Artículos

Desarrollo de un modelo predictivo utilizando técnicas de aprendizaje supervisado para detectar la moniliasis en plantas de cacao de la Provincia de Orellana

Development of a predictive model using supervised learning techniques to detect moniliasis in cocoa plants in the Province of Orellana

Castillo Lapo, Danny Jesiel; Ramírez Cambo, Mariuxi Noemí; Chango Sailema, Wilson Gustavo; Aguilar Encarnación, Pedro Stalyn

Danny Jesiel Castillo Lapo
jesiel.castillo@espoch.edu.ec
Escuela Superior Politécnica de Chimborazo, Ecuador

Mariuxi Noemí Ramírez Cambo
mariuxi.ramirez@espoch.edu.ec
Escuela Superior Politécnica de Chimborazo, Ecuador

Wilson Gustavo Chango Sailema
wilson.chango@espoch.edu.ec
Escuela Superior Politécnica de Chimborazo, Ecuador

Pedro Stalyn Aguilar Encarnación
pedro.aguilar@espoch.edu.ec
Escuela Superior Politécnica de Chimborazo, Ecuador

Revista Tecnológica ESPOL - RTE
Escuela Superior Politécnica del Litoral, Ecuador
ISSN: 0257-1749
ISSN-e: 1390-3659
Periodicidad: Semestral
vol. 35, núm. 3, 2023
rte@espol.edu.ec

Recepción: 05 Septiembre 2023
Aprobación: 14 Diciembre 2023

URL: http://portal.amelica.org/ameli/journal/844/8444930003/

Resumen: La respuesta al enigma de la moniliasis se encuentra en la ciencia y la tecnología con el proyecto desarrollado en la Provincia de Orellana, en donde la moniliasis es una enfermedad fúngica que causa efectos devastadores incluyen do la pudrición de las flores, vainas y frutos de cacao, lo que conlleva pérdidas significativas a los agricultores. La moniliasis afecta gravemente a los cultivos de cacao y resulta difícil detectar su presencia tempranamente. Para lograr la detección de esta enfermedad, se recopilaron datos obtenidos de sensores y registros manuales para entrenar y validar un modelo predictivo mediante aprendizaje supervisado, en donde se analizó las condiciones ambientales y los síntomas de la enfermedad. Se aplicó la metodología de la ciencia del diseño basada en tres ciclos: el ciclo de relevancia, rigor y diseño. En el ciclo de relevancia se definió el problema y la necesidad del modelo, en el ciclo de rigor se realizó una investigación preliminar para determinar la viabilidad del objetivo y, por último, en el ciclo de diseño se modelaron los datos con algoritmos de aprendizaje automático y se implementó el modelo de predicción, probándolo para verificar su correcto funcionamiento.
The model was shared with cocoa farming families in Orellana, demonstrating its effectiveness. This will allow farmers to take appropriate and timely control measures to prevent the spread of the disease and thus increase cocoa production and quality.

Palabras clave: Scikit-Learn, PWA, MongoDB, React.js, Python.

Abstract: The answer to the moniliasis enigma lies in science and technology with the project developed in Orellana Province, where moniliasis is a fungal disease that causes significant losses to farmers. Moniliasis severely affects cocoa crops, and it is difficult to detect its presence early. Data from sensors and manual records were collected to train and validate a predictive model using supervised learning, where environmental conditions and disease symptoms were analysed. Design science methodology was applied based on three cycles: the relevance, rigour and design cycle. In the relevance cycle the

problem and the need for the model were defined, in the rigour cycle a preliminary investigation was carried out to determine the feasibility of the objective and finally in the design cycle the data was modelled with machine learning algorithms and the prediction model was implemented and tested to verify its correct functioning.
The model was shared with cocoa farming families in Orellana, demonstrating its effectiveness. This will allow farmers to take appropriate and timely control measures to prevent the spread of the disease and thus increase cocoa production and quality.

Keywords: Scikit-Learn, PWA, MongoDB, React.js, Python.

Introducción

El cacao es un cultivo de importancia a escala mundial, pero su rendimiento está severamente limitado por enfermedades como la moniliophthora Pod Rot (MPR) causada por el hongo *Moniliophthora roreri*. Varios estudios demuestran que esta enfermedad es uno de los principales factores limitantes de la producción de cacao en América Latina (Leandro-Muñoz et al., 2017).

Caicedo (2019) destaca que el hongo *moniliophthora roreri* es considerado el mayor problema en Ecuador, generando pérdidas significativas a los agricultores y afectando la rentabilidad. Por otro lado, Jha et al. (2019) argumentan que la agricultura enfrenta desafíos todos los días, los que van desde la siembra hasta la cosecha de cultivos, la inteligencia artificial y el aprendizaje automático desempeñan un papel importante en la calidad de la cosecha de cultivos. Además, se menciona que la detección temprana de la moniliasis una enfermedad fúngica que afecta a varios tipos de plantas y frutas, en particular al cacao causando la pudrición de las vainas, lo que a su vez daña las semillas es crucial identificar rápidamente la presencia de esta enfermedad y tomar medidas preventivas o de control antes de que la infección se propague y cause daños significativos. Al identificar la moniliasis en sus etapas iniciales, se pueden implementar tratamientos adecuados y prácticas de gestión para minimizar el impacto en los cultivos y, en última instancia, proteger la producción agrícola.

El problema de investigación planteado es la detección de la moniliasis en plantas de cacao de la Provincia de Orellana. El objetivo principal es desarrollar un modelo predictivo utilizando técnicas de aprendizaje supervisado para detectar esta enfermedad.

Para cumplir con esta finalidad, se utilizó la metodología Ciencia del Diseño, un enfoque basado en la investigación científica, proporcionando un marco estructurado para abordar problemas complejos basándose en tres ciclos. Esta investigación es de suma importancia porque se puede detectar la presencia de la moniliasis a través de variables microclimáticas y cuantitativas: lluvia, temperatura, reacción de hipersensibilidad, punto de rocío, velocidad del viento, dirección y ráfagas, cantidad de plantas, frutos, incidencia y el porcentaje severidad de datos históricos recopilados manualmente y mediante el uso de un pluviómetro S-RGF-M002.

Para lograr que el modelo sea accesible de manera sencilla para los agricultores, se diseñó una Aplicación Web Progresiva (PWA). De acuerdo con la definición de Bernardi et al. (2018), una PW representa una innovadora metodología de desarrollo de software. A través de esta aplicación, los usuarios tienen la posibilidad de introducir datos relacionados con sus plantas de cacao y, como resultado, obtendrán el porcentaje de precisión de tener o no la enfermedad.

Por otro lado, el modelo se entrenó utilizando técnicas de aprendizaje supervisado el cual "está compuesto por algoritmos que intentan encontrar relaciones y dependencias entre un elemento objetivo" (Ovalle, 2022). Asimismo, para su entrenamiento se utilizó la librería scikit-learn. El backend del modelo está desarrollado

en Python, "un lenguaje de programación de alto nivel debido a su facilidad y código abierto" (Susilo et al., 2021).

Para el frontend se utilizó React.js, una librería de JavaScript. Boersma y Lungu (2021) argumentan que es ampliamente utilizada, ya que permite a los desarrolladores crear interfaces de usuario para la web. La base de datos utilizada fue MongoDB.

Se espera que el modelo predictivo desarrollado pueda detectar la presencia de moniliasis con un alto porcentaje de precisión, utilizando los datos históricos recopilados manualmente mediante el uso de un pluviómetro. Además, se espera que este modelo pueda ayudar a los agricultores a prevenir la aparición de la enfermedad y reducir las pérdidas en la cosecha del cacao.

El esquema para el desarrollo del presente proyecto es el siguiente: en el apartado 2 se describe la metodología utilizada para la implementación del modelo, en el apartado 3 se presentan los resultados obtenidos y discusión, y en la sección 4 se muestran las conclusiones del proyecto.

Metodología

Este proyecto se realizó en la Provincia de Orellana, cantón Francisco de Orellana. Se tomó una muestra de 20 familias cacaoteras para realizar el entrenamiento del modelo. Para implementar el modelo predictivo se utilizó la metodología Ciencia del Diseño (Horst Rittel, 1960). Esta metodología se basa en tres ciclos: relevancia, rigor y diseño. "La herramienta principal para estos ciclos es la investigación y búsqueda de información útil para la construcción de un artefacto dentro de un contexto" (Robles et al., 2019).

El ciclo de relevancia implica examinar los requisitos del mercado y el entorno en el que se utilizará el producto. El ciclo de rigor se basa en la búsqueda de información pertinente, como soluciones previas y los conocimientos técnicos necesarios. Finalmente, en el ciclo de diseño se evalúan varias respuestas al problema utilizando una variedad de herramientas para confirmar su eficacia. Estos tres ciclos se representan esquemáticamente en la Figura 1.

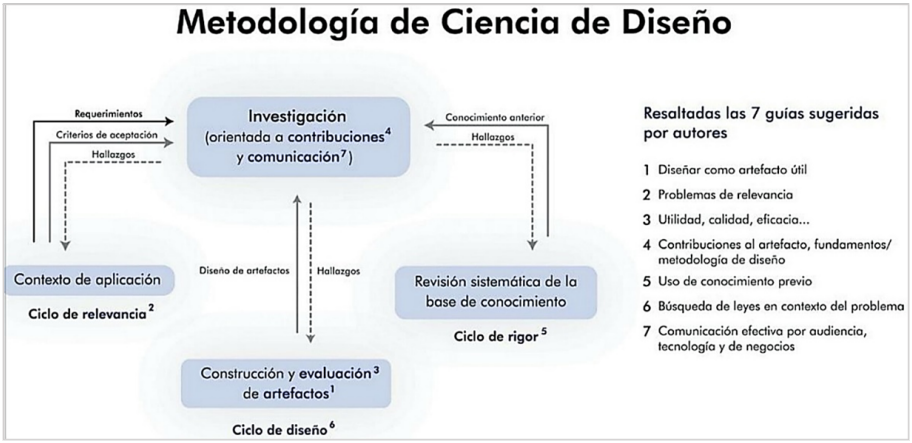

FIGURA 1

Metodología de Ciencia de Diseño aplicada al modelo predictivo

Nota: Esquema de Ciencia de Diseño tomada de (Robles et al., 2019).

Ciclo de relevancia

Definición del problema

Este estudio, se propuso abordar el problema de la moniliasis en plantas de cacao en la Provincia de Orellana. "La moniliasis es una enfermedad fúngica que ataca el cultivo de cacao, causada por el basidiomycete *Moniliophthorararoreri*" (Correa et al., 2014). Esta enfermedad genera pérdidas significativas en las cosechas de cacao, y para definir claramente este problema, se realizó una revisión de la literatura para entender cómo afecta al cultivo de cacao. También se entrevistó a expertos en el campo, como agrónomos y agricultores, de esta manera se obtuvo información de primera mano sobre el impacto de la enfermedad en la región. A partir de estos datos, se definió el problema como "La necesidad de predecir la aparición de la moniliasis en plantas de cacao en la provincia de Orellana para ayudar a los agricultores a prevenir la enfermedad y reducir las pérdidas en la cosecha".

Ciclo de rigor

Investigación preliminar

De acuerdo con sus investigaciones, Carrera et al. (2014) han demostrado que el cacao es de gran importancia económica y social en Ecuador, pues aproximadamente el 13% de la población económicamente activa agrícola de este país se relaciona de algún modo con dicho cultivo. De igual manera, Ricardez et al. (2016) mencionan que la moniliasis ocasiona daños en los frutos de cacao como deformaciones y manchas color café ("chocolate") en cualquier etapa de desarrollo, lo que tiene un alto impacto económico que ocasiona el abandono del cultivo, o su reemplazo.

La Provincia de Orellana está situada en la parte nororiental de la región amazónica donde habitan familias indígenas cacaoteras que cultivan orgánicamente este cultivo.

El cultivo de cacao y su impacto económico y social en Ecuador, particularmente en este sector, es de gran relevancia. Sin embargo, es importante destacar que, además de estos aspectos agrícolas y sociales, se presentan las diferentes herramientas y tecnologías que se utilizarán para llevar a cabo este proyecto.

El Aprendizaje Supervisado es una técnica de aprendizaje automático que construye un modelo predictivo utilizando datos de entrenamiento a partir de datos no etiquetados, "este algoritmo busca crear un modelo que pueda realizar predicciones acerca de los valores de respuesta para un nuevo conjunto de datos" (Gramajo et al., 2020).

Python es un lenguaje de programación de alto nivel interpretado, orientado a objetos, con semántica dinámica y administración automática de memoria (Fernández et al., 2018).

Scikit-Learn es una librería de código abierto en Python que se puede utilizar para el procesamiento de datos, reducción de la dimensionalidad, clasificación, regresión, agrupamiento y selección de modelos. Los resultados de la evaluación pueden ser en forma de tiempo de ejecución, precisión, matriz de confusión, tasa de falsos positivos, tasa de falsos negativos, precisión, recordar, y otros (Susanto et al., 2020).

Mongo DB es una base de datos con un entorno de código abierto que se fundamenta en el almacenamiento masivo de datos a través de archivos distribuidos con eficiencia de acceso.

React.js es una librería JavaScript de código abierto utilizada para construir interfaces de usuario interactivas y creativas que se emplea ampliamente en el desarrollo de aplicaciones web de una sola página (Single-Page Applications) y aplicaciones móviles. Fortunato & Bernardino (2018) afirman que la PWA es una nueva tecnología que permite que una aplicación esté disponible en cualquier dispositivo con acceso

a un navegador web, sin necesidad de desarrollar la aplicación de forma nativa, específicamente, para un dispositivo o sistema operativo determinado.

Docker es un proyecto de código abierto, independiente de lenguajes y bases de datos, ejecutándolos dentro de contenedores. Un contenedor es una agrupación de aplicaciones junto con sus dependencias, que comparten el kernel del sistema operativo (Oliveira et al., 2022).

Ciclo de diseño

Se definieron las siguientes variables para el entrenamiento del conjunto de datos: lluvia, temperatura, HR, punto de rocío, velocidad del viento, dirección y ráfagas, cantidad de plantas, frutos, incidencia y el porcentaje de severidad. Se realizaron pruebas para identificar valores atípicos y se evaluó el rendimiento de diferentes algoritmos para elegir el que obtenga mejor resultado de precisión.

Después de recolectar los datos de la muestra, se realizaron pruebas con diferentes algoritmos para identificar el mejor resultado. Se emplearon técnicas de análisis estadístico para evaluar la eficiencia, capacidad de almacenamiento, tiempo de respuesta y otras métricas pertinentes.

Para el modelado de datos se utilizaron las siguientes tecnologías: herramienta GitHub para el control de versiones del proyecto, lenguaje de programación Python para el backend, React.js para el frontend. Se eligió la base de datos mongodb y Docker para el despliegue de la aplicación.

Se implementaron medidas adecuadas para garantizar la privacidad y confidencialidad de los datos de acuerdo con la ley vigente, lo que implica garantizar que los datos sean almacenados y utilizados de manera segura, y que solo sean accesibles para las personas autorizadas involucradas en el proyecto.

Obtener una muestra representativa de la población de plantas de cacao puede ser un desafío logístico y requerir un muestreo cuidadoso al igual que la variabilidad de las condiciones ambientales puede dificultar la creación de un modelo efectivo en diferentes escenarios y ubicaciones. Otra limitación es la evolución y cambios en la moniliasis que pueden afectar la eficacia del modelo predictivo a medida que se enfrenta a nuevas cepas o cambios en la enfermedad.

Se anticipa que los hallazgos y la estructura del estudio son presentados de manera exhaustiva y comprensible para facilitar la reproducción de la investigación por parte de otros investigadores. Es esencial proporcionar una descripción precisa de los procedimientos y enfoques utilizados, asegurándose de que se presenten sin ambigüedades. Además, es necesario definir minuciosamente las variables y medidas involucradas en el estudio, brindando detalles específicos que permitan una comprensión completa de su significado y aplicabilidad.

Diseño conceptual

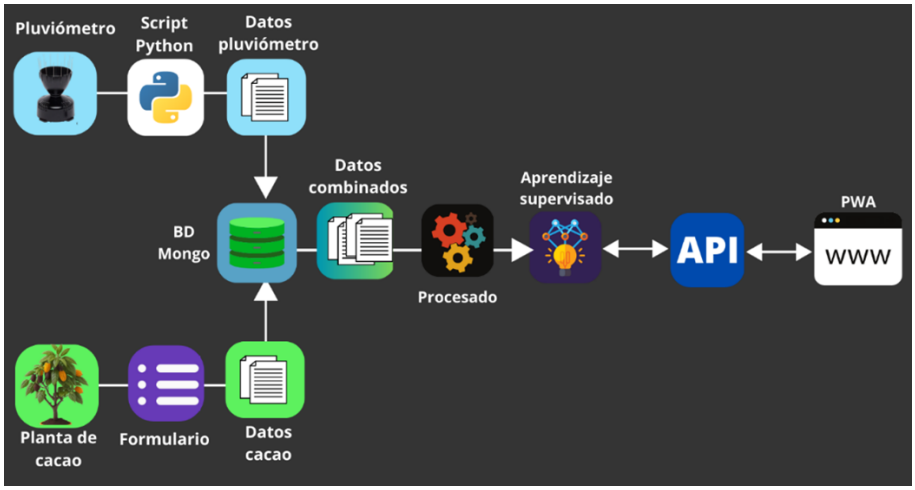

FIGURA 2
Arquitectura del Modelo de Predicción

Implementación y evaluación

Para evaluar el rendimiento de diferentes algoritmos en la predicción de la moniliasis en plantas de cacao, se utilizó un conjunto de datos que contenía información del sensor y de la planta. El conjunto de datos incluía 10 variables características: Rain, Temperature, RH (Relative Humidity), DewPoint, WindSpeed, GustSpeed, WindDirection, planta, fruto y severidad (%), y una variable objetivo para la predicción, que fue incidencia.

Así también, con el objetivo de comparar los resultados entre los datos originales, los datos normalizados y los datos discretizados, se aplicaron las técnicas de normalización y discretización a las variables características. La normalización se utilizó para ajustar los valores de las variables dentro de un rango específico y eliminar diferencias de escala, mientras que la discretización se empleó para convertir las variables continuas en categorías o intervalos discretos.

Al obtener los conjuntos de datos normalizados y discretizados, se procedió a evaluar los diferentes algoritmos utilizando estos conjuntos junto con los datos originales. De esta manera, se pudo analizar la influencia de la normalización y la discretización en el rendimiento de los algoritmos y determinar cuál de ellos producía las mejores predicciones. Esta comparación permitió examinar la efectividad de los procesos de normalización y discretización en la mejora de los resultados de predicción, así como comprender cómo estos procesos afectan la capacidad de los algoritmos para detectar patrones y relaciones en los datos.

Algoritmos para reducir la dimensionalidad

En la primera prueba se comparó el rendimiento de algoritmos KPC con tres kernels. Para cada algoritmo, se midió su precisión utilizando la métrica de precisión (accuracy score).

KPCA (Kernel PCA)

El método KPCA (Kernel PCA) es una extensión del PCA que permite realizar la reducción de dimensionalidad, empleando funciones de kernel no lineales. Se probaron tres kernels diferentes en KPCA:

Kernel: Linear

Se aplicó KPCA con un kernel lineal, lo que significa que se utilizó una función de kernel lineal para la reducción de dimensionalidad.

Kernel: Polynomial

En este caso, se utilizó un kernel polinomial en KPCA, lo que implica que se usó una función de kernel polinomial para la disminución de dimensionalidad.

Kernel: RBF (Radial Basis Function)

Se implementó un kernel RBF en KPCA, lo que significa que se empleó una función de kernel de base radial para la reducción de dimensionalidad.

Los resultados de las pruebas usando los datos originales, normalizar y discretizar, se presentan en la Tabla 1.

TABLA 1
Rendimiento de algoritmos KPC

ALGORITMO KPCA	KERNEL	VALOR OBTENIDO
Datos originales	Linear	1.0
	Polynomial	0.9865
	RBF	0.9373
Datos normalizados	Linear	0.8611
	Polynomial	0.8194
	RBF	0.8835
Datos discretizados	Linear	0.9373
	Polynomial	0.9492
	RBF	0.7731

El análisis de los resultados indica claramente que el kernel lineal obtuvo el mejor rendimiento en comparación con los kernels polinómico y RBF. Este kernel logró un puntaje más alto en todas las métricas evaluadas: el puntaje original, el puntaje normalizado y el puntaje discretizado. Estos resultados sugieren que la proyección de los datos en un espacio de menor dimensión, utilizando el kernel lineal, conservó mejor la estructura de los datos originales en comparación con los otros kernels. Por lo tanto, si se busca obtener el mejor rendimiento posible con el algoritmo KPCA, el kernel lineal sería la elección preferida con base a estos resultados.

En cuanto a las pruebas realizadas con los algoritmos PCA e IPCA, arrojaron los siguientes resultados (Tabla 2).

TABLA 2
Rendimiento de los algoritmos PCA e IPCA

TIPO DE DATOS	ALGORITMOS	VALOR OBTENIDO
Datos originales	PCA	1.0
	IPCA	1.0
Datos Normalizados	PCA	0.8373
	IPCA	0.8179
Datos discretizados	PCA	0.9373
	IPCA	0.9552

Los resultados de las pruebas muestran que, al trabajar con los datos originales, es decir sin normalizar y discretizar, se logró un resultado perfecto. Entonces, comparando los algoritmos PCA, IPCA y KPCA con el conjunto de datos originales, se obtuvo que los puntajes son los mismos. Los resultados se presentan en la Tabla 3.

TABLA 3
Rendimiento de algoritmos PCA, IPCA y KPCA

MÉTODO	KERNEL	VALOR OBTENIDO
PCA	-	1.0
IPCA	-	1.0
KPCA	Lineal	1.0

El algoritmo de PCA demostró un rendimiento excelente al obtener un valor de 1.0 en la métrica evaluada. Esto indica que PCA fue capaz de capturar eficientemente la varianza en los datos y proporcionar una representación compacta y significativa de las características originales. Además, PCA es ampliamente utilizado y reconocido en la comunidad científica, lo que brinda confianza en su aplicabilidad y resultados.

Una de las principales razones para elegir PCA es su capacidad de interpretación y comprensión de los datos. Al extraer los componentes principales, se puede identificar las características más relevantes y entender mejor las relaciones entre las variables originales. Esta interpretación es crucial para este proyecto, ya que se busca obtener conocimientos significativos y explicables.

Otra consideración importante es que no se requiere explícitamente la capacidad de no linealidad en el análisis. Dado que el kernel lineal en KPCA obtuvo el mismo rendimiento que PCA, no hay una ventaja clara en utilizar la extensión no lineal en este caso. Al elegir PCA, este estudio se puede beneficiar de su simplicidad y eficiencia computacional en comparación con KPCA.

Algoritmos para abordar valores atípicos

Para realizar experimentos y mejorar la precisión del modelo de predicción, se implementaron tres modelos de Scikit-learn: SVR, RANSACRegressor y HuberRegressor. El objetivo de esta investigación fue abordar el desafío de los valores atípicos en el conjunto de datos. Para ello, se ejecutó cada modelo utilizando tres enfoques diferentes en los datos: los datos originales, los datos normalizados y los datos discretizados. Para cada enfoque y modelo, se ajustó el modelo a los datos de entrenamiento y se realizó predicciones en los datos de prueba. Se calculó el Error Cuadrático Medio (MSE) para evaluar el desempeño de cada modelo y enfoque (Tabla 4).

TABLA 4
Rendimiento de los algoritmos SVR, HUBER Y RANSAC

TIPOS DE DATOS	ALGORITMOS	MSE
Datos originales	SVR	0.0185
	HUBER	0.0477
	RANSAC	0.0526
Datos normalizados	SVR	0.0198
	HUBER	0.0395
	RANSAC	0.3417
Datos discretizados	SVR	0.0216
	HUBER	0.0399
	RANSAC	24.4857

Después de realizar estos experimentos, se evidenció que el modelo SVR con los datos originales mostró el MSE más bajo, lo que sugiere una mejor capacidad de predicción y una mayor eficacia en la detección y manejo de valores atípicos.

Técnicas de Regularización

Para abordar el problema de la multicolinealidad y la selección de características en el conjunto de datos, se manejaron técnicas de regularización aplicadas a modelos de regresión lineal, los resultados se los puede ver en la Tabla 5.

TABLA 5
Resultados aplicando técnicas de regularización

TIPO DE DATOS	ALGORITMOS	RESULTADOS
Datos originales	Lineal	0.8317
	Lasso	0.8023
	Ridge	0.8282
	ElasticNet	0.8016
Datos normalizados	Lineal	0.8317
	Lasso	0.6223
	Ridge	0.8313
	ElasticNet	-6.4228
Datos discretizados	Lineal	0.8114
	Lasso	0.7032
	Ridge D	0.8113
	ElasticNet	0.2845

Después de realizar estos experimentos, se encontró que el modelo lineal es el mejor candidato para abordar el problema de la multicolinealidad. Este modelo obtuvo puntajes altos en todas las versiones de los datos. Esto indica que este modelo tiene un buen rendimiento en diferentes contextos.

El modelo lineal demostró un buen desempeño en términos de predicción, superando a los modelos Ridge, Lasso y ElasticNet en la mayoría de las métricas evaluadas. Sus puntajes fueron consistentemente altos, lo que indica que es capaz de capturar las relaciones entre las variables y hacer predicciones precisas. Los coeficientes del modelo lineal indican la contribución relativa de cada característica para predecir la variable objetivo (Incidencia). Observando los coeficientes del modelo lineal, se notó que, en el caso de los datos originales, la

característica "Rain" tiene el coeficiente más alto, lo que sugiere que puede ser la característica más importante para el modelo en cuestión. En el segundo y tercer caso de normalización y discretización, la característica “Severidad (%)” tiene el mayor peso, lo que sugiere que puede ser la más importante para esos modelos. Estos coeficientes indican que un aumento en estas características tiende a estar asociado con un aumento en la variable objetivo (Incidencia).

Basado en esto, se puede decir que la característica "Rain" tiene el mayor peso en el modelo y es el factor más importante para predecir la variable objetivo (Incidencia) según el modelo lineal.

Variables que tienen mayor peso o influencia en la predicción

TABLA 6
Variable con mayor peso en Datos originales

VARIABLES QUE TIENEN MAYOR PESO EN LA PREDICCIÓN CON DATOS ORIGINALES	
ALGORITMO	COEFICIENTE
Linear	1.22095824e+00
Lasso	1.11271847e-02
Ridge	9.30460714e-01
ElasticNet	0.0109246

*Linea*r

En el modelo de regresión lineal se encontró que la variable más influyente fue "Rain", con un coeficiente de 1.22095824e+00. Este coeficiente indicó que un aumento unitario en la cantidad de precipitación medida en un período de tiempo específico resulta en un incremento de aproximadamente 1.22 unidades en la variable objetivo.

Lasso

El modelo Lasso utiliza la técnica de regularización L1, que penaliza los coeficientes de las variables menos importantes, haciendo que algunos de ellos sean exactamente cero.

En el modelo de regularización Lasso se identificó que la variable más significativa fue "Severidad", con un valor de 1.11271847e-02. Esta variable representó la gravedad del daño causado por la moniliasis en las plantas de cacao.

Ridge

El modelo Ridge utiliza la técnica de regularización L2, que penaliza los coeficientes de las variables menos importantes, haciéndolos cercanos a cero, pero no exactamente cero.

En la técnica de regularización Ridge, nuevamente, se encontró que la variable más importante fue "Rain", al igual que en el modelo lineal con un coeficiente de 9.30460714e-01. Esto significa que la regularización ha disminuido la magnitud del impacto de "Rain" en la variable objetivo en comparación con el modelo lineal, lo que puede ayudar a evitar el sobreajuste.

ElasticNet

ElasticNet combina tanto Lasso como Ridge mediante una combinación lineal de las regularizaciones L1 y L2. En este caso, la variable "Severidad" también se identificó como la más relevante, al igual que en Lasso. El coeficiente asociado a "Severidad" en ElasticNet fue de 0.0109246, lo que indica que tiene un efecto similar al modelo Lasso, pero también se vio afectada por la regularización Ridge.

Con base en los resultados obtenidos con los datos originales, el algoritmo Linear con la variable 'Rain' (precipitación) obtuvo el mayor impacto en la predicción, ya que tuvo el coeficiente más alto de

1.22095824e+00. Esto evidenció que la cantidad de precipitación tuvo una influencia significativa en la variable incidencia.

TABLA 7
Variable con mayor peso en Datos Normalizados

VARIABLES QUE TIENEN MAYOR PESO EN LA PREDICCIÓN CON DATOS NORMALIZADOS	
ALGORITMO	COEFICIENTE
Linear	0.4140
Lasso	0.2219
Ridge	0.4137
ElasticNet	0

Linear

La variable con mayor peso fue "Severidad", que indicó el porcentaje de la moniliasis en la planta de cacao. Su coeficiente fue 0.4140.

Lasso

Al igual que en el modelo "Linear", la variable con mayor influencia fue "Severidad", pero su coeficiente fue 0.2219 siendo menor al coeficiente Linear.

Ridge

Al igual que en los modelos "Linear" y "Lasso", la variable "Severidad" también obtuvo el mayor peso. Sin embargo, su coeficiente fue 0.4137, el cual fue menor al modelo Linear y mayor a Lasso.

ElasticNet

En este caso, todas las variables obtuvieron un valor de coeficiente de 0, lo que significó que ninguna variable tuvo un impacto relevante en la predicción con la técnica de regularización "ElasticNet".

En general, para estos datos normalizados, el modelo "Linear" alcanzó la variable "Severidad" con el mayor peso, seguido por el modelo "Ridge". El modelo "Lasso" tuvo un coeficiente menor para la misma variable, mientras que "ElasticNet" no capturó ninguna influencia significativa de ninguna variable.

TABLA 8
Variable con mayor peso en Datos Discretizados

VARIABLES QUE TIENEN MAYOR PESO EN LA PREDICCIÓN CON DATOS DISCRETIZADOS	
ALGORITMO	COEFICIENTE
Linear	0.2517
Lasso	0.1794
Ridge	0.2517
ElasticNet	0.0529

Linear

La variable más influyente en el modelo Linear fue "Severidad" con un coeficiente de 0.2517. Esto significa que el porcentaje de moniliasis en la planta de cacao tuvo un impacto representativo en la predicción de la enfermedad.

Lasso

Al igual que en el modelo Linear, la variable más importante en el modelo Lasso también fue "Severidad" con un coeficiente ligeramente menor de 0.1794. Aunque su peso fue menor que en el modelo Linear, sigue siendo una característica crucial para la predicción.

Ridge

Una vez más, la variable "Severidad" destacó como la más influyente en el modelo Ridge con un coeficiente similar al del modelo Linear, es decir, 0.2517. Esto sugiere que la severidad de la moniliasis sigue siendo un factor clave en la detección de la enfermedad, incluso con la regularización de Ridge.

ElasticNet

En el modelo ElasticNet, la variable "Severidad" también fue la más importante, pero con el coeficiente más bajo de 0.0529 entre todos los algoritmos. Aunque su influencia fue menor en este modelo, sigue siendo un componente relevante en la predicción de la moniliasis.

La "Severidad" es un factor crítico en la detección y predicción de la moniliasis en plantas de cacao utilizando técnicas de regularización con datos discretizados, en este caso, el algoritmo Ridge y Linear tuvieron la variable y resultado con mayor peso.

TABLA 9

Comparación de variables con mayor peso

DATOS	ALGORITMO	VARIABLE	COEFICIENTE
Originales	Linear	Rain	1.22095824e+00
Normalizados	Linear	Severidad	0.4140
Discretizados	Ridge y Linear	Severidad	0.2517

En los datos originales, la variable cantidad de lluvia ("Rain") tuvo el mayor peso en el modelo Linear, pero al normalizar los datos, la variable "Severidad" se convirtió en la más influyente. En los datos discretizados, la variable "Severidad" mantuvo su importancia en los algoritmos Ridge y Linear, con un coeficiente similar en ambos casos. En conclusión, la variable con mayor peso fue "Rain" con el modelo Linear, utilizando datos originales. Esto indica que la cantidad de lluvia fue un factor significativo en la predicción de la variable objetivo.

Modelos ensamblados basados en bagging y boosting

Para elegir el modelo que permita mejorar el rendimiento y la precisión de la predicción, se realizó pruebas con modelos ensamblados basados en bagging y boosting, los que combinaron técnicas de aprendizaje supervisado con datos sin normalizar y normalizados para determinar la diferencia. Los resultados se pueden visualizar en la Tabla 10 y Tabla 11.

Bagging

TABLA 10
Resultados de modelos ensamblados basados en bagging

TIPOS DE DATOS	MODELO	RESULTADOS
Datos originales	LogisticRegression	1.0
	SVC	0.9846
	LinearSVC	1.0
	SGD	1.0
	KNN	0.9936
	DecisionTreeClf	1.0
	RandomTreeFores	1.0
Datos normalizados	LogisticRegression	0.9820
	SVC	0.9808
	LinearSVC	0.9923
	SGD	0.9923
	KNN	0.9603
	DecisionTreeClf	1.0
	RandomTreeFores	1.0
Datos discretizados	LogisticRegression	0.9923
	SVC	0.9782
	LinearSVC	0.9923
	SGD	0.9884
	KNN	0.9641
	DecisionTreeClf	0.9872
	RandomTreeFores	0.9923

En términos generales, todos los modelos mostraron un rendimiento bastante sólido en los tres escenarios. Algunos modelos destacaron en ciertos aspectos, pero es importante considerar que la elección del mejor modelo depende de las características y requisitos específicos del problema en cuestión.

El modelo LogisticRegression obtuvo una puntuación perfecta de precisión (1.0) en el escenario de datos originales, lo que indica que pudo clasificar correctamente todas las muestras de prueba. Sin embargo, también consiguió un rendimiento muy bueno en los otros dos escenarios, con puntuaciones de precisión superiores al 0.98. Esto sugiere que LogisticRegression es un modelo sólido y confiable en general.

Otros modelos, como SVC, LinearSVC, SGD y RandomTreeForest, también obtuvieron puntuaciones muy altas en los tres escenarios, aunque ligeramente inferiores a las del modelo LogisticRegression. Estos modelos demuestran una capacidad consistente para clasificar correctamente las muestras.

El modelo KNN mostró un rendimiento ligeramente inferior en comparación con los anteriores. Aunque obtuvo puntuaciones de precisión superiores al 0.96 en los tres escenarios, es importante tener en cuenta que KNN se basa en la cercanía de los vecinos, lo que puede resultar en un rendimiento variable dependiendo de los datos y la distribución de las muestras.

Por último, los modelos DecisionTreeClf y RandomTreeForest también mostraron un rendimiento sólido en los datos originales y normalizados, con puntuaciones de precisión perfectas (1.0). Sin embargo, el rendimiento en el escenario de datos discretizados fue ligeramente inferior, lo que indicó que estos modelos pueden no ser tan eficientes al tratar con datos discretizados.

Considerando los resultados obtenidos, el modelo LogisticRegression parece ser el más adecuado en términos de rendimiento general en los tres escenarios evaluados.

Boosting

TABLA 11
Resultados del modelo ensamblado basado en boosting

TIPO DE DATOS	PRECISIÓN	NÚMERO DE ESTIMADORES
Originales	1.0	4
Normalizados	1.0	4
Discretizados	0.9885	4

Los resultados obtenidos revelaron que, en todos los casos, el algoritmo de boosting logró una alta precisión en la clasificación. Tanto los datos originales como los datos normalizados alcanzaron una precisión perfecta del 100% con un número de estimadores igual a 4. Esto indica que el modelo fue capaz de aprender eficientemente y realizar una clasificación precisa utilizando cualquiera de los dos conjuntos de datos.

Por otro lado, los datos discretizados también ofrecieron un rendimiento muy sólido, con una precisión cercana al 98.85%. Aunque ligeramente inferior a los otros dos conjuntos de datos, sigue siendo un resultado muy satisfactorio. Estos resultados sugirieron que el algoritmo de boosting utilizado fue robusto y capaz de manejar diferentes tipos de datos. Tanto los datos originales como los datos normalizados demostraron ser igualmente efectivos, mientras que la discretización de los datos introdujo una leve disminución en el rendimiento, pero aún ofreció una precisión destacable.

Con base en los hallazgos presentados, se puede concluir que tanto los datos originales como los datos normalizados alcanzaron un rendimiento excelente con una precisión del 100%. Dado que no hubo una diferencia significativa entre estos dos conjuntos de datos en términos de rendimiento, se puede elegir cualquiera de ellos para entrenar el modelo de boosting.

Resultados

Después de revisar los resultados y considerar los puntajes obtenidos por diferentes algoritmos, he llegado a la conclusión de que LogisticRegression es el mejor algoritmo en comparación de los demás algoritmos de aprendizaje supervisado, ya que con los datos de prueba obtuvo el mejor resultado. Este modelo ha demostrado un desempeño sobresaliente al obtener un puntaje perfecto de 1.0 en los datos originales utilizados, lo que indica que LogisticRegression ha logrado un ajuste óptimo a los datos originales y puede realizar predicciones precisas en ese conjunto de datos específico. Esto sugiere que el modelo ha capturado de manera efectiva los patrones y las relaciones presentes en los datos originales.

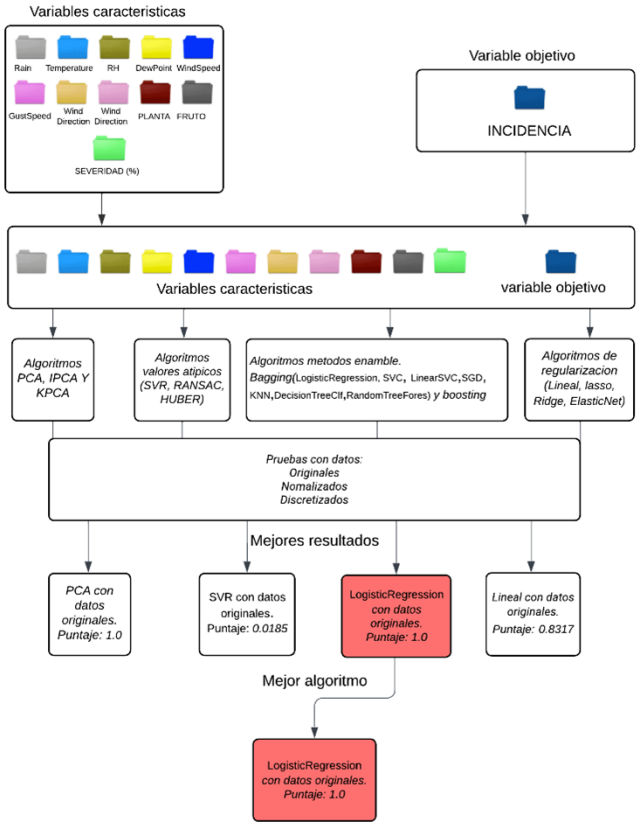

FIGURA 3
Esquema de elección del algoritmo de aprendizaje supervisado que obtuvo el mejor resultado

Validación de modelos

Se ha validado el modelo entrenado para predecir la moniliasis utilizando tres enfoques diferentes de validación cruzada: k-folds, LOOCV y Hold-Out, así también, se corroboró el modelo con un conjunto de datos diferente que no fue utilizado para entrenar el modelo y las predicciones fueron igualmente precisas.Los resultados se muestran a continuación.

K-Fold Cross-Validation

La Validación Cruzada K-Fold (K-Fold Cross-Validation) es una técnica de evaluación de modelos de aprendizaje automático ampliamente utilizada para medir la capacidad de generalización de un modelo en un conjunto de datos. Su objetivo principal es obtener una estimación más precisa del rendimiento del modelo al usar los datos de manera más eficiente.

El procedimiento de K-Fold Cross-Validation consiste en dividir el conjunto de datos en "k" partes o subconjuntos (folds), aproximadamente, iguales. Luego, el modelo se entrena y evalúa "k" veces, donde en cada iteración se utiliza una partición diferente como conjunto de prueba, y las restantes se emplean como conjunto de entrenamiento. Esto asegura que cada instancia del conjunto de datos sea utilizada tanto para entrenar como para evaluar el modelo.

En cada iteración, se registran las métricas de rendimiento, como el Error Cuadrático Medio (MSE), precisión, recall, entre otras, para evaluar el rendimiento del modelo en cada conjunto de prueba.

Por último, se calcula el promedio de las métricas de rendimiento obtenidas en las "k" iteraciones para obtener una estimación general del rendimiento del modelo.

Análisis del Resultado de K-Fold Cross-Validation

Se aplicó K-Fold Cross-Validation con "k=3" particiones para evaluar el modelo de regresión previamente entrenado. Los resultados muestran tres valores de MSE para cada iteración de K-Fold, y se obtuvo un MSE promedio de 0.0.

El MSE promedio de 0.0 indicó una coincidencia perfecta entre las predicciones del modelo y los valores reales en todos los conjuntos de prueba utilizados en la validación cruzada.

Leave-One-Out Cross-Validation

LOOCV (Leave-One-Out Cross-Validation) es una técnica de validación cruzada que se utiliza para evaluar el rendimiento de un modelo estadístico o de aprendizaje automático. Su objetivo es estimar cómo se comportará el modelo en datos no vistos y comprobar su capacidad para generalizar a nuevos datos.

El funcionamiento de LOOCV es relativamente sencillo. En primer lugar, se toma el conjunto de datos original y se divide en dos partes: un punto de datos individual (una muestra) se separa para ser utilizado como conjunto de validación, mientras que el resto de los datos forman el conjunto de entrenamiento. El modelo se entrena usando el conjunto de entrenamiento y luego se evalúa su rendimiento manejando el punto de datos de validación único que se dejó fuera previamente.

Este proceso de entrenamiento y evaluación se repite para cada punto de datos en el conjunto original, dejando uno diferente fuera en cada iteración. Por lo tanto, si el conjunto de datos original tiene N puntos, se realizarán N iteraciones en total. Al finalizar, se promedian los resultados de evaluación obtenidos en cada iteración para obtener una medida de rendimiento general del modelo.

Análisis del Resultado de Leave-One-Out Cross-Validation

Se aplicó LOOCV para evaluar el modelo de regresión previamente entrenado. Los valores de MSE resultantes fueron todos cero. Esto significa que el error cuadrático medio (MSE) obtenido, utilizando la técnica de validación cruzada LOOCV, fue cero para todos los datos de prueba, esto indicó que el modelo pudo ajustarse perfectamente a los datos de entrenamiento y pudo hacerse predicciones precisas para los datos de prueba.

Tras aplicar el método de validación Hold-Out al modelo aquí propuesto, se obtuvo un error cuadrático medio (MSE) de cero. Esto mostró que las predicciones del modelo son perfectamente precisas y no hay diferencia entre los valores reales y los valores predichos. Además, se validó con un conjunto de datos diferente que no fue empleado para entrenar el modelo y las predicciones fueron igualmente precisas. Esto sugiere que el modelo planteado ha capturado bien las relaciones subyacentes en los datos y puede generalizar bien si se aplica a nuevos datos.

Hold-Out Validation

La validación Hold-Out es una técnica de evaluación de modelos de aprendizaje supervisado que consiste en dividir el conjunto de datos en dos subconjuntos disjuntos: un conjunto de entrenamiento y un conjunto de prueba. El de entrenamiento se utiliza para entrenar el modelo, mientras que el conjunto de prueba se reserva

exclusivamente para evaluar su rendimiento de manera independiente. Es decir, el modelo no ha visto los datos del conjunto de prueba durante su proceso de entrenamiento, lo que permite obtener una estimación más objetiva de su capacidad para generalizar a datos no vistos previamente.

El funcionamiento de la validación Hold-Out consiste en dividir el conjunto de datos en dos partes mutuamente excluyentes: el conjunto de entrenamiento y el conjunto de prueba. El conjunto de entrenamiento se utiliza para entrenar el modelo, ajustando sus parámetros y aprendiendo patrones en los datos. Posteriormente, el modelo se evalúa con el conjunto de prueba, que contiene datos no vistos durante el entrenamiento, para medir su capacidad de generalización y su rendimiento en nuevas instancias. Esta técnica proporciona una estimación inicial del desempeño del modelo y permite detectar problemas como el ajuste excesivo (overfitting). Aunque la validación Hold-Out es sencilla y rápida, su representatividad puede depender del tamaño del conjunto de prueba y, por tanto, es aconsejable combinarla con otras técnicas, como la validación cruzada, para obtener una evaluación más robusta del modelo.

Análisis del Resultado de Hold-Out Validation

Tras aplicar el método de validación Hold-Out al modelo, se obtuvo un error cuadrático medio (MSE) de cero. Esto indica que las predicciones del modelo fueron precisas y no hay diferencia entre los valores reales y los predichos.

Comparativa entre métodos de validación

Según los resultados, todos los métodos de validación utilizados (k-folds, LOOCV y Hold-Out) dieron un MSE promedio de 0.0, como se ve en la Tabla 12. Esto indica que las predicciones del modelo son precisas y no hay diferencia entre los valores reales y los valores predichos en ninguno de los métodos de validación utilizados.

TABLA 12
Resultados de métodos de Validación

MÉTODO DE VALIDACIÓN	MSE PROMEDIO
k-folds	0.0
LOOCV	0.0
Hold-Out	0.0

El modelo muestra un alto nivel de precisión en la tarea de predicción, independientemente del método de validación utilizado. Esto sugiere que el modelo ha capturado bien las relaciones subyacentes en los datos y puede generalizar cuando se aplica a nuevos datos.

Optimización paramétrica

Se ha realizado una optimización paramétrica del modelo propuesto utilizando tres enfoques diferentes: manual, grilla y búsqueda aleatoria. Estos son métodos comunes para ajustar los parámetros de un modelo y mejorar su rendimiento.

Los resultados se muestran a continuación.

Optimización manual

Este enfoque implica ajustar manualmente los parámetros del modelo y evaluar su rendimiento. Este proceso se repite hasta encontrar una combinación de parámetros que proporcione el mejor rendimiento.

Tras aplicar una optimización manual de los parámetros del modelo de regresión de bosques aleatorios, se encontró que la mejor combinación de parámetros fue n_estimators=4, criterion= 'squared_error' y max_depth=2, como se muestra en la Tabla 13.

TABLA 13
Resultados de la optimización manual

PARÁMETRO	MEJOR VALOR
n_estimators	4
criterion	squared_error
max_depth	2

Esto significa que el mejor modelo encontrado tuvo 4 árboles, utiliza el error cuadrático como criterio para medir la calidad de las divisiones y tuvo una profundidad máxima de 2.

Optimización por grilla

Este enfoque implica definir un conjunto de valores posibles para cada parámetro y evaluar el rendimiento del modelo para todas las combinaciones posibles de parámetros. La combinación de parámetros que proporcione el mejor rendimiento se selecciona como la mejor.

Tras aplicar una búsqueda en grilla para optimizar los parámetros del modelo de regresión de bosques aleatorios, se detectó que la mejor combinación de parámetros fue n_estimators=4, criterion= 'squared_error' y max_depth=2 como se evidencia en la Tabla 14.

TABLA 14
Resultados de la optimización por grilla

PARÁMETRO	MEJOR VALOR
n_estimators	4
criterion	squared_error
max_depth	2

Esto significa que el mejor modelo encontrado tuvo 13 árboles, utiliza el error absoluto como criterio para medir la calidad de las divisiones y tuvo una profundidad máxima de 9.

Optimización por Búsqueda aleatoria

Este enfoque implica muestrear aleatoriamente combinaciones de parámetros y evaluar el rendimiento del modelo para cada combinación. La combinación de parámetros que proporcione el mejor rendimiento se selecciona como la mejor.

Tras aplicar una búsqueda aleatoria para optimizar los parámetros del modelo de regresión de bosques aleatorios, encontramos que la mejor combinación de parámetros fue n_estimators=13, criterion= 'absolute_error' y max_depth=9 como se oberva en la Tabla 15.

TABLA 15
Resultados de optimización por Búsqueda aleatoria

PARÁMETRO	MEJOR VALOR
n_estimators	13
criterion	absolute_error
max_depth	9

Esto significa que el mejor modelo encontrado tuvo 13 árboles, utiliza el error absoluto como criterio para medir la calidad de las divisiones y tuvo una profundidad máxima de 9.

Comparativa entre métodos de optimización

Se ha aplicado diferentes métodos de optimización de parámetros para el modelo de regresión de bosques aleatorios propuesto y se encontró diferentes combinaciones óptimas de parámetros dependiendo del método utilizado. Después de aplicar una optimización manual y una búsqueda en grilla, la mejor combinación de parámetros encontrada fue n_estimators=4, criterion= 'squared_error' y max_depth=2. Por otro lado, después de aplicar una búsqueda aleatoria, la mejor combinación de parámetros encontrada fue n_estimators=13, criterion= 'absolute_error' y max_depth=9 (Tabla 16).

TABLA 16
Comparativa entre métodos de optimización

MÉTODO DE OPTIMIZACIÓN	N_ESTIMATORS	CRITERION	MAX_DEPTH
Manual	4	squared_error	2
Grilla	4	squared_error	2
Búsqueda aleatoria	13	absolute_error	9

Estos resultados muestran que diferentes métodos de optimización pueden llevar a diferentes combinaciones óptimas de parámetros para este modelo.

Implementación del modelo predictivo en la PWA

Una vez que se validó y se comprobó que el modelo predictivo da buenos resultados, se implementó el modelo en la PWA, el cual, quedó de la siguiente manera:

En la pantalla de inicio, los elementos que se encuentran son: una barra de navegación en la parte superior, en el lado izquierdo, se destaca una imagen de un árbol de cacao. Justo del lado derecho, se muestra un título grande que dice "Moniliasis" y un subtítulo que indica "Enfermedad del cacao", y más abajo, un párrafo que ofrece información importante sobre la moniliasis y su impacto en el cultivo de cacao. En la parte de abajo del contenido principal, hay dos botones, uno que ofrecen la posibilidad de acceder a ver la información sobre

los datos de los sensores, y otro que dirige a la página para que el usuario pueda predecir la moniliasis en su planta de cacao (Figura 4).

FIGURA 4
Pantalla de inicio de la PWA

En la página para ver lo datos de los sensores, los datos que se muestran son extraídos de la base de datos de MongoDB mediante una API.

El contenido principal está dividido en dos columnas, en la columna izquierda, hay un recuadro rectangular que muestra la temperatura, en la columna derecha, hay varios recuadros, cada uno con un título descriptivo, un ícono correspondiente y un campo que muestra datos de la lluvia, humedad relativa, punto de rocío, velocidad y dirección del viento, así como velocidad de ráfaga. Además, se almacenan localmente los datos obtenidos para acceder a ellos cuando el dispositivo esté fuera de línea. Si el dispositivo cambia de estado de fuera de línea a en línea, se vuelven a cargar los datos desde la API, Figura 5.

FIGURA 5
Página de la PWA que muestra los datos de los sensores

En la página para predecir la moniliasis, el usuario podrá ingresar datos de una planta, como el identificador de la planta, los fruto y la severidad. El contenido se divide en dos columnas. En la columna izquierda hay un rectángulo que muestra un título "Ingrese los datos de la planta" seguido de un formulario con tres campos de entrada para los datos mencionados anteriormente. El usuario puede completar estos campos con la información deseada. Luego, hay un botón "Enviar" que, cuando se hace clic, realiza una solicitud POST al servidor para procesar los datos ingresados y mostrar la predicción echa en la columna derecha. En la columna derecha se muestra la predicción como tal, Figura 6.

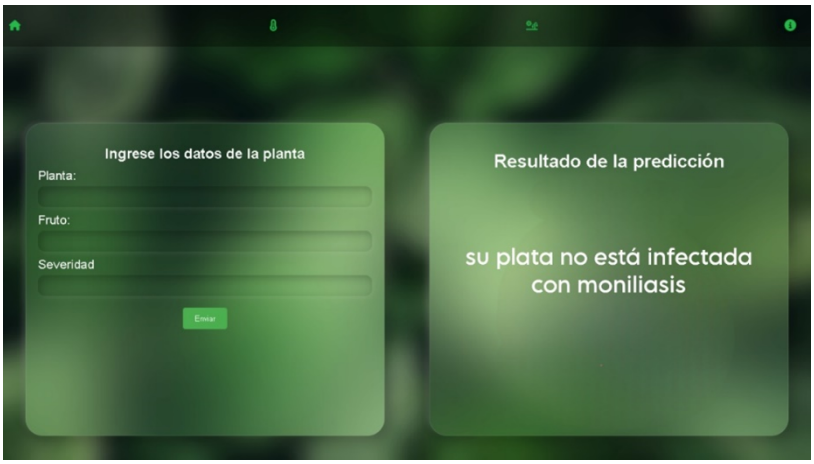

FIGURA 6
Página para predecir la moniliasis

Discusión

La presente investigación se centró en el desarrollo de un modelo predictivo para detectar la moniliasis en plantas de cacao en la Provincia de Orellana. Los resultados obtenidos destacan la importancia del uso de técnicas de aprendizaje supervisado en la detección de esta enfermedad fúngica, con el objetivo de reducir las pérdidas en los cultivos de cacao.

En primer lugar, los resultados de esta investigación demuestran que el modelo desarrollado presenta una precisión significativa en la predicción de la presencia de moniliasis en las plantas de cacao. Esta capacidad predictiva resulta fundamental para los agricultores, ya que les permite tomar medidas preventivas y aplicar tratamientos específicos en etapas tempranas, contribuyendo así a reducir la propagación de la enfermedad y minimizar las pérdidas.

Asimismo, la recopilación de datos detallados sobre las características de las plantas de cacao y las condiciones ambientales, tanto a través de sensores como de registros manuales, resultó de vital importancia para entrenar el modelo presentado de manera efectiva. Estos hallazgos enfatizan la necesidad de obtener información precisa y completa, a fin de mejorar la precisión de los modelos predictivos en el ámbito agrícola.

No obstante, es importante reconocer ciertas limitaciones del estudio. Por ejemplo, la disponibilidad de datos históricos sobre la moniliasis fue limitada, lo que pudo haber afectado la capacidad del modelo para capturar toda la variabilidad de la enfermedad. Además, es importante destacar que la investigación se enfocó específicamente en la Provincia de Orellana, por lo que, los resultados podrían no ser generalizables a otras regiones con diferentes condiciones climáticas y de cultivo.

Conclusiones

Este estudio ha demostrado la eficacia de un modelo predictivo basado en aprendizaje supervisado para detectar la moniliasis en plantas de cacao en la Provincia de Orellana. Los resultados obtenidos resaltan la importancia de utilizar herramientas de análisis de datos en el campo agrícola, especialmente en la detección temprana de enfermedades que pueden afectar la producción y calidad de los cultivos.

Se identificó la importancia de recopilar datos detallados sobre las características de las plantas de cacao y las condiciones ambientales para mejorar la precisión del modelo predictivo. Esto resalta la necesidad de contar con información precisa y completa, obtenida a través de sensores y registros manuales, para desarrollar modelos más efectivos en el futuro.

Es importante tener en cuenta algunas limitaciones de este estudio. La disponibilidad de datos históricos sobre la moniliasis fue escasa, lo que podría haber afectado la capacidad del modelo para capturar toda la variabilidad de la enfermedad. Además, los resultados se limitan a la Provincia de Orellana y pueden no ser generalizables a otras regiones con diferentes condiciones climáticas y de cultivo.

Reconocimientos

Los autores desean expresar su agradecimiento a la Escuela Superior Politécnica de Chimborazo, de igual manera a nuestros docentes, el distinguido Wilson Gustavo Chango Sailema, Ph.D. y al Ing. Pedro Stalyn Aguilar Encarnación. Gracias por su colaboración en este estudio.

Referencias

Bernardi, L., Branco da Motta, & Bernardi Lucioana, C. (2018). Development of an app as a tool to support research and the prevention of osteoporosis. *Revista Brasileira de Geriatria e Gerontologia*. https://doi.org/10.1590/1981-22562018021.170189

Boersma, S., & lungu, mircea. (2021). React-bratus: visualización de jerarquías de componentes de React. *IEEE*.

Caicedo, C. (2019). *Primer Simposio Internacional Innovaciones Tecnológicas para Fortalecer la Cadena de Cacao en la*.

Carrera, K., Mosquera, L., & Leiva, M. (2014). Protocolo para el aislamiento de Moniliophthora roreri (Cif y Par)Evans et al. en frutos de cacao cv. 'Nacional' de la Amazoníaecuatoriana. *Biotecnología Vegetal , 14*.

Correa, J., Castro, S., & Coy, J. (2014). Estado de la moniliasis del cacao causada por Moniliophthora roreri en Colombia . *Sistema de Información Científica Redalyc*.

Fernández, T., Fernández Leonardo, Ricciardi, T., Ugarte, L., & Almeida, M. (2018). Lenguaje de programación Python para el análisis de sistemas de potencia Educación e investigación. *IEEE*.

Fortunato, D., & Bernardino, Jorge. (2018). Aplicaciones web progresivas: una alternativa a las aplicaciones móviles nativas. *IEEE*.

Gramajo, M. G., Ballejos, L., & Ale, M. (2020). Seizing Requirements Engineering Issues through Supervised Learning Techniques. *IEEE Latin America Transactions, 18*(7), 1164–1184. https://doi.org/10.1109/TLA.2020.9099757

Jha, K., Doshi, A., Patel, P., & Shah, M. (2019). A comprehensive review on automation in agriculture using artificial intelligence. In *Artificial Intelligence in Agriculture* (Vol. 2, pp. 1–12). KeAi Communications Co. https://doi.org/10.1016/j.aiia.2019.05.004

Leandro-Muñoz, M. E., Tixier, P., Germon, A., Rakotobe, V., Phillips-Mora, W., Maximova, S., & Avelino, J. (2017). Effects of microclimatic variables on the symptoms and signs onset of Moniliophthora roreri, causal agent of Moniliophthora pod rot in cacao. *PLoS ONE, 12*(10). https://doi.org/10.1371/JOURNAL.PONE.0184638

Oliveira, D., Barbosa, U., CRO Bergland, A., & Resende, O. (2022). G-SOJA - SITIO WEB CON PREDICCIÓN DE LA CLASIFICACIÓN DE LA SOJA UTILIZANDO MACHINE LEARNING. *IEEE*.

Ovalle, C. (2022). Modelo predictivo basado en Machine Learning para la Cadena de Suministro y su influencia en la gestión logística de una empresa de venta de autos. *Journal of the ACM ER* .

Ricardez, D. la C., Espinoza, L., García, O., & Pérez, P. (2016). ACTIVIDAD ANTIFÚNGICA in vitro DEL EXTRACTO ACUOSO Y ALCALOIDEO DE Lupinus spp. SOBRE Moniliophthora rorer. *Agroproductividad*.

Robles, S., Vásquez, H., & Naranjo, L. (2019). *Vista de Adaptación de la metodología de ciencia de diseño en el desarrollo de luminarias | Tecnología Vital*. https://revistas.ulatina.ac.cr/index.php/tecnologiavital/article/view/252/265

Susanto Stiawan, D., Arifin, M. A. S., Idris, M. Y., & Budiarto, R. (2020). Iot botnet malware classification using weka tool and scikit-learn machine learning. *International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), 2020-October*, 15–20. https://doi.org/10.23919/EECSI50503.2020.9251304

Susilo, A., Karna, N., & Mayasari, R. (2021). Decision Tree-Based Bok Choy Growth Prediction Model for Smart Farm. *2021 4th International Conference on Information and Communications Technology (ICOIACT)*, 169–174. https://doi.org/10.1109/ICOIACT53268.2021.9563914