	Decision Tree	Multilayer Regressor	SVR	Random forest	M5 Prime
Hiperparametros	Max_Depth=None	Activation=relu	C=30	max_Depth= 5	Max_Depth=6
	Min_simples_split=10	Hidden_layers_sizes=10	Degree=5	Min_simples_leaf=1	Min_simples_leaf=4
	Min_simples_leaf=4	learning_rate_init=0.001	Epsilon=0.2	Min_simples_Split=2	Min_simples_Split=12
		Max_iter=1000	Kernel=rbf	N_estimators=1000	use_pruning/Smoothing=false
	Cross validation=5	Cross validation=5	Cross validation=5	Cross validation=5	Cross_Validation=5
	Random_state=1	Random_state=1	Random_state=1	Random_state=1	Random_State=1
R2 train	1.0	0.7229	0.7506	0.9525	0.7215
R2 test	0.1848	0.3362	0.4524	0.5094	0.4886
RMSE train	4.1838e-17	0.5263	0.4993	0.2590	0.6274
RMSE test	1.0080	0.8146	0.7399	0.7819	0.7983
Tiempo de ejecución	<1s	3m 14s	27s	5m 48s	5m 40s

En las figuras 14[a] y 14[b], se muestran los resultados obtenidos al aplicar la técnica de Decision Tree (DT); se tienen las parejas de datos de entrenamiento (R2 Train y RMSE Train) y las parejas de datos de prueba (R2 Test y RMSE Test). Las barras

de color azul muestran los resultados obtenidos utilizando los hiperparámetros estándar del DT y las de color gris muestran los resultados obtenidos utilizando los hiperparámetros sintonizados.

Como se mencionó en el punto 4.3, se utiliza un 80% de los datos para entrenar el modelo (Train) y un 20% para los datos de prueba (Test).

En la Figura 14ª, se observa que la barra azul de R2 Train alcanzan un valor de 1, lo cual indica un ajuste perfecto del modelo al conjunto de datos. Además, los datos RMSE Train muestran un valor de 4.1838 e-17, lo que refleja un error cuadrático prácticamente nulo y confirma, asimismo, un ajuste perfecto del modelo al conjunto de datos.

La barra gris representa los resultados obtenidos con los hiperparámetros ajustados. Se observa una coincidencia exacta con los datos de la barra azul, ambos valores son 1 y 4.1838 e-17, lo cual indica un sobre entrenamiento del modelo a este conjunto de datos específico. Al hacer un cambio en el conjunto de datos, se observó un aumento en el error RMSE y una disminución en el coeficiente de determinación R2, alejándose de su valor ideal de 0 y 1 respectivamente.

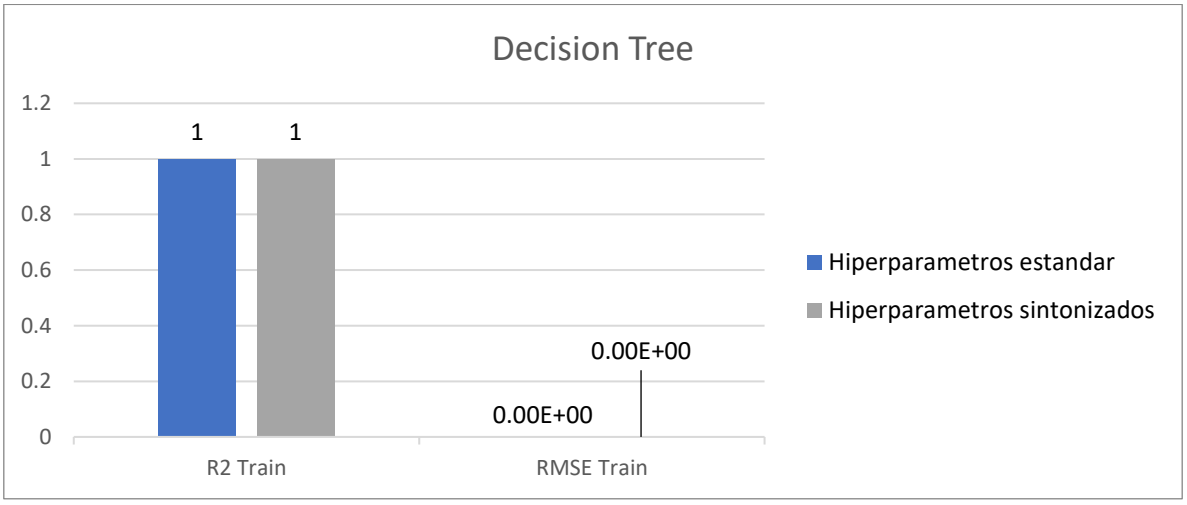

Figura 10a Resultados al aplicar Decision Tree en los datos de entrenamiento (Train)

La figura 14ᵇ, muestra que la barra azul de R2 Test alcanza un valor de 0.1848, lo que indica un mal desempeño del modelo con respecto al conjunto de datos.

Además, los datos RMSE Test muestran un valor de 1.0080, lo que refleja un error cuadrático elevado y confirma, el mal desempeño del modelo con los datos de prueba.

La barra gris representa los resultados obtenidos con los hiperparámetros ajustados. Se observa una coincidencia exacta con los datos de la barra azul, lo cual indica un sobre entrenamiento del modelo a este conjunto de datos específico (Scikit Learn, 2023). Al cambiar el conjunto de datos, se observa un aumento en el error RMSE (1.0080) y una disminución en el coeficiente de determinación R2 (0.1848), alejándose de su valor ideal de 1. Por los datos obtenidos, este modelo fue desechado por el sobre entrenamiento presentado, lo que no lo hace apto para el estudio.

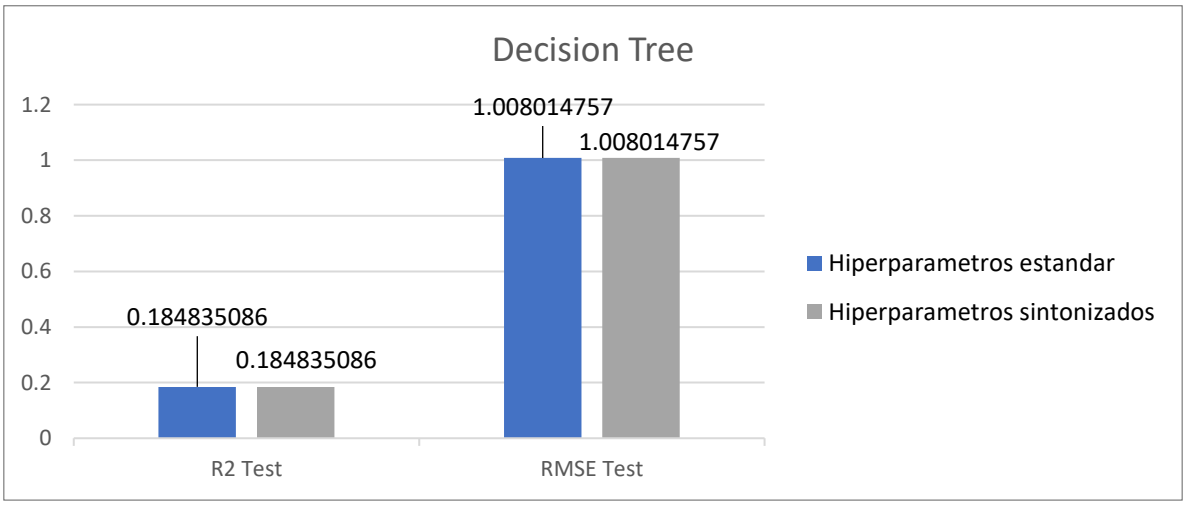

Figura 11b Resultados al aplicar Decision Tree en los datos de prueba (Test)

Las Figuras 15ª y 15ᵇ muestran gráficamente los resultados obtenidos por la técnica de Multilayer Regressor (MR). Al igual que en las gráficas 14, se tienen las parejas de datos de entrenamiento (R2 Train y RMSE Train) y las parejas de datos de prueba (R2 Test y RMSE Test). Las barras de color azul muestran los resultados obtenidos utilizando los hiperparámetros estándar del MR y las de color gris muestran los resultados obtenidos utilizando los hiperparámetros sintonizados.

La Fig. 15ª muestra la ejecución del modelo con los datos Train, se observa que la barra azul de R2 Train alcanzan un valor de -5.6986, lo cual indica un mal desempeño al obtener valores negativos su valor óptimo debe ser cercano o igual a 1. Los datos RMSE Train muestran un valor de 3.0776, lo que refleja un error cuadrático alto y confirma, que no se ajusta de manera óptima al conjunto de datos, porque su valor debe ser lo más cercano a cero.

Las barras grises representan los resultados obtenidos con los hiperparámetros ajustados. Para los resultados de R2 Train, se observa un resultado de 0.7229, cercano a uno, que es un buen resultado, sin embargo, el nivel de error es de 0.5263, que no permite tomar una decisión porque puede dar un buen o mal resultado. Poe los valores obtenidos, es un modelo que no se recomienda aplicar.

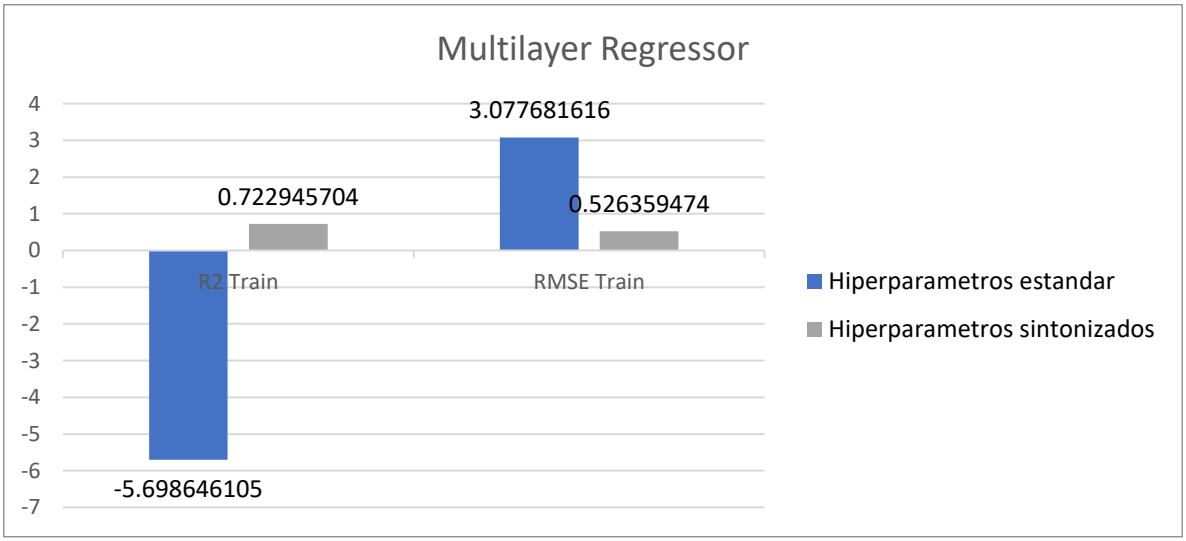

Figura 12a Resultados al aplicar Multilayer Regressor en los datos de entrenamiento (Train)

En la Figura 15ᵇ, se evidencia que la barra azul de R2 Test logra un valor de 0.1848, lo que representa un rendimiento superior al obtenido con los datos de entrenamiento, aunque se encuentra notablemente alejado del valor ideal de 1. Por otro lado, los datos de RMSE Test revelan un valor de 5.5072, indicando un error cuadrático considerablemente elevado en comparación con el ideal (cero), confirmando así un rendimiento deficiente del modelo con los datos de prueba.

Las barras grises representan los resultados obtenidos (0.3362 y 0.8146 respectivamente) mediante la optimización de hiperparámetros. Se nota una mejora significativa en comparación con los resultados obtenidos utilizando los hiperparámetros estándar. Sin embargo, según los datos recopilados, este modelo fue descartado debido a sus resultados: su coeficiente de determinación R2 es bajo, inferior a 1, y el RMSE arroja valores demasiado elevados, lo que lo descarta como adecuado para el estudio.

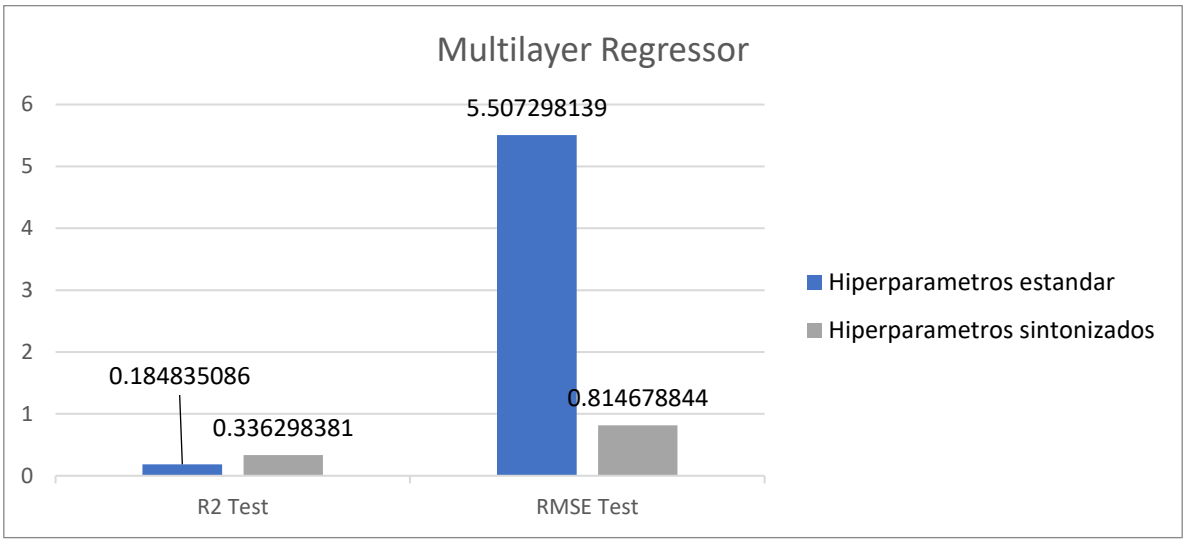

Figura 13b Resultados al aplicar Multilayer Regressor en los datos de prueba (Test)

Las Figuras 16ª y 16ᵇ muestran de una manera gráfica los resultados obtenidos por la técnica Support Vector Regressor (SVR). Al igual que en las gráficas 14 y 15, se tienen las parejas de datos de entrenamiento (R2 Train y RMSE Train) y las parejas de datos de prueba (R2 Test y RMSE Test). Las barras de color azul muestran los resultados obtenidos utilizando los hiperparámetros estándar del SVR y las de color gris muestran los resultados obtenidos utilizando los hiperparámetros sintonizados.

La Fig. 16ª muestra la ejecución del modelo con los datos Train, se observa que la barra azul de R2 Train alcanzan un valor de 0.6532, lo cual indica un desempeño aceptable al obtener valores cercanos a 1. Los datos RMSE Train muestran un valor de 0.7002, lo que refleja un error cuadrático alto y confirma, que no se ajusta de

manera óptima al conjunto de datos, debido a que su valor se aleja bastante del cero.

Las barras grises representan los resultados obtenidos con los hiperparámetros ajustados. Para los resultados de R2 Train, se observa un resultado de 0.7506, cercano a uno, que es un buen resultado, por otro lado, el nivel de error es de 0.4993, que no permite tomar una decisión porque puede dar un buen o mal resultado. Por consecuencia, es un modelo que no se recomienda aplicar a la ligera.

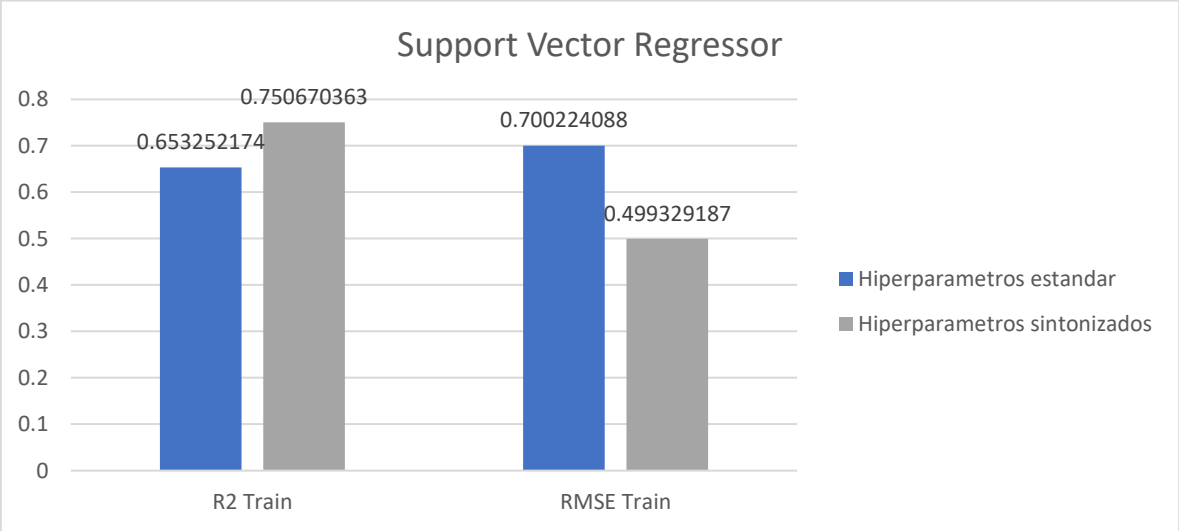

Figura 14a Resultados al aplicar Support Vector Regressor en los datos de entrenamiento (Train)

La Fig. 16 [b] muestra la ejecución del modelo con los datos Test, se observa que la barra azul de R2 Test alcanzan un valor de 0.3614, lo cual indica un desempeño poco aceptable al obtener valores lejanos a 1. Los datos RMSE Test muestran un valor de 0.8921, lo que refleja un error cuadrático alto y confirma, que no se ajusta de manera óptima al conjunto de datos, debido a que su valor es muy cercano a 1.

Las barras grises representan los resultados obtenidos con los hiperparámetros ajustados. Para los resultados de R2 Test, se observa un resultado de 0.4524, resultado que sigue siendo bajo, ergo presenta mejora si se compara con el obtenido de los datos train, por otro lado, el nivel de error es de 0.7399, lo cual indica

un nivel de error elevado. Por consecuencia, es un modelo que no se recomienda aplicar.

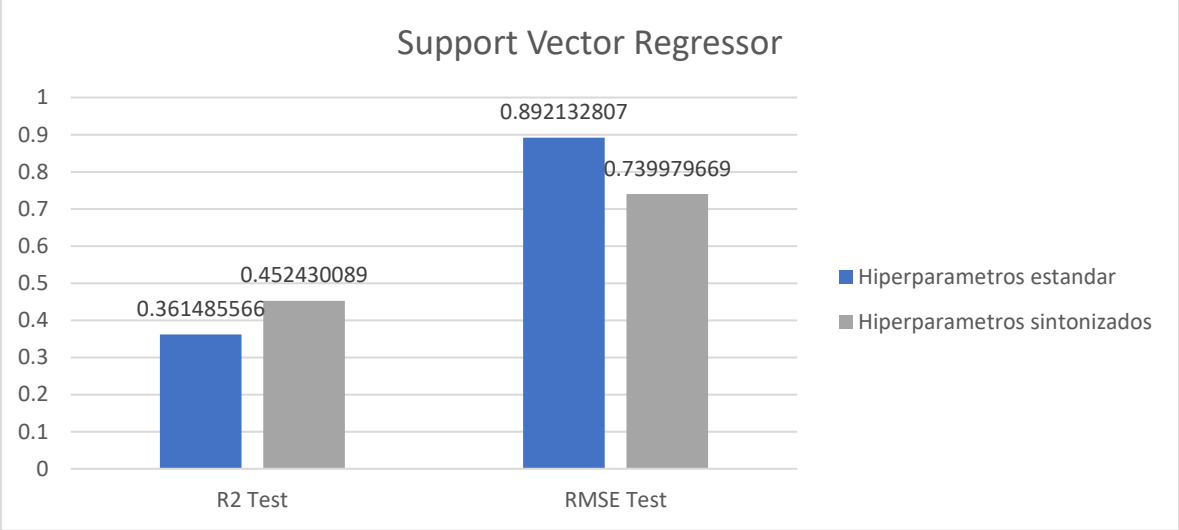

Figura 15b Resultados al aplicar Support Vector Regressor en los datos de prueba (Test)

Las Figuras 17ª y 17ᵇ muestran de una manera gráfica los resultados obtenidos por la técnica Random Forest (RF). De igual modo que en las gráficas vistas anteriormente, se tienen las parejas de datos de entrenamiento (R2 Train y RMSE Train) y las parejas de datos de prueba (R2 Test y RMSE Test). Las barras de color azul muestran los resultados obtenidos utilizando los hiperparámetros estándar del RF y las de color gris muestran los resultados obtenidos utilizando los hiperparámetros sintonizados.

La Fig. 17ª muestra la ejecución del modelo con los datos Train, se observa que la barra azul de R2 Train alcanzan un valor de 0.9525, lo cual indica un desempeño muy bueno al obtener valores cercanos a 1. Los datos RMSE Train muestran un valor de 0.2590, lo que refleja un error cuadrático considerablemente bajo y confirma que el modelo se ajusta de manera óptima al conjunto de datos, debido a la buena respuesta de ambos valores.

Las barras grises representan los resultados obtenidos con los hiperparámetros ajustados. Para los resultados de R2 Train, se observa un resultado de 0.9525, cercano a uno, que es un muy buen resultado, así mismo, el nivel de error es de 0.2590, lo cual ayuda a tomar una decisión debido a sus buenos resultados. Por consecuencia, es el modelo escogido para su implementación.

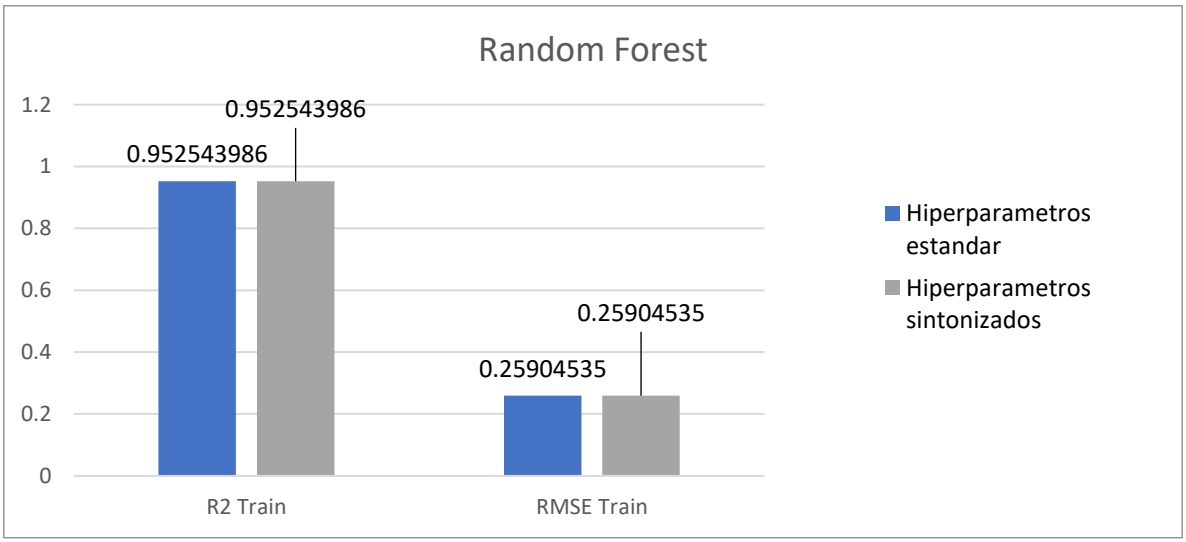

Figura 17a Resultados al aplicar Random Forest en los datos de entrenamiento (Train)

La Fig. 17 [b] muestra la ejecución del modelo con los datos Test, podemos observar que la barra azul de R2 Test alcanza un valor de 0.5094, lo cual indica un desempeño aceptable, pero dejando aún que desear. Los datos RMSE Test muestran un valor de 0.7819, lo que refleja un error cuadrático alto en contraste con su nivel de R2, debido a que su valor es muy cercano a 1.

Las barras grises representan los resultados obtenidos con los hiperparámetros ajustados. Para los resultados de R2 Test, se observa un resultado de 0.5094, resultado que sigue siendo bajo, ergo presenta mejora si se compara con el obtenido de los datos Test, por otro lado, el nivel de error es de 0.7891, lo cual indica un nivel de error elevado. Sin embargo, al haber obtenido este modelo los mejores resultados del estudio sin haber caído en sobreajuste, es el modelo que se recomienda utilizar.

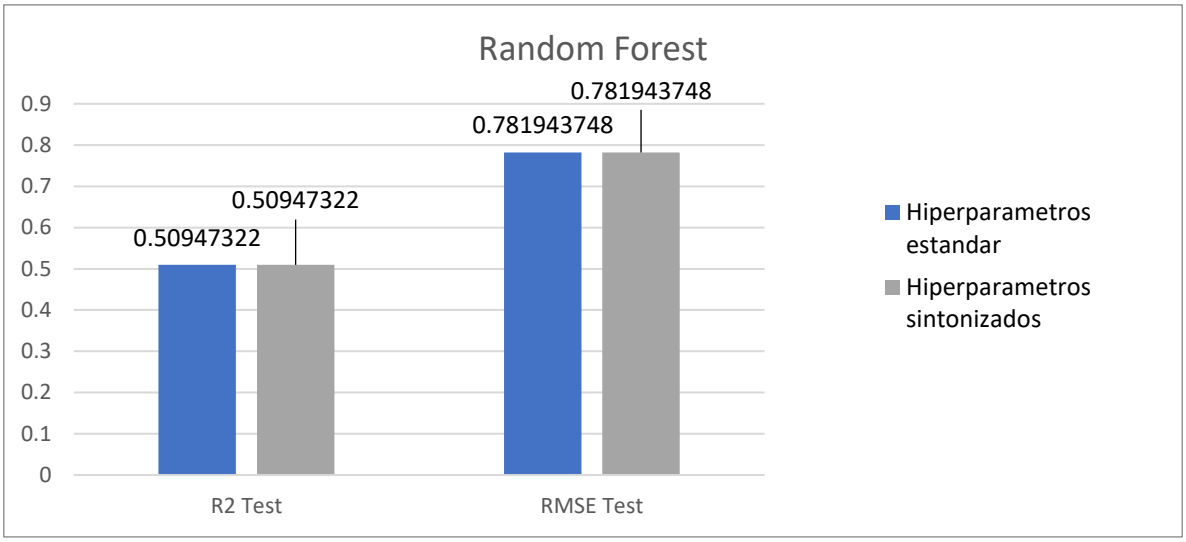

Figura 17b Resultados al aplicar Random Forest en los datos de entrenamiento (Test)

Las Figuras 18ª y 18ᵇ muestran de una manera gráfica los resultados obtenidos por la técnica M5 Prime (M5P). Del mismo modo que en las gráficas vistas anteriormente, se tienen las parejas de datos de entrenamiento (R2 Train y RMSE Train) y las parejas de datos de prueba (R2 Test y RMSE Test). Las barras de color azul muestran los resultados obtenidos utilizando los hiperparámetros estándar del algoritmo y las de color gris muestran los resultados obtenidos utilizando los hiperparámetros sintonizados.

La Fig. 18ª muestra la ejecución del modelo con los datos de entrenamiento (Train), se observa que la barra azul de R2 Train alcanzan un valor de 0.7764, lo cual indica un desempeño bueno al obtener valores cercanos a 1. Los datos RMSE Train muestran un valor de 0.5622, lo que refleja un error cuadrático medio, lo cual no permite tomar una decisión porque puede dar un buen o mal resultado. Por consecuencia, es un modelo que no se recomienda aplicar.

Las barras grises representan los resultados obtenidos con los hiperparámetros ajustados. Para los resultados de R2 Train, se observa un resultado de 0.7215, cercano a uno, que es un muy resultado, ergo, el nivel de error es de 0.6274, lo cual

es un demerito de esta técnica. Por consecuencia, no se escogió como modelo, pero queda a reserva su uso.

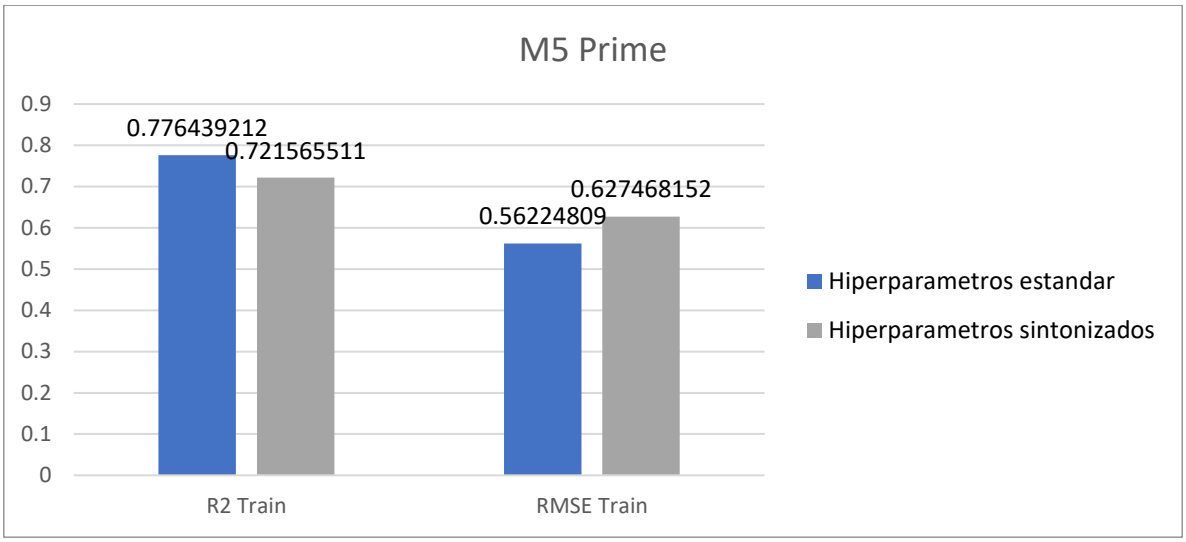

Figura 18ª Resultados al aplicar M5 Prime en los datos de entrenamiento (Train)

La Fig. 18 [b] muestra la ejecución del modelo con los datos Test, podemos observar que la barra azul de R2 Test alcanza un valor de 0.4804, lo cual indica un desempeño aceptable, pero dejando aún que desear. Los datos RMSE Test muestran un valor de 0.8047, lo que refleja un error cuadrático alto en contraste con su nivel de R2, debido a que su valor es muy cercano a 1.

Las barras grises representan los resultados obtenidos con los hiperparámetros ajustados. Para los resultados de R2 Test, se observa un resultado de 0.4886, resultado que sigue siendo bajo, ergo presenta mejora si se compara con el obtenido de los datos Test, por otro lado, el nivel de error es de 0.78983, lo cual indica un nivel de error elevado. Este modelo no se recomienda debido a su nivel de error elevado y su nivel de R2 de apenas casi 0.5.

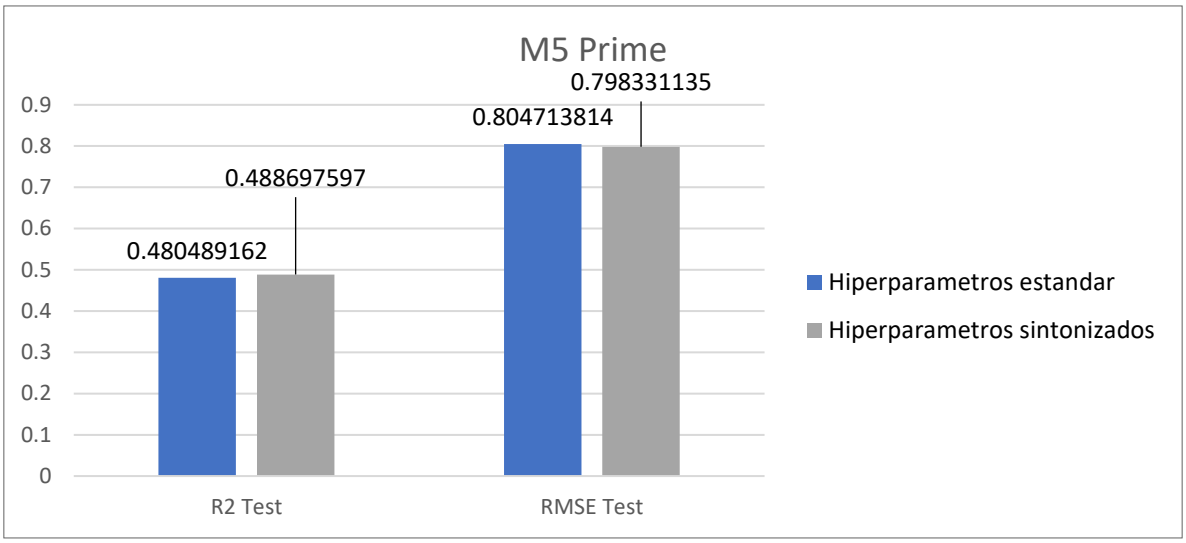

Figura 16 M5 Prime

A continuación, en las figuras 19 a la 23, se presentan los gráficos de precisión de los modelos, empleando los parámetros óptimos obtenidos mediante Grid Search (GS), tanto para el conjunto de datos de entrenamiento (Train) como para el conjunto de datos de prueba (Test).

Una gráfica de precisión se refiere a un tipo de representación visual que muestra cómo varía la precisión de un modelo en función de diferentes umbrales de clasificación. La precisión en este contexto se refiere a la proporción de instancias correctamente clasificadas como positivas con respecto al total de instancias clasificadas como positivas. Muestra cómo la precisión del modelo cambia a medida que se ajusta el umbral de decisión, permitiendo evaluar y ajustar el rendimiento del modelo (Raschka & Mirjalili, 2017).

En la figura 19 podemos observar el comportamiento de los datos de entrenamiento (Train) y datos de prueba (test) utilizando los hiperparámetros ajustados para el modelo decisión tree. Se puede observar de una manera gráfica la predicción del modelo siendo comparada con la de los datos obtenidos.

En dichas gráficas se comparan en primer lugar las predicciones obtenidas usando los datos de entrenamiento contra los valores reales, como segundo recuadro tenemos un proceso similar, pero utilizando los datos de prueba.

De igual manera, en el tercer recuadro encontramos una gráfica de valores residuales. Dicha gráfica muestra los datos residuales a la hora de correr el modelo con ambos datos entrenamiento (en color azul y prueba (en color verde). En esta se puede ver que entre más alineados con el cero se encuentren dichos residuales mayor es la efectividad de este.

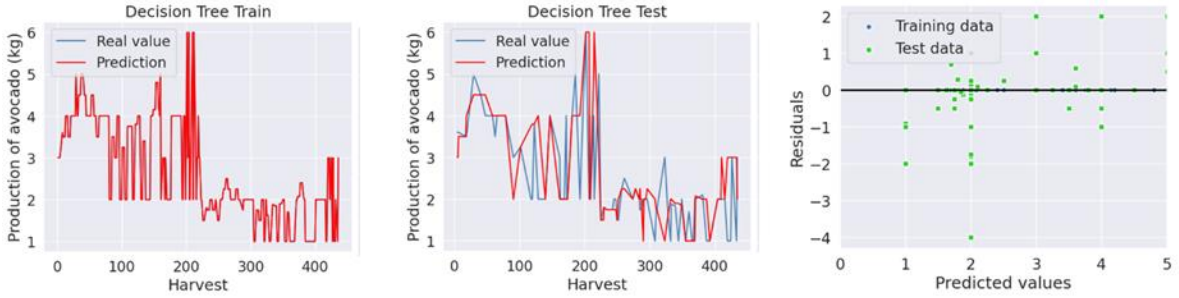

Figura 1917 Gráficos de precisión de los modelos Decision Tree Train y Test

En la figura 20 observamos el cómo se comportan de los datos de entrenamiento (Train) y datos de prueba (test) siendo usados los hiperparámetros ajustados para el modelo Multilayer Regressor. Observamos de una manera gráfica la predicción del modelo siendo comparada con la de los datos obtenidos.

En dichas gráficas se comparan las predicciones obtenidas usando los datos de entrenamiento contra los valores reales, y en segundo recuadro tenemos un proceso similar, pero utilizando los datos de prueba.

Al igual que anteriormente, tenemos una gráfica de valores residuales. Dicha gráfica muestra los datos residuales al momento de correr el modelo con ambos conjuntos

de datos (entrenamiento y prueba). Se puede ver que entre más alineados con el cero se encuentren dichos residuales mayor es la efectividad de este.

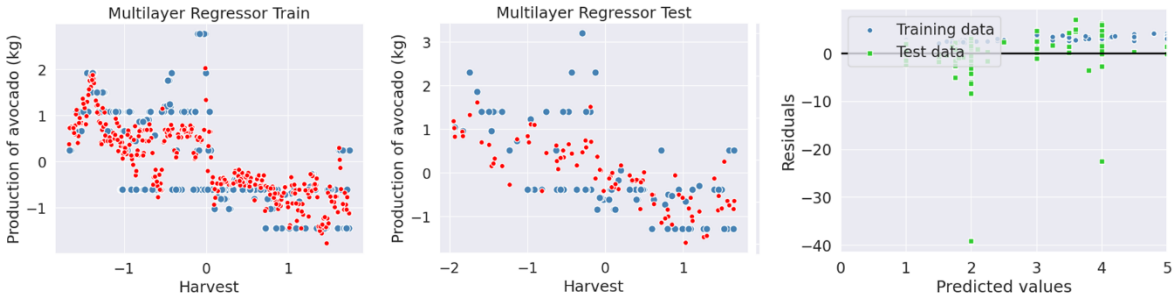

Figura 180 Gráficos de precisión de los modelos Multilayer Regressor Train y Test

En la Figura 21, se presenta el comportamiento de los datos de entrenamiento (Train) y de prueba (Test) al emplear los hiperparámetros ajustados para el modelo Support Vector Regressor. Esta representación gráfica permite comparar la predicción del modelo con los valores reales obtenidos.

Las gráficas muestran, en primer lugar, la comparación de las predicciones utilizando los datos de entrenamiento frente a los valores reales. En un segundo recuadro, se realiza un proceso similar, pero utilizando los datos de prueba.

Además, se incluye una gráfica de valores residuales que ilustra la diferencia entre los valores predichos y los reales al ejecutar el modelo tanto con los datos de entrenamiento (en color azul) como con los de prueba (en color verde). La efectividad del modelo se refleja en la alineación de estos residuales con el cero; cuanto más cercanos estén, mayor será la precisión del modelo.

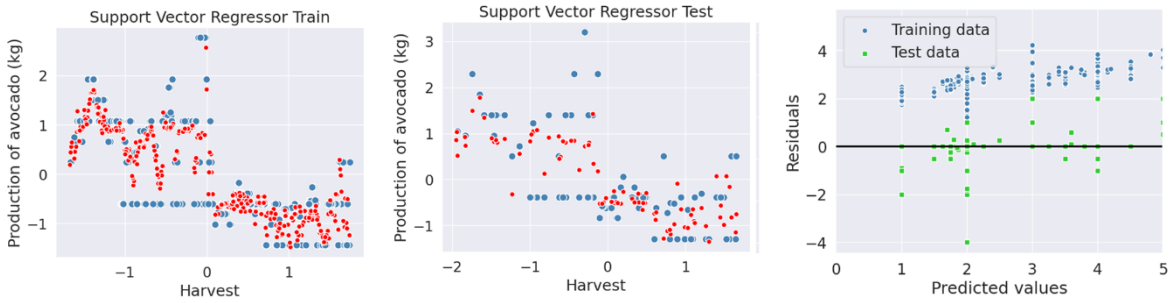

Figura 191 Gráficos de precisión de los modelos Support Vector Regressor Train y Test

En la Figura 22, se analiza el comportamiento de los datos de entrenamiento (Train) y de prueba (Test) al emplear los hiperparámetros ajustados para el modelo Random Forest. La representación gráfica permite comparar la predicción del modelo con los valores reales obtenidos.

En estas gráficas, se compara inicialmente las predicciones obtenidas utilizando los datos de entrenamiento con los valores reales, y posteriormente, se presenta un proceso similar en un segundo recuadro, utilizando los datos de prueba.

Asimismo, se incluye una gráfica de valores residuales que ilustra la diferencia entre los valores predichos y los reales al ejecutar el modelo con ambos conjuntos de datos (entrenamiento y prueba). La efectividad del modelo se refleja en la alineación de estos residuales con el cero; cuanto más cercanos estén, mayor será la precisión del modelo.

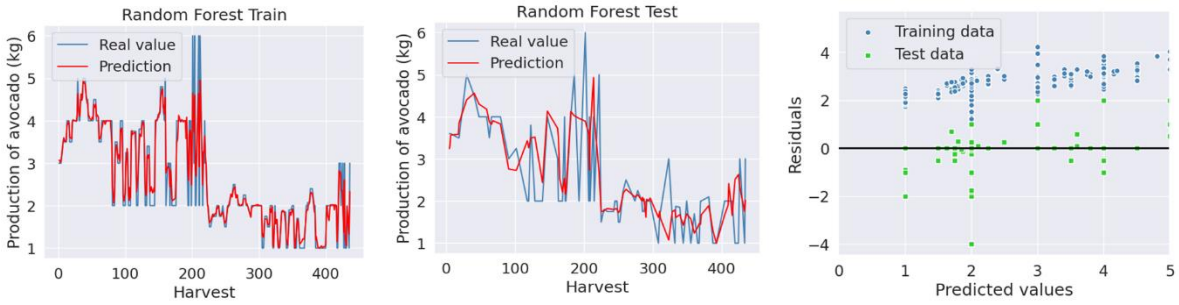

Figura 202 Gráficos de precisión de los modelos Random Forest Train y Test

Finalmente, en la Figura 23, se examina el comportamiento de los conjuntos de datos de entrenamiento (Train) y prueba (Test) al aplicar los hiperparámetros ajustados para el modelo M5 Prime. La representación gráfica facilita la comparación entre las predicciones del modelo y los valores reales obtenidos.

En estas gráficas, se inicia comparando las predicciones obtenidas con los datos de entrenamiento respecto a los valores reales. Luego, se presenta un proceso similar en un segundo recuadro, pero esta vez utilizando los datos de prueba.

Además, se incorpora una gráfica de valores residuales que visualiza la discrepancia entre los valores predichos y los reales al ejecutar el modelo con ambos conjuntos de datos (entrenamiento y prueba). La eficacia del modelo se refleja en la alineación de estos residuales con el cero; a mayor proximidad, mayor es la precisión del modelo.

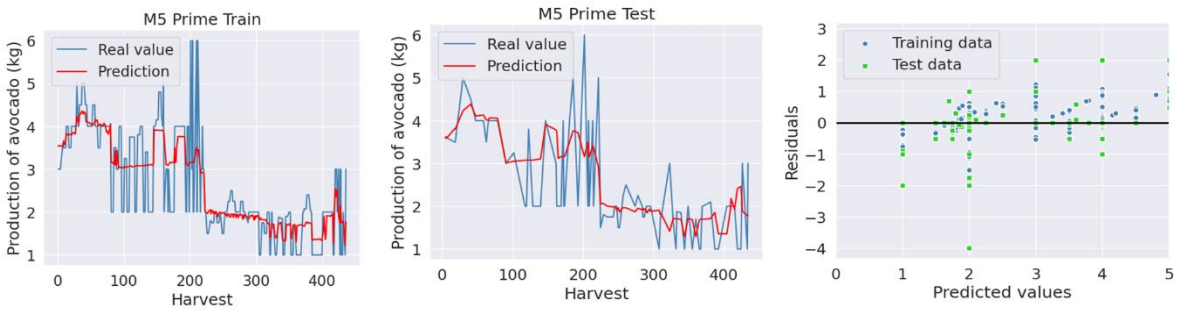

Figura 23 Gráficos de precisión de los modelos M5 Prime Train y Test

Con las gráficas vistas anteriormente, se puede afirmar que el mejor modelo ha sido Random Forest (Bosques aleatorios), y que ha presentado los mejores números en cuando a R2 y RMSE, además de que en las gráficas de precisión y de elementos residuales ha mostrado un comportamiento sobresaliente, siendo por estos motivos el modelo elegido.

4 Conclusiones y trabajos futuros

Finalmente podemos llegar a los siguientes resultados y conclusiones:

- Se ha demostrado que el modelo de Random Forest Regressor sobresale como la elección óptima en este estudio, al exhibir un alto índice R^2 y un bajo error cuadrático medio (RMSE) tanto antes como después de la sintonización de hiperparámetros. Este modelo ha demostrado un rendimiento consistente tanto en los datos de entrenamiento como en los datos de prueba, lo que respalda su idoneidad para predecir con precisión los valores de interés en nuestro conjunto de datos.
- Los resultados de este estudio indican que la técnica de búsqueda en la cuadrícula (Grid Search) es efectiva en la optimización de los hiperparámetros del modelo. Sin embargo, se recomienda que futuras investigaciones consideren un conjunto de datos más amplio para mejorar aún más el rendimiento y obtener resultados óptimos. La inclusión de una mayor cantidad de lecturas podría proporcionar una visión más completa y precisa de las relaciones entre las variables, lo que contribuiría a una mejor optimización de los hiperparámetros.
- Se ha creado una base de datos a partir de los datos recopilados en la estación de monitoreo meteorológico y nutricional de la UPA, que ha sido procesada y normalizada utilizando la metodología de Machine Learning propuesta. Este proceso de limpieza y normalización es esencial para garantizar la calidad de los datos y permitir un análisis preciso.
- Nuestros hallazgos indican de manera concluyente que el algoritmo Random Forest es la elección preferida para este conjunto de datos en particular. Tras la sintonización de los hiperparámetros, ha demostrado la mayor precisión tanto en los datos de entrenamiento como en los datos de prueba. Esto subraya su idoneidad para la predicción de valores nutricionales basados en las condiciones meteorológicas y otros factores.

- Se cumplieron los objetivos establecidos al inicio del estudio, ya que se probaron distintos modelos de ML,
- En el futuro, se abre la oportunidad de continuar expandiendo este estudio mediante la adquisición de más datos. Además, se podría considerar una actualización de la estación de monitoreo meteorológico y nutricional (p.e, incluir lecturas de nutrientes adicionales, tales como el zinc o el hierro, lecturas de P.H y velocidad del viento), lo que permitiría la obtención de información aún más relevante y precisa para futuros análisis.
- Además, este modelo de obtención de datos y predicción basado en Machine Learning podría replicarse en otras áreas de la zona norte del Estado de Morelos. Esto podría contribuir significativamente a una comprensión más completa de las relaciones entre las condiciones meteorológicas y los valores nutricionales, lo que podría tener un impacto positivo en la toma de decisiones relacionadas con la agricultura y la nutrición en la región.

BIBLIOGRAFÍA

Ahmed, N., Debashis, D., & Hussain, I. (Diciembre de 2018). Internet of Things (IoT) for Smart Precision Agriculture and Farming in Rural Areas. *IEEE Internet of Things Journal, 5*(6), 4890-4899.

Alphabet Inc. (21 de Marzo de 2023). *Google Earth*. Recuperado el 21 de Marzo de 2023, de Google Earth: https://earth.google.com/web/@19.00828417,-99.26930851,2268.01030679a,311.26090518d,35y,0h,0t,0r/data=OgMKATA

Bacco, M., Barsocchi, P., Ferro, E., Gotta, A., & Ruggeri, M. (15 de Octubre de 2019). The Digitisation of Agriculture: a survey of Research Activities on Smart Farming. *Array*. doi:https://doi.org/10.1016/j.array.2019.100009

Baez-Gonzalez, A. D., Kiniry, J. R., & Williams, J. (2016). Agricultural Land Management Alternatives with Numerical Assessment Criteria Model (ALMANAC) with Mexican Interface. *INIFAP, 1*(44), 18.

Balducci, F., Impedovo, D., & Pirlo, G. (1 de Septiembre de 2018). Machine learning applications on agricultural datasets for smart farm enhancement. *Machines*, 6-38.

Blei, D. M., Ng, A. Y.-T., & Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 993-1022.

Boters Pitarch, J. (2021). *Bosques aleatorios supervisados y no supervisados. Aplicación al análisis de desigualdades económicas en el mundo.* Universitat Jaume I. Castellón de la Plana: Universitat Jaume I.

Cohen, J. E. (Febrero de 2001). World population in 2050: Assessing the projections. *Federal Reserve Bank of Boston*, 83-113.

Diaz, I., Mazza, S. M., Combarro, E. F., Giménez, L. I., & Gaiad, J. E. (Junio de 2017). Machine learning applied to the prediction of citrus production. *Spanish Journal of Agricultural Research, 15*(2), 7.

Doshi, J., Patel, T., & Bharti, S. (November de 2019). Smart Farming using IoT, a solution for optimally monitoring farming conditions. *Procedia Computer Science, 160*, 746-751. doi:https://doi.org/10.1016/j.procs.2019.11.016

El Naqa, I., & Murphy, M. (2015). What Is Machine Learning? . *Machine learning in radiation oncology*, 3-11.

El-Bendary, N., El Hariri, E., Hassanien, A., & Badr, A. (Marzo de 2015). Using machine learning techniques for evaluating tomato ripeness. *Expert Systems with Applications, 42*(4), 1892-1905.

Espinosa, A., Ponte, D., Gibeaux, S., & González, C. (Junio de 2021). Estudio de Sistemas IoT Aplicados a la Agricultura Inteligente. *Plus Economía, 9*, 33-42.

Farooq, M., Riaz, S., Abid, A., Abid, K., & Naeem, M. (2019). A Survey on the Role of IoT in Agriculture for the Implementation of Smart Farming. *Ieee Access*, 156237-156271.

García Cañón, H. S. (Mayo de 2019). Implementación de técnicas de machine learning para mla predicción de variables meteorológicas y del suelo que afectan a la agricultura. Bogotá, Colombia: Universidad de los Andes.

Gomes Alves, R., Souza, G., Filev Maia, R., Tran, A., Kamienski, C., Soininen, J.-P., . . . Lima, F. (12 de Marzo de 2019). A digital twin for smart farming. *IEEE Xplore*. Recuperado el 06 de Junio de 2023

Guerrero Cano, M., Luque Sendra, A., Lama Ruiz, J. R., & Córdoba Roldán, A. (2019). PREDICTIVE MAINTENANCE USING MACHINE LEARNING TECHNIQUES. *23rd International Congress on Project Management and Engineering* (págs. 721-730). Málaga: AEIPRO.

Hillar, G. C. (2017). Understanding convenient scenarios for the MQTT protocol. En *MQTT Essentials - A Lightweight IoT Protocol* (págs. 9, 10). Birmingham, West Midlands, United Kingdom: Packt.

Icarte Ahumada, G. A. (7 de Marzo de 2016). Aplicaciones de inteligencia artificial en procesos de cadenas de suministros: una revisión sistemática. *Ingeniare. Revista chilena de ingeniería, 4*(24), 663-679.

Kiggins, J. (10 de Octubre de 2020). *Kaggle*. Recuperado el 10 de mayo de 2022, de Kaggle: https://www.kaggle.com/datasets/neuromusic/avocado-prices

Lázaro Enguita, P. (2018). *Machine Learning en la industria del automóvil.* Escuela Politécnica Superior. Alcalá: Universidad de Alcalá .

Legorreta Anguiano, D. (7 de Abril de 2015). *DLegorreta*. Obtenido de https://dlegorreta.wordpress.com/2015/04/07/maquina-de-soporte-vectorial-svm-sopport-vector-machine/

Liakos, K. G., Busato, P., Moshou, D., Pearson, S., & Bochtis, D. (14 de August de 2018). Machine Learning in agriculture: a review. *Sensors*, 2674.

Liashchynskyi, P., & Liashchynskyi, P. (Diciembre de 2019). Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS. *ResearchGate*.

López Herraiz, J. (17 de Diciembre de 2021). *LibreTexts Español*. Recuperado el 2022, de LibreTexts Español: https://espanol.libretexts.org/Matematicas/Las_matematicas_de_la_inteligencia_artificial/06:_Redes_Neuronales/6.3:_Perceptron_Multicapa

Maisueche Cuadrado, A. (2019). *Utilización del Machine Learning en la industria 4.0.* Universidad de Valladolid, Escuela de Ingenierías Industriales. Valladolid: Universidad de Valladolid.

Marié, S. (12 de Mayo de 2022). M5 Prime regression trees in python, compliant with scikit-learn. Recuperado el 10 de Junio de 2022, de https://www.youtube.com/watch?v=KkEVD3JncdI

Martínez Heras, J. (2 de Octubre de 2020). *IArtificial*. Recuperado el 19 de Junio de 2022, de https://www.iartificial.net/regresion-lineal-con-ejemplos-en-python/

miRiego. (10 de Octubre de 2019). *Inteligencia Artificial en la Agricultura*. Obtenido de miRiego: https://miriego-blog.com/2019/10/10/inteligencia-artificial-en-la-agricultura/

Morales García, J. L., Rodríguez Guzmán, M. d., Azpíroz Rivero, H. S., & Pedraza Santos, M. E. (2009). Modelo para la estimación del área del fruto en la evaluación de la antracnosis en aguacate. *Revista UDO Agrícola*(9), 421 - 424.

Moreno-Bernal, P., Arizmendi-Peralta, P. P., Hernández-Aguilar, J. A., Nesmachnow, S., Peralta-Abarca, J. C., & Velásquez-Aguilar, J. G. (28 de Noviembre de 2022). IoT platform for monitoring nutritional and weather conditions of avocado production. *Ibero-American Congress of Smart Cities*, 95-109.

Nosratabadi, S., Ardabili, S., Lakner, Z., Mako, C., & Mosavi, A. (Noviembre de 2021). Prediction of Food Production Using Machine Learning. *Agriculture*, 408 - 418.

Nyalala, I., Okinda, C., Nyalala, L., Makange, N., Chao, Q., Chao, L., . . . Chen, K. (14 de Julio de 2019). Tomato volume and mass estimation using computer vision and machine learning algorithms: Cherry tomato model. *Journal of Food Engineering*(263), 288-298. Recuperado el 16 de Junio de 2022

Ojeda-Beltrán, A. (Junio de 2022). Plataformas tecnologicas en la Agricultura 4.0: Una mirada al desarrollo en Colombia. *J. Comput. Electron. Sci: Theory Appl., 3*(1), 9-18.

Ramírez Gómez, C. A. (Julio de 2020). Application of Machine Learning in precision agriculture. *Cintex, 25*(2), 14-27. Recuperado el Enero de 2023

Raschka, S., & Mirjalili, V. (2017). Python Machine Learning. En *Python Machine Learning* (2 ed., págs. 3285-331). Birmingham, Tierras Medias Occidentales, Inglaterra: Packt Publishing Ltd. Recuperado el 29 de Junio de 2023

Rincón Patiño, J., Lasso, E., & Corrales, J. C. (29 de September de 2019). Estimating Avocado Sales Using Machine Learning Algorithms and Weather Data. *Sustainability*, 3498.

Sánchez Romero, P. A. (2021). *Aplicación de algoritmos Machine Learning para un vehículo autónomo.* Universidad Politécnica de Cartagena, Escuela Técnica Superior de Ingeniería de Telecomunicación. Cartagena de Indias: Universidad Politécnica de Cartagena.

Scikit Learn. (29 de Junio de 2023). *scikit-learn*. Recuperado el 29 de Junio de 2023, de scikit-learn: https://scikit-learn.org/stable/modules/model_evaluation.html#scoring

Serikul, P., Nakpong, N., & Nakjuatong, N. (17 de Enero de 2019). Smart Farm Monitoring via the Blynk IoT Platform : Case Study: Humidity Monitoring and Data Recording. *IEEE Xplore*. doi:10.1109/ICTKE.2018.8612441

Sharma, A., Jain, A., Gupta, P., & Chowdary, V. (11 de Enero de 2021). Machine Learning Applications for Precision. *IEEE Access*, 4843-4873.

Trilles, S., González-Pérez, A., & Huerta, J. (2020). An IoT Platform Based on Microservices and Serverless Paradigms for Smart Farming Purposes. *Sensors*.

Anexos

Código utilizado

```
#pd es el alias comun de pandas
import pandas as pd
import numpy as np

bd_huerta =
pd.read_csv('https://thedaytobringdowngods.000webhostapp.com/BaseGranular3.csv')

def lin_regplotml(X, y, model):
    plt.scatter(X['Lectura'], y, c='steelblue', edgecolor='white', s=70)
    plt.scatter(X['Lectura'], model.predict(X), c='red', edgecolor='white', s=35)

    return

def lin_regplotmlALTER(X, y, model):
    p = model.predict(X)
    Z = X.filter(['Lectura'])
    Z["y"] = y
    Z["predict"]= p
    Z = Z.sort_values("Lectura")
    plt.plot(Z['Lectura'], Z["y"],c='steelblue',  label='Real value' )
    plt.plot(Z['Lectura'], Z["predict"], c= 'red', label = 'Prediction')
    plt.legend(loc='upper left')
    return

def lin_regplotml2(X, y, model):
    plt.scatter(X[:, 0], y, c='steelblue', edgecolor='white', s=70)  # Acceder a la primera columna de X
```

```
   plt.scatter(X[:, 0], model.predict(X), c='red', edgecolor='white', s=35)  # Acceder a la primera columna de X
   return


import seaborn as sns
cols=['Ts', 'Ta','Hs','HsPercent','LumenRaw','Lux','Ha','NS','PS','KS','Kilos','Unidades','Lectura']
cm = np.corrcoef(bd_huerta[cols].values.T)
sns.set(font_scale=1.5)
hm= sns.heatmap(cm, cbar=True, annot=True, square= True, fmt='.2f', annot_kws={'size':8}, yticklabels=cols, xticklabels=cols)


bd_huerta.info()


from sklearn.linear_model import LogisticRegression, LinearRegression, Perceptron
from sklearn.tree import DecisionTreeRegressor
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC, LinearSVC #Verificar si hay regresor
from sklearn import linear_model
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
from sklearn.neural_network import MLPRegressor
from sklearn.datasets import make_regression
from sklearn.model_selection import train_test_split
from sklearn.feature_selection import RFECV
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_score, recall_score
from sklearn.model_selection import cross_val_predict
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import cross_val_score
import matplotlib.pyplot as plt
```

```
import warnings #Elimina los warnings
warnings.filterwarnings("ignore")

from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score,classification_report
from sklearn.model_selection import cross_validate
from sklearn.metrics import make_scorer
from sklearn.model_selection import cross_val_score
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
import numpy as np
from sklearn.preprocessing import MinMaxScaler

X = bd_huerta.iloc[:,0:11]# Faltaba KS
y = bd_huerta.iloc[:,11]

print (X.shape)
print (y.shape)

X_train, X_test, y_train, y_test = train_test_split(X, y,
                                    test_size=0.2,
                                    random_state=1)

print (X)
print (y)

print(X_train, y_train)
```

```
#ÁRBOLES DE DECISIÓN REGRESORES (DTR)#

from sklearn.metrics import r2_score, mean_squared_error
from math import sqrt
from sklearn.tree import DecisionTreeRegressor

decision_tree_regressor = DecisionTreeRegressor(random_state=1)
decision_tree_regressor.fit(X_train, y_train)

y_pred = decision_tree_regressor.predict(X_train)
r2 = r2_score(y_train, y_pred)
rmse = sqrt(mean_squared_error(y_train, y_pred))

y_predTest = decision_tree_regressor.predict(X_test)
r2Test = r2_score(y_test, y_predTest)
rmseTest = sqrt(mean_squared_error(y_test, y_predTest))


print("R^2 train:", r2)
print("RMSE train:", rmse)

print("R^2 test:", r2Test)
print("RMSE test:", rmseTest)

lin_regplotmlALTER(X_train, y_train,decision_tree_regressor)
plt.xlabel('Harvest')
plt.ylabel('Production of avocado (kg)')
plt.show()
```

```
lin_regplotmlALTER(X_test, y_test,decision_tree_regressor)

plt.xlabel('Harvest')

plt.ylabel('Production of avocado (kg)')

plt.show()


plt.scatter(y_train, y_train - y_pred, c='steelblue', marker='o', edgecolor='white', label='Training data')

plt.scatter(y_test, y_test - y_predTest, c='limegreen', marker='s', edgecolor='white', label='Test data')

plt.xlabel('Predicted values')

plt.ylabel('Residuals')

plt.legend(loc='upper left')

plt.hlines(y=0, xmin=-0, xmax=5, color='black', lw=2)

plt.xlim([0, 5])

plt.show()



from sklearn.tree import DecisionTreeRegressor

from sklearn.model_selection import GridSearchCV

from sklearn.metrics import make_scorer, r2_score, mean_squared_error

from math import sqrt


param_grid = {

   'random_state': [1]

}


decision_tree = DecisionTreeRegressor()
```

```
scoring = {'R2': make_scorer(r2_score), 'RMSE': make_scorer(mean_squared_error,
squared=False)}


#Datos train

grid_search = GridSearchCV(decision_tree, param_grid, cv=5, scoring=scoring, refit='R2')

grid_search.fit(X_train, y_train)


#Best params Train

best_params = grid_search.best_params_


best_decision_tree = DecisionTreeRegressor(**best_params)

best_decision_tree.fit(X_train, y_train)


y_pred = best_decision_tree.predict(X_train)

y_pred_Test = best_decision_tree.predict(X_test)



r2 = r2_score(y_train, y_pred)

r2_Test = r2_score(y_test, y_pred_Test)


rmse_Test = sqrt(mean_squared_error(y_test, y_pred_Test))

rmse = sqrt(mean_squared_error(y_train, y_pred))


print("Mejores hiperparámetros:", best_params)

print("Mejor puntaje (R2):", r2)

print("RMSE:", rmse)


print("Mejor puntaje (R2) Test:", r2_Test)

print("RMSE Test:", rmse_Test)
```

```
lin_regplotmlALTER(X_train, y_train, best_decision_tree)

plt.xlabel('Harvest')

plt.ylabel('Production of avocado (kg)')

plt.title('Decision Tree Test')

plt.show()


lin_regplotmlALTER(X_test, y_test,best_decision_tree)

plt.xlabel('Harvest')

plt.ylabel('Production of avocado (kg)')

plt.title('Decision Tree Test')

plt.show()


plt.scatter(y_train, y_train - y_pred, c='steelblue', marker='o', edgecolor='white', label='Training
data')

plt.scatter(y_test, y_test - y_pred_Test, c='limegreen', marker='s', edgecolor='white', label='Test
data')

plt.xlabel('Predicted values')

plt.ylabel('Residuals')

plt.legend(loc='upper left')

plt.hlines(y=0, xmin=-0, xmax=5, color='black', lw=2)

plt.xlim([0, 5])

plt.show()


#Multilayer Regressor#


from sklearn.metrics import r2_score, mean_squared_error

from math import sqrt

from sklearn.tree import DecisionTreeRegressor
```

```
perceptron = MLPRegressor(random_state=1)
perceptron.fit(X_train, y_train)

y_pred = perceptron.predict(X_train)
y_pred_Test = perceptron.predict(X_test)

r2 = r2_score(y_train, y_pred)
rmse = sqrt(mean_squared_error(y_train, y_pred))
r2_test = r2_score(y_test, y_pred_Test)
rmse_Test = sqrt(mean_squared_error(y_test, y_pred_Test))

print("R^2:", r2)
print("RMSE:", rmse)

print("R^2 Test:", r2_Test)
print("RMSE Test:", rmse_Test)

lin_regplotmlALTER(X_train, y_train,perceptron)
plt.xlabel('Harvest')
plt.ylabel('Production of avocado (kg)')
plt.title('Multilayer Regressor Train')
plt.show()

lin_regplotmlALTER(X_test, y_test,perceptron)
plt.xlabel('Harvest')
plt.ylabel('Production of avocado (kg)')
plt.show()
```

```
plt.scatter(y_train, y_train - y_pred, c='steelblue', marker='o', edgecolor='white', label='Training data')

plt.scatter(y_test, y_test - y_pred_Test, c='limegreen', marker='s', edgecolor='white', label='Test data')

plt.xlabel('Predicted values')

plt.ylabel('Residuals')

plt.legend(loc='upper left')

plt.hlines(y=0, xmin=-0, xmax=5, color='black', lw=2)

plt.xlim([0, 5])

plt.show()



from sklearn.preprocessing import StandardScaler



# Normalización de los datos de entrenamiento

scaled_x = StandardScaler()

scaled_y = StandardScaler()


X_train_scaled = scaled_x.fit_transform(X_train)

y_train_scaled = scaled_x.fit_transform(y_train[:,np.newaxis]).flatten()


X_test_scaled = scaled_x.fit_transform(X_test)

y_test_scaled = scaled_x.fit_transform(y_test[:,np.newaxis]).flatten()


# Definir los hiperparámetros y sus valores posibles

param_grid = {

   'hidden_layer_sizes': [(1,), (10,),(50,),(120,),(150,)],

   'activation': ['relu', 'tanh', 'logistic','identity'],

   'learning_rate_init': [0.1,0.001, 0.0001],
```

```
    'max_iter': [1,10, 100, 1000],
}



perceptron = MLPRegressor(random_state=1) # Crear el modelo base

scoring = {'R2': make_scorer(r2_score), 'RMSE': make_scorer(mean_squared_error,
squared=False)} #Se añade el scoring con R2 y RMSE


################################################

grid_search = GridSearchCV(perceptron, param_grid, cv=5, scoring=scoring, refit='R2') # Crear el
objeto GridSearchCV con el modelo y la grilla de hiperparámetros

grid_search.fit(X_train_scaled, y_train_scaled) # Entrenar el modelo para cada combinación de
hiperparámetros


# Obtener la mejor combinación de hiperparámetros y su rendimiento

best_params = grid_search.best_params_

best_score = grid_search.best_score_


# Entrenar un nuevo modelo con los mejores hiperparámetros encontrados

best_perceptron = MLPRegressor(random_state=1, **best_params)

best_perceptron.fit(X_train_scaled, y_train_scaled)


# Calcular las métricas de evaluación del modelo optimizado

y_pred = best_perceptron.predict(X_train_scaled)

y_pred_test = best_perceptron.predict(X_test_scaled)


##############################


r2 = r2_score(y_train_scaled, y_pred)

rmse = np.sqrt(mean_squared_error(y_train_scaled, y_pred))
```

```
r2_test = r2_score(y_test_scaled, y_pred_test)

rmse_test = np.sqrt(mean_squared_error(y_test_scaled, y_pred_test))


print("Mejores hiperparámetros:", best_params)

print("R2 Train:", r2)

print("RMSE Train:", rmse)


print("R2 Test:", r2_test)

print("RMSE Test:", rmse_test)


lin_regplotml2(X_train_scaled, y_train_scaled, best_perceptron)

plt.xlabel('Harvest')

plt.ylabel('Production of avocado (kg)')

plt.title('Multilayer Regressor Train')

plt.show()


lin_regplotml2(X_test_scaled, y_test_scaled, best_perceptron)

plt.xlabel('Harvest')

plt.ylabel('Production of avocado (kg)')

plt.title('Multilayer Regressor Test')

plt.show()


plt.scatter(y_train, y_train - y_pred, c='steelblue', marker='o', edgecolor='white', label='Training
data')

plt.scatter(y_test, y_test - y_pred_Test, c='limegreen', marker='s', edgecolor='white', label='Test
data')

plt.xlabel('Predicted values')

plt.ylabel('Residuals')

plt.legend(loc='upper left')
```

```
plt.hlines(y=0, xmin=-0, xmax=5, color='black', lw=2)
plt.xlim([0, 5])
plt.show()

#SVR#

from sklearn.svm import SVR
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import r2_score, mean_squared_error
from math import sqrt

regr = make_pipeline( StandardScaler(),SVR(C=1.0, epsilon=0.1))
regr.fit(X_train, y_train)

y_pred = regr.predict(X_train)
y_pred_test = regr.predict(X_test)

r2 = r2_score(y_train, y_pred)
r2_test = r2_score(y_test, y_pred_test)
rmse = sqrt(mean_squared_error(y_train, y_pred))
rmse_test = sqrt(mean_squared_error(y_test, y_pred_test))

print("R^2:", r2)
print("RMSE:", rmse)
print("R^2 test:", r2_test)
print("RMSE test:", rmse_test)

lin_regplotmlALTER(X_train, y_train, regr)
```

```
plt.xlabel('Harvest')

plt.ylabel('Production of avocado (kg)')

plt.title('Support Vector Regressor')

plt.show()


lin_regplotmlALTER(X_test, y_test, regr)

plt.xlabel('Harvest')

plt.ylabel('Production of avocado (kg)')

plt.title('Support Vector Regressor')

plt.show()


plt.scatter(y_train, y_train - y_pred, c='steelblue', marker='o', edgecolor='white', label='Training
data')

plt.scatter(y_test, y_test - y_predTest, c='limegreen', marker='s', edgecolor='white', label='Test
data')

plt.xlabel('Predicted values')

plt.ylabel('Residuals')

plt.legend(loc='upper left')

plt.hlines(y=0, xmin=-0, xmax=5, color='black', lw=2)

plt.xlim([0, 5])

plt.show()


################################ VERSIÓN ACTUALIZADA DE SVR QUE CALCULA R2 Y EL RMSE
########################################

from sklearn.model_selection import GridSearchCV

from sklearn.svm import SVR

from sklearn.metrics import mean_squared_error, r2_score

np.random.seed(1)


sc_x = StandardScaler()
```

```
sc_y = StandardScaler()


X_train_sc = sc_x.fit_transform(X_train)
y_train_sc = sc_x.fit_transform(y_train[:,np.newaxis]).flatten()


X_test_sc = sc_x.fit_transform(X_test)
y_test_sc = sc_x.fit_transform(y_test[:,np.newaxis]).flatten()


parameters = {'C': [0.1, 1, 10, 20, 25, 30],
          'kernel': ['poly', 'linear', 'rbf', 'sigmoid'],
          'epsilon': [0.1, 0.2],
          'degree': [5, 10, 15]
}


svr = SVR()
#Se añade el scoring con R2 y RMSE
scoring = {'R2': make_scorer(r2_score), 'RMSE': make_scorer(mean_squared_error,
squared=False)}


# Crear el objeto GridSearchCV
grid_search = GridSearchCV(svr, parameters, cv=5, scoring=scoring, refit='R2')
grid_search.fit(X_train_sc, y_train_sc)


best_params = grid_search.best_params_


# Entrenar un nuevo modelo con los mejores hiperparámetros encontrados
best_svr = SVR(**best_params)
best_svr.fit(X_train_sc, y_train_sc)
```

```
# Calcular las métricas de evaluación del modelo optimizado
y_pred = best_svr.predict(X_train_sc)
y_pred_test = best_svr.predict(X_test_sc)

##############################

r2 = r2_score(y_train_sc, y_pred)
rmse = np.sqrt(mean_squared_error(y_train_sc, y_pred))

r2_test = r2_score(y_test_sc, y_pred_test)
rmse_test = np.sqrt(mean_squared_error(y_test_sc, y_pred_test))



print("Mejores parámetros:", best_params)
print("R2:", r2)
print("RMSE:", rmse)
print("R2 Test:", r2_test)
print("RMSE Test:", rmse_test)

lin_regplotml2(X_train_sc, y_train_sc, best_svr)
plt.xlabel('Harvest')
plt.ylabel('Production of avocado (kg)')
plt.title('Support Vector Regressor Train')
plt.show()

lin_regplotml2(X_test_sc, y_test_sc, best_svr)
plt.xlabel('Harvest')
plt.ylabel('Production of avocado (kg)')
plt.title('Support Vector Regressor Test')
```

```
plt.show()


plt.scatter(y_train, y_train - y_pred, c='steelblue', marker='o', edgecolor='white', label='Training data')

plt.scatter(y_test, y_test - y_predTest, c='limegreen', marker='s', edgecolor='white', label='Test data')

plt.xlabel('Predicted values')

plt.ylabel('Residuals')

plt.legend(loc='upper left')

plt.hlines(y=0, xmin=-0, xmax=5, color='black', lw=2)

plt.xlim([0, 5])

plt.show()


#RANDOM FOREST#


#################### NUEVA VERSIÓN PROPUESTA DE RANDOM FOREST CON R2 Y RMSE ####################


from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import r2_score, mean_squared_error

from math import sqrt


random_forest_regressor = RandomForestRegressor(random_state=1)

random_forest_regressor.fit(X_train, y_train)


y_prediction = random_forest_regressor.predict(X_train)

y_prediction_test = random_forest_regressor.predict(X_test)


r2 = r2_score(y_train, y_prediction)

rmse = sqrt(mean_squared_error(y_train, y_prediction))
```

```
r2_test = r2_score(y_test, y_prediction_test)

rmse_test = sqrt(mean_squared_error(y_test, y_prediction_test))




print("R^2:", r2)

print("RMSE:", rmse)

print("R^2_test:", r2_test)

print("RMSE Test:", rmse_test)


lin_regplotmlALTER(X_train, y_train, random_forest_regressor)

plt.xlabel('Harvest')

plt.ylabel('Production of avocado (kg)')

plt.title('Random Forest')

plt.show()


lin_regplotmlALTER(X_test, y_test, random_forest_regressor)

plt.xlabel('Harvest')

plt.ylabel('Production of avocado (kg)')

plt.title('Random Forest')

plt.show()


plt.scatter(y_train, y_train - y_pred, c='steelblue', marker='o', edgecolor='white', label='Training
data')

plt.scatter(y_test, y_test - y_predTest, c='limegreen', marker='s', edgecolor='white', label='Test
data')

plt.xlabel('Predicted values')

plt.ylabel('Residuals')

plt.legend(loc='upper left')
```

```
plt.hlines(y=0, xmin=-0, xmax=5, color='black', lw=2)

plt.xlim([0, 5])

plt.show()


##################### VERSIÓN "2" DEL RANDOM FOREST REGRESSOR QUE CALCULA R2 Y RMSE
#################################



from sklearn.ensemble import RandomForestRegressor

from sklearn.model_selection import GridSearchCV

from sklearn.metrics import r2_score, mean_squared_error

from math import sqrt



param_grid = {

   'n_estimators': [100],

   'max_depth': [None],

   'min_samples_split': [2],

   'min_samples_leaf': [1],

   'max_leaf_nodes': [None]

}



random_forest = RandomForestRegressor(random_state=1)


scoring = {'R2': make_scorer(r2_score)}



grid_search = GridSearchCV(random_forest, param_grid, cv=5, scoring=scoring, refit='R2')
```

```
grid_search.fit(X_train, y_train)

best_params = grid_search.best_params_

best_random_forest_regressor = RandomForestRegressor(**best_params, random_state=1)
best_random_forest_regressor.fit(X_train, y_train)

y_prediction = best_random_forest_regressor.predict(X_train)
y_prediction_test = best_random_forest_regressor.predict(X_test)

r2 = r2_score(y_train, y_prediction)
rmse = sqrt(mean_squared_error(y_train, y_prediction))

r2_test = r2_score(y_test, y_prediction_test)
rmse_test = sqrt(mean_squared_error(y_test, y_prediction_test))



print("Mejores hiperparámetros:", best_params)
print("R^2:", r2)
print("RMSE:", rmse)
print("R^2 Test:", r2_test)
print("RMSE Test:", rmse_test)

lin_regplotmlALTER(X_train, y_train, best_random_forest_regressor)
plt.xlabel('Harvest')
plt.ylabel('Production of avocado (kg)')
plt.title('Random Forest Train')
plt.show()
```

```
lin_regplotmlALTER(X_test, y_test, best_random_forest_regressor)

plt.xlabel('Harvest')

plt.ylabel('Production of avocado (kg)')

plt.title('Random Forest Test')

plt.show()


plt.scatter(y_train, y_train - y_pred, c='steelblue', marker='o', edgecolor='white', label='Training data')

plt.scatter(y_test, y_test - y_predTest, c='limegreen', marker='s', edgecolor='white', label='Test data')

plt.xlabel('Predicted values')

plt.ylabel('Residuals')

plt.legend(loc='upper left')

plt.hlines(y=0, xmin=-0, xmax=5, color='black', lw=2)

plt.xlim([0, 5])

plt.show()


%pip install m5py

X_train

X_test


# M5PRIME simplificado

# https://smarie.github.io/python-m5p/generated/gallery/

from m5py import M5Prime, export_text_m5


# Conversion a numpy para la versión actual

Xtrain = X_train.to_numpy()

ytrain = y_train.to_numpy()
```

```
Xtest = X_test.to_numpy()
ytest = y_test.to_numpy()

print(Xtrain.shape)
print(ytrain.shape)

regr_1 = M5Prime(random_state=1)
regr_1_label = "Tree"
regr_1.fit(Xtrain, ytrain)

# Metricas
# Realizar predicciones en los datos de prueba
y_pred_1 = regr_1.predict(Xtrain)
# Calcular el coeficiente de determinación R2
r2 = r2_score(ytrain, y_pred_1)
# Calcular el error cuadrático medio RMSE
rmse = np.sqrt(mean_squared_error(ytrain, y_pred_1))


# Realizar predicciones en los datos de prueba
y_pred_1_test = regr_1.predict(Xtest)
# Calcular el coeficiente de determinación R2
r2_1_test = r2_score(ytest, y_pred_1_test)
# Calcular el error cuadrático medio RMSE
rmse_1_test = np.sqrt(mean_squared_error(ytest, y_pred_1_test))

# Imprimir los resultados
print("R2:", r2)
print("RMSE:", rmse)
```

```
print("R2 test:", r2_1_test)
print("RMSE test:", rmse_1_test)

# M5PRIME
import numpy as np
import matplotlib.pyplot as plt
from m5py import M5Prime, export_text_m5



regr_1 = M5Prime(use_smoothing=False, use_pruning=False, random_state=1)
regr_1_label = "Tree 1"
regr_1.fit(X_train.to_numpy(), y_train.to_numpy())

# Prediction
y_1 = regr_1.predict(X_train.to_numpy())

# Print the trees
print("\n----- %s" % regr_1_label)
print(regr_1.as_pretty_text())

################################### propuesta 2 que calcula R2 y RMSE
###################################

from sklearn.model_selection import GridSearchCV
from sklearn.metrics import mean_squared_error, r2_score

parameters = {
    'max_depth': [2, 4, 6, 8, 10],
```

```
    'min_samples_leaf': [1, 2, 3, 4, 5, 6],
    'min_samples_split': [2, 4, 6, 8, 10, 12],
    'use_smoothing': [False, True],
    'use_pruning': [False, True]
}

model = M5Prime(random_state=1)

scoring = {'R2': make_scorer(r2_score), 'RMSE': make_scorer(mean_squared_error,
squared=False)}

grid_search = GridSearchCV(model, parameters, cv=5, scoring=scoring, refit='R2')
grid_search.fit(Xtrain, ytrain)

best_params = grid_search.best_params_
best_model = M5Prime(**best_params)
best_model.fit(Xtrain, ytrain)

y_pred = best_model.predict(Xtrain)
r2 = r2_score(ytrain, y_pred)
rmse = mean_squared_error(ytrain, y_pred, squared=False)


y_pred_test = best_model.predict(Xtest)
r2_test = r2_score(ytest, y_pred_test)
rmse_test = mean_squared_error(ytest, y_pred_test, squared=False)
```

```
print("Mejores parámetros:", best_params)

print("R2:", r2)

print("RMSE:", rmse)

print("R2 test:", r2_test)

print("RMSE test:", rmse_test)


lin_regplotmlALTER(X_train, y_train, best_model)

plt.xlabel('Harvest')

plt.ylabel('Production of avocado (kg)')

plt.title('M5 Prime Train')

plt.show()


lin_regplotmlALTER(X_test, y_test, best_model)

plt.xlabel('Harvest')

plt.ylabel('Production of avocado (kg)')

plt.title('M5 Prime Test')

plt.show()


plt.scatter(y_train, y_train - y_pred, c='steelblue', marker='o', edgecolor='white', label='Training
data')

plt.scatter(y_test, y_test - y_predTest, c='limegreen', marker='s', edgecolor='white', label='Test
data')

plt.xlabel('Predicted values')

plt.ylabel('Residuals')

plt.legend(loc='upper left')

plt.hlines(y=0, xmin=-0, xmax=5, color='black', lw=2)

plt.xlim([0, 5])

plt.show()


################################### FIN ###################################
```

©2024 Paris Arizmendi/Universidad Autónoma del Estado de Morelos

Cuernavaca, Morelos a 7 marzo del 2024.

DR. FELIPE DE JESÚS BONILLA SÁNCHEZ
DIRECTOR DE LA FACULTAD DE CONTADURÍA, ADMINISTRACIÓN E INFORMÁTICA.
PRESENTE

En mi carácter de revisor de tesis, hago de su conocimiento que he leído con interés la tesis para obtener el grado de la Maestría en Optimización y Cómputo Aplicado, del estudiante Paul Paris Arizmendi Peralta, con matrícula 10053356, con el título **Estudio de técnicas de Machine Learning para estimar la producción de aguacate en la zona norte del Estado de Morelos**, por lo cual, me permito informarle que después de una revisión cuidadosa de dicha tesis, me permito expresar mi **VOTO APROBATORIO** por lo que de mi parte no existe inconveniente para que el estudiante continúe con los trámites que la Universidad Autónoma del Estado de Morelos tenga establecidos para obtener el grado mencionado.

Atentamente
Por una humanidad culta

Dr. José Alberto Hernández Aguilar
Profesor- investigador
Facultad de Contaduría, Administración e Informática

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS

Se expide el presente documento firmado electrónicamente de conformidad con el ACUERDO GENERAL PARA LA CONTINUIDAD DEL FUNCIONAMIENTO DE LA UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS DURANTE LA EMERGENCIA SANITARIA PROVOCADA POR EL VIRUS SARS-COV2 (COVID-19) emitido el 27 de abril del 2020.

El presente documento cuenta con la firma electrónica UAEM del funcionario universitario competente, amparada por un certificado vigente a la fecha de su elaboración y es válido de conformidad con los LINEAMIENTOS EN MATERIA DE FIRMA ELECTRÓNICA PARA LA UNIVERSIDAD AUTÓNOMA DE ESTADO DE MORELOS emitidos el 13 de noviembre del 2019 mediante circular No. 32.

Sello electrónico

JOSE ALBERTO HERNANDEZ AGUILAR | Fecha:2024-03-07 16:49:09 | Firmante

vdNmAOKVsUaWH9jhBgU/dqJ2VIBeCymHycJcAm9icDAm6z76cJXg8/XrEnMTXm02mG5xQO+4yT9U2qbH1KV/DZ0K1DnYLNSAaURS7F4JKMDSFwsJnzNRF1vFU3TVtu6nzhfXHNnqQMJb6HTCB0hSMzWdv5Ehi4DIzVtgcoMmvrD71FuZZfKN50W56brmLBUmOECeYOeuLvUkK+nTQBNpYjgfziwOIxZIXEzOIk5Uf8XNfG3ddlGx+KflbIk17+AcSTwEOlnmqYmajUwfvsIx6XmJN2NAX8ldj+Cd7xeVDd1XeehP3zXbWGnRqernPwOPQ9HD+kBNmkJckk+yLnCQAg==

Puede verificar la autenticidad del documento en la siguiente dirección electrónica o escaneando el código QR ingresando la siguiente clave:

kc4hNPzi5

https://efirma.uaem.mx/noRepudio/IqkhP7aHSk7VytMf259XzrCz125VWCaV

Cuernavaca, Morelos a 7 de marzo del 2024.

DR. FELIPE DE JESÚS BONILLA SÁNCHEZ
DIRECTOR DE LA FACULTAD DE CONTADURÍA,
ADMINISTRACIÓN E INFORMÁTICA.
PRESENTE

En mi carácter de revisor de tesis, hago de su conocimiento que he leído con interés la tesis para obtener el grado de la Maestría en Optimización y Cómputo Aplicado, del estudiante Paul Paris Arizmendi Peralta, con matrícula 10053356, con el título **Estudio de técnicas de Machine Learning para estimar la producción de aguacate en la zona norte del Estado de Morelos**, por lo cual, me permito informarle que después de una revisión cuidadosa de dicha tesis, me permito expresar mi **VOTO APROBATORIO** por lo que de mi parte no existe inconveniente para que el estudiante continúe con los trámites que la Universidad Autónoma del Estado de Morelos tenga establecidos para obtener el grado mencionado.

Atentamente
Por una humanidad culta

Dr. Pedro Moreno Bernal
Profesor- investigador
Facultad de Contaduría, Administración e Informática

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS

Se expide el presente documento firmado electrónicamente de conformidad con el ACUERDO GENERAL PARA LA CONTINUIDAD DEL FUNCIONAMIENTO DE LA UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS DURANTE LA EMERGENCIA SANITARIA PROVOCADA POR EL VIRUS SARS-COV2 (COVID-19) emitido el 27 de abril del 2020.

El presente documento cuenta con la firma electrónica UAEM del funcionario universitario competente, amparada por un certificado vigente a la fecha de su elaboración y es válido de conformidad con los LINEAMIENTOS EN MATERIA DE FIRMA ELECTRÓNICA PARA LA UNIVERSIDAD AUTÓNOMA DE ESTADO DE MORELOS emitidos el 13 de noviembre del 2019 mediante circular No. 32.

Sello electrónico

PEDRO MORENO BERNAL | Fecha:2024-03-07 15:19:17 | Firmante

PW2Nw7aXCEDJYoWjjhYt0oClvhW+Ed2a+EJ0RhPx3ZSNw8frYpCQLc2vdwmm31f1BN9iWvHRxzE3qOgBrzUYMzVvLPYSy7hM2zRFRu2Dq1IlI5PqSiLYnAhqaS5dq3j8ra6tZSwId32cr4fbYM40Plo6DlZW/4zHHqEryPLsljIYK+PJ+9BkRyMN4SXIo5kn8B8FXGzsWpu9vKn8pKDqDJ+c8+HwQruWSEmh93r5fkud/2LL/A13+3g5j2CZaWXJrYuOQ0xoX9Xs3kaMq/2vEOk0B56k+DAggfU7ee8i33XPwO853E6m6+Krck/OC5VjRKFMyIk5xbGq9T7yMNHU9A==

Puede verificar la autenticidad del documento en la siguiente dirección electrónica o escaneando el código QR ingresando la siguiente clave:

VWLM3irKp

https://efirma.uaem.mx/noRepudio/3kZmT9RbmI5vqM946qr6MePJNAV6SKkQ

Cuernavaca, Morelos a 7 de marzo del 2024.

DR. FELIPE DE JESÚS BONILLA SÁNCHEZ
DIRECTOR DE LA FACULTAD DE CONTADURÍA,
ADMINISTRACIÓN E INFORMÁTICA.
PRESENTE

En mi carácter de revisor de tesis, hago de su conocimiento que he leído con interés la tesis para obtener el grado de la Maestría en Optimización y Cómputo Aplicado, del estudiante Paul Paris Arizmendi Peralta, con matrícula 10053356, con el título **Estudio de técnicas de Machine Learning para estimar la producción de aguacate en la zona norte del Estado de Morelos**, por lo cual, me permito informarle que después de una revisión cuidadosa de dicha tesis, me permito expresar mi **VOTO APROBATORIO** por lo que de mi parte no existe inconveniente para que el estudiante continúe con los trámites que la Universidad Autónoma del Estado de Morelos tenga establecidos para obtener el grado mencionado.

Atentamente
Por una humanidad culta

Dr. Sergio Nesmachnow
Profesor- investigador

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS

Se expide el presente documento firmado electrónicamente de conformidad con el ACUERDO GENERAL PARA LA CONTINUIDAD DEL FUNCIONAMIENTO DE LA UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS DURANTE LA EMERGENCIA SANITARIA PROVOCADA POR EL VIRUS SARS-COV2 (COVID-19) emitido el 27 de abril del 2020.

El presente documento cuenta con la firma electrónica UAEM del funcionario universitario competente, amparada por un certificado vigente a la fecha de su elaboración y es válido de conformidad con los LINEAMIENTOS EN MATERIA DE FIRMA ELECTRÓNICA PARA LA UNIVERSIDAD AUTÓNOMA DE ESTADO DE MORELOS emitidos el 13 de noviembre del 2019 mediante circular No. 32.

Sello electrónico

SERGIO NESMACHNOW | Fecha:2024-03-07 17:05:10 | Firmante

VnWhD7oE08X56L/m+T36K3UKyv92uFHuL8PUP0U/yzpQWWz88OJquQUbM9kb3brK5SbbWNGnHE0zQQI1pCcVvhV2cpoKCPUkAen2d13B3z1IxdFjAkuJWXj5bL/h68Jww3pt7tF9SwP0WW+IfUU48uJV6+fAYWLv+g+vDqfKKFu5djVfLIdj3osQVvW16cUqdZ6ZG4Kbk7tYFNWJaX+wwJ0NtSyvVW+was4gkPJxv8IxoBuGSFUR07JToTPKLSABCyB6b7rh4aNGyBVyBs5sk3W95hjLtHkks8nqoTBbAfeOuvW9FL2QIu8+DQmWmWx2gFOFNdI0Xcd0VQy8Jt6rHg==

Puede verificar la autenticidad del documento en la siguiente dirección electrónica o escaneando el código QR ingresando la siguiente clave:

Lnlqdbpyh

https://efirma.uaem.mx/noRepudio/fVsyC6UJjYOnBNB71Fi2tzDqU4jTkoOf

Cuernavaca, Morelos a 7 marzo del 2024.

DR. FELIPE DE JESÚS BONILLA SÁNCHEZ
DIRECTOR DE LA FACULTAD DE CONTADURÍA, ADMINISTRACIÓN E INFORMÁTICA.
PRESENTE

En mi carácter de revisor de tesis, hago de su conocimiento que he leído con interés la tesis para obtener el grado de la Maestría en Optimización y Cómputo Aplicado, del estudiante Paul Paris Arizmendi Peralta, con matrícula 10053356, con el título **Estudio de técnicas de Machine Learning para estimar la producción de aguacate en la zona norte del Estado de Morelos**, por lo cual, me permito informarle que después de una revisión cuidadosa de dicha tesis, me permito expresar mi **VOTO APROBATORIO** por lo que de mi parte no existe inconveniente para que el estudiante continúe con los trámites que la Universidad Autónoma del Estado de Morelos tenga establecidos para obtener el grado mencionado.

Atentamente
Por una humanidad culta

Dr. Outmane Oubram
Profesor- investigador
Facultad de Contaduría, Administración e Informática

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS

Se expide el presente documento firmado electrónicamente de conformidad con el ACUERDO GENERAL PARA LA CONTINUIDAD DEL FUNCIONAMIENTO DE LA UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS DURANTE LA EMERGENCIA SANITARIA PROVOCADA POR EL VIRUS SARS-COV2 (COVID-19) emitido el 27 de abril del 2020.

El presente documento cuenta con la firma electrónica UAEM del funcionario universitario competente, amparada por un certificado vigente a la fecha de su elaboración y es válido de conformidad con los LINEAMIENTOS EN MATERIA DE FIRMA ELECTRÓNICA PARA LA UNIVERSIDAD AUTÓNOMA DE ESTADO DE MORELOS emitidos el 13 de noviembre del 2019 mediante circular No. 32.

Sello electrónico

OUTMANE OUBRAM | Fecha:2024-03-07 15:44:09 | Firmante

GtADEgXbuHGBWCFR4OsMp+6O02AO+ryGJReCx7p3sUfiEEy0fsR/j6VHjAsIs9rmL+SP21En6tT6j+03BvAsnnxL9FCTm4pR4cQNuo45K6YUTylGl5PQ46GchN8jbgOAUse/QXqfUmn/5+HjnhlTEGq73dBLVjs5TVanlcvkIli1ISwyBwHZI1xYyuOOrBkVEuo5Bbyh8Su2s5XAMX3a49BZ0+X1H66pp6j9wITBBg1tZkKedzdyIDBC5Z+oATzpoP0MsYglkukM284fIdLZeP41+IHSRC0FMq9u+puvwR9ZI4tTVXkEOUWdr7MI+Eat9dAOMzt7pqTxFf8QWqVVzQ==

Puede verificar la autenticidad del documento en la siguiente dirección electrónica o escaneando el código QR ingresando la siguiente clave:

9c2OQRThC

https://efirma.uaem.mx/noRepudio/f21xfxzng3IfnVHCe3GbyCcPsNwFf1l8

Cuernavaca, Morelos a 07 de marzo del 2024.

DR. FELIPE DE JESÚS BONILLA SÁNCHEZ
DIRECTOR DE LA FACULTAD DE CONTADURÍA,
ADMINISTRACIÓN E INFORMÁTICA.
PRESENTE

En mi carácter de revisor de tesis, hago de su conocimiento que he leído con interés la tesis para obtener el grado de la Maestría en Optimización y Cómputo Aplicado, del estudiante Paul Paris Arizmendi Peralta, con matrícula 10053356, con el título **Estudio de técnicas de Machine Learning para estimar la producción de aguacate en la zona norte del Estado de Morelos**, por lo cual, me permito informarle que después de una revisión cuidadosa de dicha tesis, me permito expresar mi **VOTO APROBATORIO** por lo que de mi parte no existe inconveniente para que el estudiante continúe con los trámites que la Universidad Autónoma del Estado de Morelos tenga establecidos para obtener el grado mencionado.

Atentamente
Por una humanidad culta

Dra. Lorena Díaz González
Profesora- investigadora
Facultad de Contaduría, Administración e Informática

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS

Se expide el presente documento firmado electrónicamente de conformidad con el ACUERDO GENERAL PARA LA CONTINUIDAD DEL FUNCIONAMIENTO DE LA UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS DURANTE LA EMERGENCIA SANITARIA PROVOCADA POR EL VIRUS SARS-COV2 (COVID-19) emitido el 27 de abril del 2020.

El presente documento cuenta con la firma electrónica UAEM del funcionario universitario competente, amparada por un certificado vigente a la fecha de su elaboración y es válido de conformidad con los LINEAMIENTOS EN MATERIA DE FIRMA ELECTRÓNICA PARA LA UNIVERSIDAD AUTÓNOMA DE ESTADO DE MORELOS emitidos el 13 de noviembre del 2019 mediante circular No. 32.

Sello electrónico

LORENA DIAZ GONZALEZ | Fecha:2024-03-14 08:33:34 | Firmante

dwQc7SE5eFa455OX4pmtp11NpAHEBfTmH7eN6RDLavoFPZMzK2YrV+HEVQ7Ch43BjWGqcs4bXre+RjpeIisUOHwgwY26XGrmz7F27se6iwJN/kyOan8+XkMc9x8W+OrlsnPnSzAO/U5KWeOGJv6JoHuZ+ycCM9nY3mPeFIldCMLHsWA42Lm7z7s9Cpjiq7+wkQI+ztXqYs8rRfO1U7DU6ybY+wJc1UHN2Ux6Jznun1Psv6u1o3r74ITC9FHAgHys6RSJhG8rvPjBgG7IfgknpDdY8syj8PN1xg93ii0QmmFK745G9RSoqZNltFbfAmJbBW+bcBvr4RkgMaNdlAQtRg==

Puede verificar la autenticidad del documento en la siguiente dirección electrónica o escaneando el código QR ingresando la siguiente clave:

HCvpRaMUV

https://efirma.uaem.mx/noRepudio/7rF0gJlVoTUo1tJKKNXo5ujmNtCivrQz