https://repository.uaeh.edu.mx/revistas/index.php/sahagun/issue/archive

Ingenio y Conciencia Boletín Científico de la Escuela Superior Ciudad Sahagún

Publicación semestral, Vol. 12, No. Especial (2025) 56-67

ISSN: 2007-784X

Implementación de modelos de aprendizaje automático para el mantenimiento predictivo y clasificación de fallas de maquinaria industrial

Implementation of machine learning models for predictive maintenance and fault classification of industrial machinery

Karla I. Romero Magdaleno [a], Luis C. Méndez González [b], Isidro J. González Hernández [c], Abel E. Quezada Carreón [d]

Abstract:

The objective of this work was to use a dataset to show how to implement a mathematical model of predictive maintenance using machine learning techniques to predict the value of "Target" (failure or non-failure) of industrial machinery, using supervised learning tools, specifically, Random Forest or Random Forest and Naive Bayes, applied to classification problems of the data obtained from the Kaggle platform and performing a comparison between the two. The article will include data preprocessing, model training, and measuring model performance. The results indicate that AI can help in the early detection of faults or breakdowns and improve maintenance operations.

Keywords:

Predictive maintenance, artificial intelligence, machine learning, Random Forest, Naive Bayes

Resumen:

El objetivo del presente trabajo fue utilizar un conjunto de datos para mostrar como implementar un modelo matemático de mantenimiento predictivo empleando técnicas de aprendizaje automático para predecir el valor de “Target” (falla o no falla) de maquinaria industrial, utilizando herramientas de aprendizaje supervisado, específicamente, Bosque Aleatorio o Random Forest y Naive Bayes, aplicados a problemas de clasificación de los datos obtenidos de la plataforma Kaggle y realizando una comparación entre ambos. El articulo incluirá el preprocesamiento de datos, entrenamiento del modelo y la medición del rendimiento del modelo. Los resultados indican que la IA puede ayudar a detección anticipada de fallas o averías y mejorar las operaciones de mantenimiento.

Palabras Clave:

Mantenimiento predictivo, inteligencia artificial, aprendizaje automático, Random Forest, Naive Bayes

Introducción

El mantenimiento a equipos ha tenido gran relevancia en la industria debido a los diferentes métodos para la realización de esto y los nuevos sistemas que se han introducido que pueden ser de alta complejidad de fabricación, lo que implica que el mantenimiento tenga una mayor dificultad y tiempo de ejecución tardío. Existen diferentes tipos de mantenimientos, de los cuales los mas relevantes son: el mantenimiento reactivo, que se basa

[a] Universidad Autónoma de Ciudad Juárez | Instituto de Ingeniería y Tecnología, | Ciudad Juárez, Chihuahua | México, https://orcid.org/0009-0009-4709-3420, Email: al179943@alumnos.uacj.mx

[*b] Autor de Correspondencia, Universidad Autónoma de Ciudad Juárez | Instituto de Ingeniería y Tecnología | Ciudad Juárez, Chihuahua | México, https://orcid.org/0000-0002-2533-0036, Email: luis.mendez@uacj.mx

[c] Universidad Autónoma del Estado de Hidalgo| Escuela Superior de Ciudad Sahagún | Ciudad Sahagún, Hidalgo | México, https://orcid.org/0000-0003-2805-6674, Email: igonzalez@uaeh.edu.mx

[d] Universidad Autónoma de Ciudad Juárez | Instituto de Ingeniería y Tecnología | Ciudad Juárez, Chihuahua | México, https://orcid.org/0000-0001-6971-2848, Email: abquezad@uacj.mx

Fecha de recepción: 29/04/2025, Fecha de aceptación: 09/06/2025, Fecha de publicación: 11/07/2025

DOI: https://doi.org/10.29057/escs.v12iEspecial.15032

en reparar o reemplazar cuando ya existe la falla; el mantenimiento preventivo; que programa tareas planificadas para prevenir fallas, basándose en intervalos de tiempo y uso; y, por último, el mantenimiento predictivo, en el cual nos vamos a enfocar. Este mantenimiento utiliza tecnologías avanzadas para el análisis de un conjunto de datos obtenidos de múltiples sensores para predecir cuándo es posible que exista una falla o avería. (Ucar, 2024)
En este articulo nos centraremos en técnicas básicas para la realización de un mantenimiento predictivo adecuado, basándonos en las condiciones del equipo.
El mantenimiento predictivo se basa en algoritmos de aprendizaje automático que analizan el historial de datos del equipo en su función normal para predecir comportamientos futuros.

Un mantenimiento convencional tiende a generar un aumento en el costo operativo y gastos totales de fabricación, por lo tanto, se destaca los beneficios del mantenimiento predictivo que traen consigo reducción del tiempo de inactividad, reducción de costos y mejora de calidad del producto. (Lazaroiu, 2023).

Desde el punto de vista Mecatrónico, los mantenimientos predictivos, pueden de ser de gran ayuda para los sistemas mecatrónicos. La mecatrónica integra mecánica, electrónica, control e informática para diseñar sistemas complejos. Estos sistemas tienen componentes que sufren desgaste, es por eso, que un mantenimiento predictivo es crítico para sistemas mecatrónicos para anticipar fallas antes de que ocurran.

Revisión de Literatura

El mantenimiento predictivo se ha posicionado como una de las áreas mas prometedoras dentro de la industria, gracias a los avances en técnicas de inteligencia artificial y aprendizaje automático.

Se han realizado estudios, para demostrar mejoras en la capacidad de detectar fallas y optimizar el mantenimiento de maquinaria utilizando modelos de aprendizaje automático.

Uno de los estudios más relevantes en la detección de fallas mediante aprendizaje automático es el propuesto en el artículo (Technology, 2023), donde, se emplean técnicas de clasificación multiclase para identificar diferentes tipos de fallas en los rodamientos de maquinarias industriales, utilizando Support Vector Machines y Random Forest donde los autores lograron una precisión de 96.9%, con un puntaje F1 de 98.01%, lo que representa la efectividad del modelo para la clasificación de fallas.

En otro estudio significativo, se presenta una mejora en la clasificación de fallas utilizando AutoML PyCaret y AutoDNN con AutoKeras, donde, se automatiza el proceso de selección de mejor modelo, lo que optimiza la clasificación de fallas sin intervención humana extensa. A pesar de no contar con porcentajes exactos, los resultados obtenidos demuestran una mejora en la precisión y la capacidad de generalización de los modelos (Gupta P. K., 2024).

En otro estudio, se propone un modelo de clasificación binaria de fallas utilizando redes neuronales profundas. Este enfoque se aplica a sistemas de maquinaria pesada para predecir fallas antes de que ocurran, utilizando señales de vibración. El modelo alcanzo una exactitud de 99%, con un rendimiento notable bajo en falsos positivos y negativos (Wu, 2024).

Otro de los estudios más relevantes, se propone una técnica innovadora denominada Logistic-ELM (Extreme Learning Machine) combinada con Mapeo logístico para la detección de fallas en rodamientos. Este modelo alcanzo una precisión del 100% en la clasificación de fallas a partir de 7 subconjuntos de datos diferentes, demostrando la efectividad del enfoque para detectar fallas de manera precisa en diversas condiciones operativas (Chen, 2024).

El estudio de Gupta, Sharma y Yadav (Gupta V. S., 2025) se centra en el mantenimiento predictivo de vehículos blindados utilizando una combinación de técnicas de ensamble como LigthGBM, Random Forest y Gradient Boosting. Estos modelos fueron aplicados en un entorno industrial especifico y demostraron una precisión sobresaliente del 99.8% con un Recall del 99.03%.

El estudio final, analiza el uso de técnicas avanzadas como el Long Short-Term Memory (LSTM) para predecir fallas en industrias de manufactura. Este enfoque se comparó con modelos tradicionales como Support Vector Machines, Random Forest, Regresión Logística y una Red Neuronal hibrida basada en CNN-LSTM, destacando la capacidad del LSTM para manejar secuencias temporales con una precisión de 93% y una Accuracy del 94% (Maheshwari, 2024).

Antecedentes

La aplicación de algoritmos de aprendizaje automatizado ha demostrado una mejora significativa para la toma de decisiones con respecto al mantenimiento, reduciendo costos y tiempos de inactividad (Smith, 2024). A pesar de traer consigo estos beneficios para aumentar la competitividad de las empresas mediante la anticipación de fallas en los equipos, el mantenimiento predictivo requiere de ciertas habilidades y conocimientos para ser

implementado, (Liao, 2023) identifican factores críticos o desafíos a la hora de implementar un sistema de mantenimiento predictivo. Uno de los principales desafíos, es la integración de los datos heterogéneos y el desarrollo de sistemas de monitoreo que combinen tanto métodos basados en datos, como enfoques basados en el conocimiento (Garcia, 2023).

Para que un sistema de mantenimiento predictivo sea más capaz de predecir fallas antes de que ocurran, (Darshan, 2024) implemento una plataforma de Tecnologías de Internet de las Cosas para recopilar datos en tiempo real y utilizados para para entrenar modelos de algoritmos de clasificación, donde se alcanzó una precisión del 92%. Para soluciones más específicas como las aplicadas a generadores diésel, (Ahmed, 2023) ha demostrado la eficacia del monitoreo de condiciones habilitado por la plataforma de internet de las cosas para la predicción en tiempo real con un sistema de soporte de decisiones basado en lógica difusa. Con una nueva revolución industrial y la evolución del mantenimiento predictivo, se integra nuevas tecnologías para tener una mejor predicción de una falla (Kumar, 2024). Un estudio de sectores demostró que son escazas las industrias que obtienen los beneficios que ofrece el mantenimiento predictivo y la integración de nuevas aplicaciones o técnicas, donde, la industria automotriz y aeroespacial tienen un nivel alto de madurez tecnológica (Lopez, 2024). Esto refleja la importancia de las estrategias de mantenimiento predictivo adaptativas y sostenible en la era digital.

Método

Para obtener el modelo de clasificación y predicción con el que se desea alcanzar una buena precisión, se estará utilizando un conjunto de datos extraído de Kaggle (Shivam, 2019).

Para el desarrollo del modelo, no podemos utilizar los datos tal como fueron extraídos, son necesarias ciertas técnicas para adaptar los datos y que sea mas sencillo utilizarlos en el modelo (Molina Salgado, 2024).

Se seguirán las fases de la Figura 1 para el desarrollo.

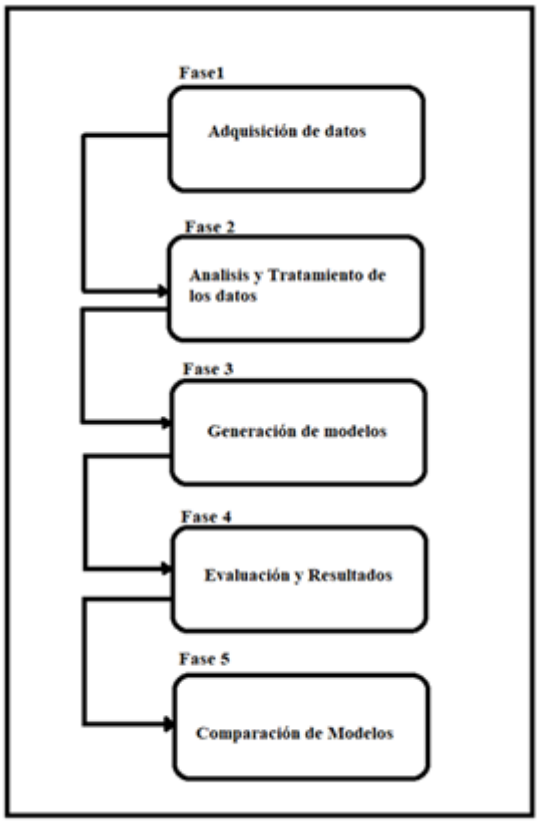


Figura 1. Fases de desarrollo del modelo. Fuente: Elaboración propia.

Fase 1: Adquisición de Datos

En esta fase se analizan el conjunto de datos obtenidos de Kaggle (Shivam, 2019). El conjunto de datos consta de una tabla de 14 columnas, cada una de ellas con una característica de la maquina y 10000 puntos de datos almacenados. Además, se incluye una etiqueta de "fallo de la máquina" que indica si la máquina ha fallado en ese punto de datos específico para cualquiera de los modos de fallo. Dado que los conjuntos de datos suelen ser difíciles de obtener se proporcionan datos ficticios que representan el mantenimiento preventivo real que se encuentra en la industria.

Fase 2: Análisis y Tratamiento de Datos

Durante esta fase se transformarán los datos a un formato adecuado para el análisis y entrenamiento del modelo.

Se realizará una codificación de variables categóricas ya que algunas columnas del conjunto de datos tienen valores de tipo texto, el tipo de dato que utilizan los algoritmos de aprendizaje automático son de tipo

numérico. En la programación en Python se utiliza LabelEncoder para el cambio de etiqueta.

Seguido de esto, se realiza la limpieza de nombres de columna, esto para saber si existe un carácter o símbolo especial, espacios o guiones en cada nombre de columna del conjunto de datos, reemplazándolos por guiones bajos para evitar que algunas funciones fallen.

A continuación, se separa el conjunto de datos en dos partes, las columnas que se usaran para predecir y las columnas que se quieren predecir para que el modelo aprenda cuales son las salidas y entradas para predecir y evitar sobreajuste o resultados incoherentes.

Posteriormente se dividen los datos en 80% entrenamientos y 20% prueba, esto permite evaluar el modelo y evitan que el modelo memorice los datos generando un sobreajuste.

Fase 3: Generación de modelos

Se crearon modelos utilizando algoritmos de clasificación como el Ramdon Forest y Naive Bayes.

El algoritmo Ramdon Forest o Bosque Aleatorio es una técnica de aprendizaje automático basado en arboles de decisión, cada árbol se entrena con una muestra de conjunto de datos y los resultados se combinan para tener un resultado superior. Esta técnica reduce el sobreajuste y muy útil en problemas de clasificación y regresión (Labañino Urbina, 2019).

El algoritmo Naive Baye, si bien también es un algoritmo de clasificación, este asume que las variables predictoras son independientes. Se calcula la probabilidad de cada clase de una entrada y asigna la clase con la probabilidad mas alta, permitiendo una implementación eficiente con menos datos de entrenamiento para estimar los parámetros necesarios. (Zhang, 2004).

Como anteriormente se menciona se utiliza el 80% de entrenamiento y el 20% de prueba.

La experimentación se llevo a cabo en lenguaje de Python en la plataforma de Google Colab para ambos algoritmos de clasificación.

Tabla 1. Estadísticas del conjunto de datos y tipo de variables. Fuente: Elaboración propia.

Estadísticas del Conjunto de Datos

Numero de Variables	10
Numero de observaciones	10000
Celdas faltantes	0
Celdas faltantes (%)	0.00%
Filas Duplicadas	0
Filas Duplicadas (%)	0.00%
Tamaño total en la memoria	781.4 KiB
Tamaño promedio de registro en memoria	80.0 B

Tipo de Variables

Numérica	8
Categórica	2

El conjunto de datos contine las siguientes variables:

Tabla 2. Variable UDI. Fuente: Elaboración propia.

UDI

Uniform Unique

Real number (ℝ)

Distinct	10000
Distinct (%)	100.00%
Missing	0
Missing (%)	0.00%
Infinite	0
Infinite (%)	0.00%
Mean	5000.5
Minimum	1
Maximum	10000
Zeros	0
Zeros (%)	0.00%
Negative	0
Negative (%)	0.00%
Memory size	78.3 KiB

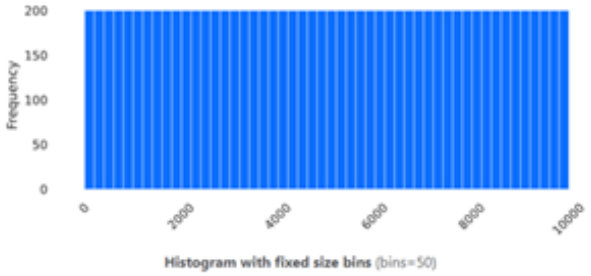


Figura 2. Histograma de UDI. Fuente: Elaboración propia.

***Tabla 3**. Variable Product ID. Fuente: Elaboración propia.*

Product_ID	
Real number (ℝ)	
High correlation	**Uniform** **Unique**
Distinct	10000
Distinct (%)	100.00%
Missing	0
Missing (%)	0.00%
Infinite	0
Infinite (%)	0.00%
Mean	4999.5
Minimum	0
Maximum	9999
Zeros	1
Zeros (%)	< 0.1%
Negative	0
Negative (%)	0.00%
Memory size	78.3 KiB

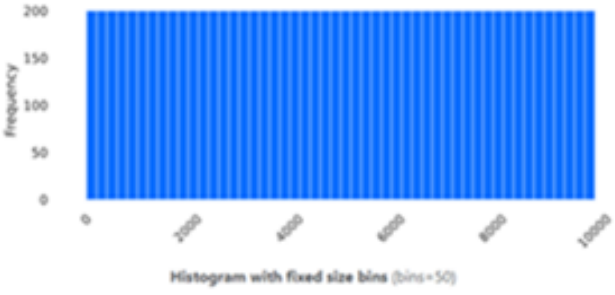


Figura 3. Histograma Product ID. Fuente: Elaboración propia.

Tabla 4. Variable Type. Fuente: Elaboración propia.

Type	
Categorical	
High correlation	
Distinct	3
Distinct (%)	< 0.1%
Missing	0
Missing (%)	0.00%
Memory size	566.5 KiB

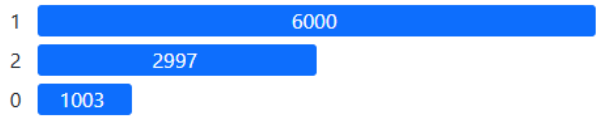
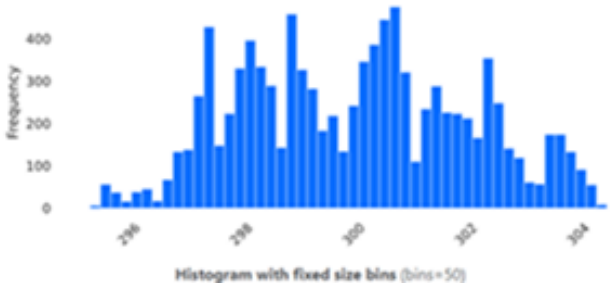


Figura 4. Grafica variable Type. Fuente: Elaboración propia.

Tabla 5. Variable Air Temperature K. Fuente: elaboración propia.

Air_temperature_K_	
Real number (ℝ)	
High correlation	
Distinct	93
Distinct (%)	0.90%
Missing	0
Missing (%)	0.00%
Infinite	0
Infinite (%)	0
Mean	300.00493
Minimum	295.3
Maximum	30450.00%
Zeros	0
Zeros (%)	0.00%
Negative	0
Negative (%)	0
Memory size	78.3 KiB



Fuente 5. Histograma Air Temperature K. Fuente: Elaboración propia.

Tabla 6. Variable Process Temperature K. Fuente: Elaboración propia.

Process_temperature_K_

Real number (ℝ)

High correlation

Distinct	82
Distinct (%)	0.80%
Missing	0
Missing (%)	0.00%
Infinite	0
Infinite (%)	0
Mean	310.00556
Minimum	305.7
Maximum	31380.00%
Zeros	0
Zeros (%)	0.00%
Negative	0
Negative (%)	0
Memory size	78.3 KiB

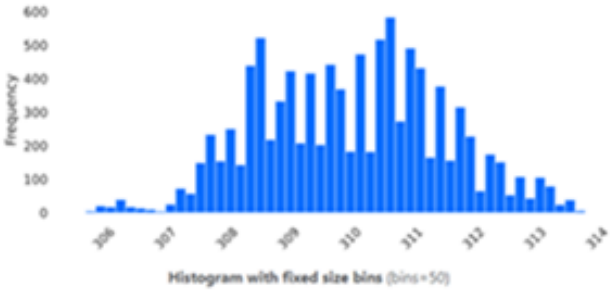


Figura 6. Histograma Process Temperature K. Fuente: Elaboración propia.

Tabla 7. Variable Rotational Speed rpm. Fuente: Elaboración propia.

Rotational_speed_rpm_

Real number (ℝ)

High correlation

Distinct	941
Distinct (%)	9.40%
Missing	0
Missing (%)	0.00%
Infinite	0
Infinite (%)	0
Mean	1538.7761
Minimum	1168
Maximum	288600.00%
Zeros	0
Zeros (%)	0.00%
Negative	0
Negative (%)	0
Memory size	78.3 KiB

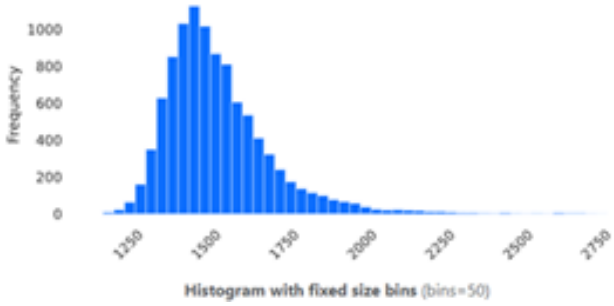


Figura 7. Histograma Rotational Speed rpm. Fuente: Elaboración propia.

Tabla 8. Variable Torque Nm. Fuente: Elaboración propia.

Torque_Nm_

Real number (ℝ)

High correlation

Distinct	577
Distinct (%)	5.80%
Missing	0
Missing (%)	0.00%

Infinite	0
Infinite (%)	0
Mean	39.98691
Minimum	3.8
Maximum	7660.00%
Zeros	0
Zeros (%)	0.00%
Negative	0
Negative (%)	0
Memory size	78.3 KiB

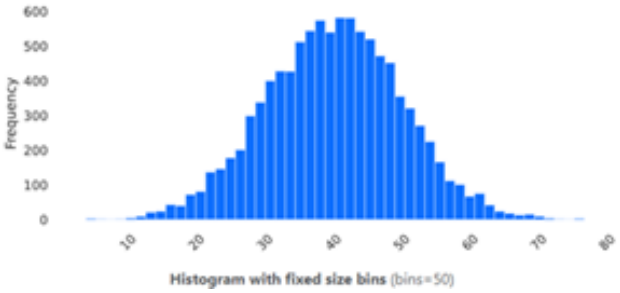


Figura 8. Histograma Torque. Fuente: Elaboración propia.

Tabla 9. Variable Tool wear min. Fuente: Elaboración propia.

Tool_wear_min_

Real number (ℝ)

Zeros

Distinct	246
Distinct (%)	2.50%
Missing	0
Missing (%)	0.00%
Infinite	0
Infinite (%)	0
Mean	107.951
Minimum	0
Maximum	25300.00%
Zeros	120
Zeros (%)	1.20%
Negative	0
Negative (%)	0
Memory size	78.3 KiB

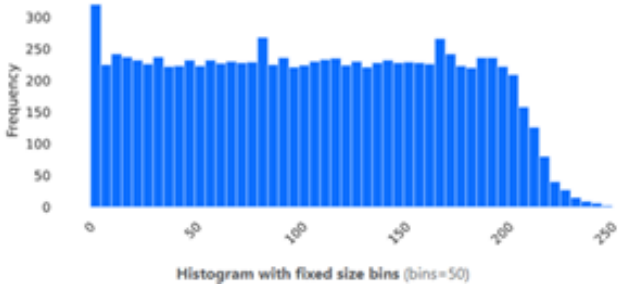


Figura 9. Histograma Tool wear min. Fuente: Elaboración propia.

Tabla 10. Variable Target. Fuente: Elaboración propia.

Target

Categorical

High correlation **Imbalance**

Distinct	2
Distinct (%)	< 0.1%
Missing	0
Missing (%)	0.00%
Memory size	566.5 KiB

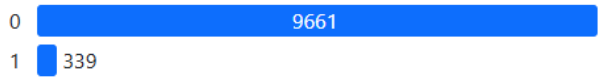


Figura 10. Grafica Target. Fuente: Elaboración propia.

Tabla 11. Variable Failure Type. Fuente: Elaboración propia.

Failure_Type

Real number (ℝ)

High correlation **Zeros**

Distinct	6
Distinct (%)	0.10%

Missing	0
Missing (%)	0.00%
Infinite	0
Infinite (%)	0
Mean	103.90%
Minimum	0
Maximum	500.00%
Zeros	112
Zeros (%)	1.10%
Negative	0
Negative (%)	0.00%
Memory size	78.3 KiB

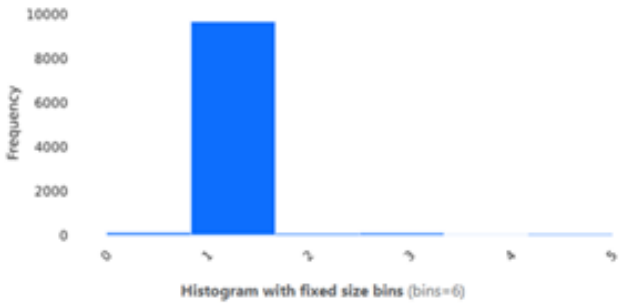


Figura 11. Histograma Failure Type. Fuente: Elaboración propia.

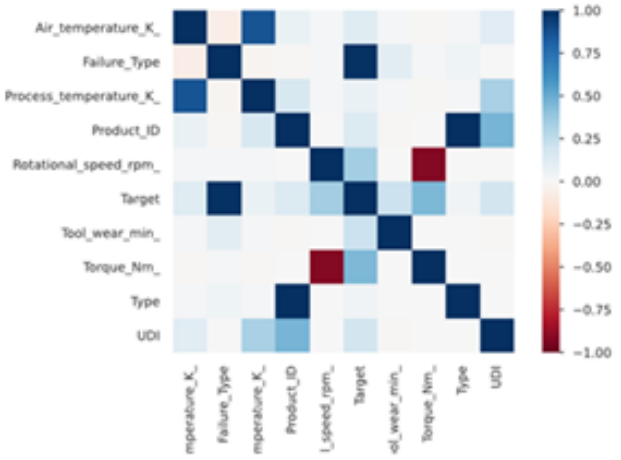


Figura 12. Matriz de Correlación. Fuente: Elaboración propia.

Fase 4: Evaluación

En la evaluación de los resultados obtenidos, utilizaremos métricas para comprobar que tan funcionales son los modelos obtenidos para predecir el valor de falla o no falla a partir de las características de entrada.

Exactitud (Accuracy): Indica cual es la proporción de predicciones correctas respecto al total de predicciones realizadas. Es el porcentaje de veces que el modelo acertó. Representada por la ecuación 1:

$$Accuracy = \frac{Tp+Tn}{Tp+fp+tn+fn} \quad (1)$$

Donde

$Tp = \text{Verdaderos positivos (True positives)}$

$Tn = \text{Verdaderos negativos (True negatives)}$

$fp = Falsos\ Positivos$

$fn = Falsos\ negativos$

Área Bajo la Curva ROC (ROC AUC):
Mide la capacidad del modelo para distinguir entre clases negativas y clases positivas. Mientras mas cerca el AUC de 1.0, mejor es el modelo.

Reporte de Clasificación: incluye las métricas de Precisión, Recall y F1-Score.

Precisión: Determina cantas predicciones fueron positivas. Representada por la ecuación 2:

$$Precision = \frac{Tp}{Tp+fp} \quad (2)$$

Donde

$Tp = \text{Verdaderos positivos (True positives)}$

$fp = Falsos\ Positivos$

Exhaustividad o Sensibilidad (Recall): De los casos positivos reales, cuantos fueron detectados. Representado por la ecuación 3:

$$Recall = \frac{Tp}{Tp+fn} \quad (3)$$

Donde

$Tp = \text{Verdaderos positivos (True positives)}$

$fn = Falsos\ negativos$

F1-Score: Combina precisión y Recall en una sola métrica. Es útil para clases desbalanceadas.

Matriz de Confusión: Esta tabla resume el rendimiento de un modelo mostrando:

$Tp =$ Verdaderos positivos (True positives)
$Tn =$ Verdaderos negativos (True negatives)
$fp = Falsos\ Positivos$
$fn = Falsos\ negativos$

Permite ver los errores específicos del modelo.
Curva ROC y Curva Precision-Recall:

Curva ROC: Muestra la relación entre la tasa de verdaderos positivos (TPR) y la tasa de falsos positivos (FRP) y funciona para elegir mejor el umbral de decisión.

Curva Precision-Recall: Se centra en las clases positivas y es útil cuando los datos son desbalanceados.

Fase 4: Resultados

Se presentan los resultados de la fase anterior de los algoritmos Random Forest y Naive Bayes.

Los resultados del Algoritmo Random Forest se muestran en la Tabla 12.

Tabla 12. Resultados Random Forest. Fuente: Elaboración propia.

Métrica	Valor
Accuracy	0.998
ROC AUC	0.9821
Precisión clase 0	1
Recall clase 0	1
Precisión clase 1	0.97
Recall clase 1	0.97
F1-Score clase 1	0.97
Support clase 0	1939
Support clase 1	61

Para este algoritmo los resultados muestran que el modelo detecta el fallo y estados normales correctamente, que fueron pocas las fallas que no se consideraron, casi no genera falsas alarmas y, además, el modelo distingue entre “falla” y “no falla” a diferentes umbrales de decisión. Algunos puntos que considerar son la clase desbalanceada, es decir, hay mas ejemplos de “no falla” que de “falla” 1939 y 61 respectivamente. Si el Recall fuera menor sería grave para el mantenimiento.

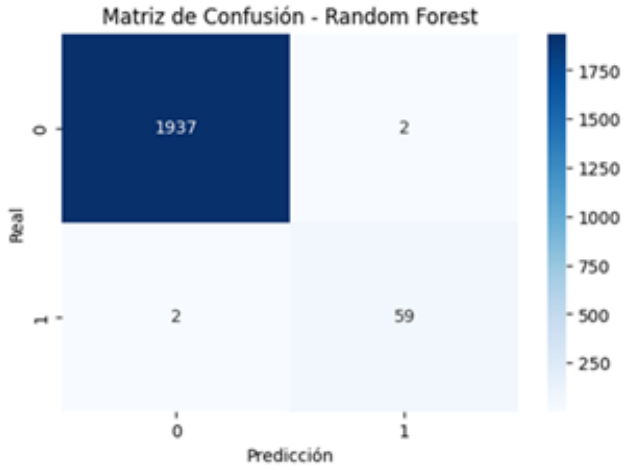


Figura 13. Matriz de Confusión Random Forest. Fuente: Elaboración propia.

La matriz de confusión representada en la ilustración 2 indica que solo existe 2 falsas alarmas, es decir, el modelo predijo “falla” cuando no la había. Así también, indica que hubo 2 falsos negativos, es decir, predijo “no falla” cuando si hubo falla.

De las predicciones correctas, detecta 59 de 61 fallas reales.

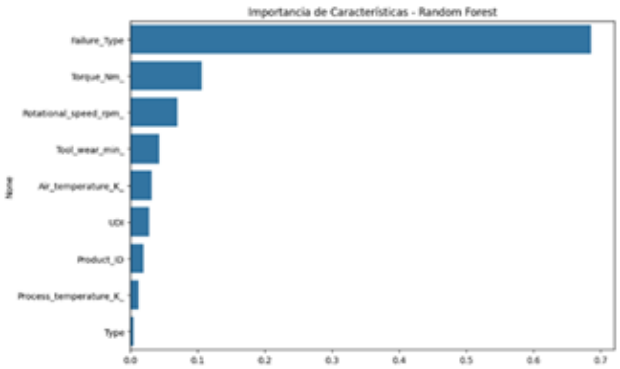


Figura 14. Características por nivel de importancia de Random Forest. Fuente: Elaboración propia.

En la Figura 14 se mide la importancia de cada característica del modelo, donde se puede observar que el tipo de falla es la característica mas importante para el modelo.

Este modelo asigna importancia basada en la capacidad de las variables para reducir la impureza en los nodos de decisión.

Los resultados del algoritmo Naive Bayes se muestran en la Tabla 13.

Tabla 13. Resultados de Naive Bayes. Fuente: Elaboración propia.

Métrica	Valor
Accuracy	0.991
ROC AUC	0.9873
Precisión clase 0	1
Recall clase 0	0.99
Precisión clase 1	0.79
Recall clase 1	0.97
F1-Score clase 1	0.87

Para este algoritmo los resultados muestran un Accuracy y Recall alto, lo cual indica que predice muy bien. Sin embargo, la precisión de fallas es considerablemente bajo, predice mas veces “falla” cuando en realidad no la hay.

Aun así, sigue siendo un modelo aceptable ya que logra una alta detección de fallas.

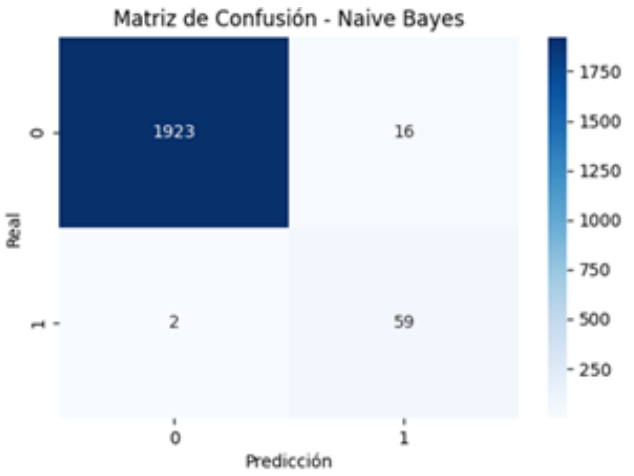


Figura 15. Matriz de Confusión Naive Bayes. Fuente: Elaboración propia.

La matriz de Confusión del algoritmo Naive Bayes mostrada en la Figura 15 indica que existe 16 falsas alarmas, es decir, el modelo predijo “falla” 16 veces cuando no ocurrió una falla. Detecta 59 de 61 fallas.

Este algoritmo es confiable para detectar fallas y detecta casi todas las fallas reales, sin embargo, se activaron mas falsas alarmas que en el algoritmo anterior.

Figura 16. Características por nivel de importancia de Naive Bayes. Fuente: Elaboración propia.

En la Figura 16 se define por orden la característica más importante del modelo. Para el modelo Naive Bayes es una aproximación. Este modelo asigna importancia basándose en la diferencia de medias entre clases.

En resumen, cualquier algoritmo que utilice el modelo tendrá una predicción de “falla” o “no falla” muy bueno. Sin embargo, en la siguiente sección se desarrolla mas a detalle la diferencia entre cada uno.

Fase 5: Comparación de modelos

Tabla 14. Diferencias entre modelos. Fuente: Elaboración propia.

Métrica	Random Forest	Naive Bayes
Accuracy	99.80%	99.10%
ROC AUC	98.21%	98.73%
Precisión (Clase Falla)	97%	79%
Recall (Clase Falla)	97%	97%
F1-Score (Clase Falla)	97%	87%
Falsas Alarmas (Falsos Positivos)	2	16

Fallas No Detectadas (Falsos Negativos)	2	2
Robustez	Alta, menos sensibles a datos ruidosos	Depende de la independencia de variables
Velocidad de entrenamiento	Más lento	Muy rápido
Facilidad de interpretación	Moderada	Alta

Comparando ambos modelos, Random Forest logra una mejor predicción de fallas con menos falsas alarmas que Naive Bayes, aunque, ambos presenten un Recall similar, indicando una alta capacidad de detección de fallas.

A pesar de que Naive Bayes genere mas falsas alarmas, es un modelo rápido y fácil de implementar, adecuado para entornos donde la velocidad es preferible antes que la precisión.

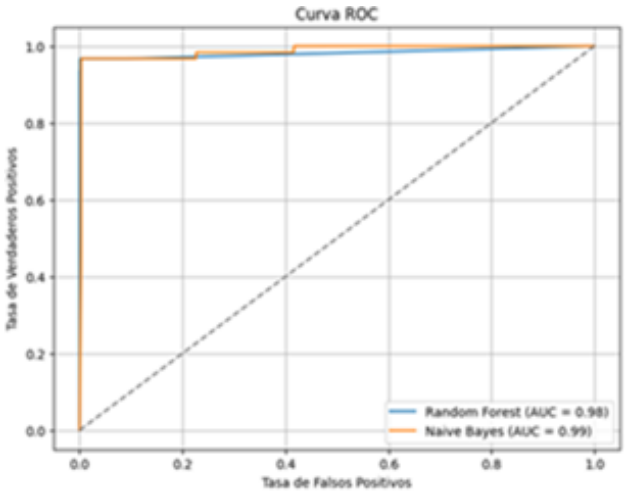


Figura 17. Curva ROC de comparación de ambos modelos. Fuente: Elaboración propia.

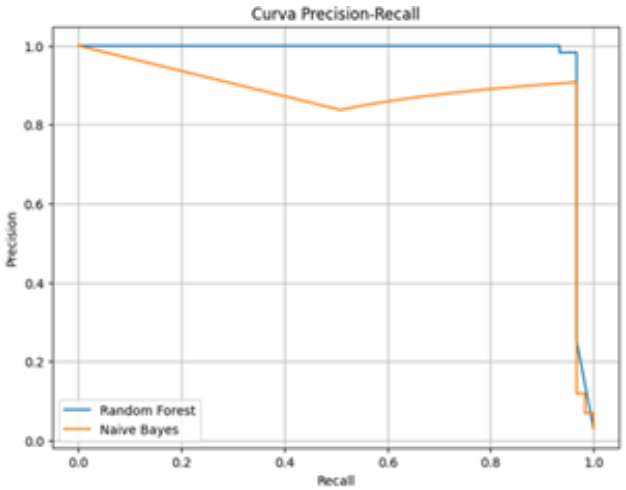


Figura 18. Curva Precisión-Recall de comparación de Modelos. Fuente: Elaboración propia.

Las Figuras 17 y 18 muestran la diferencia entre ambos modelos dependiendo de la Accuracy y las predicciones de "falla" falsas.

Aun que ambos modelos destaquen variables similares y la curva sea muy similar, existen diferencias en cada algoritmo.

Conclusión

El presente trabajo logró demostrar cómo, a partir de un conjunto de datos obtenidos es posible implementar modelos matemáticos de mantenimiento predictivo utilizando técnicas de aprendizaje automático supervisado. Se emplearon dos algoritmos de clasificación: Random Forest o Bosque Aleatorio y Naive Bayes, para predecir el valor de "Target" (falla o no falla) en maquinaria industrial.

El tratamiento de los datos, el entrenamiento de los modelos y la evaluación de su desempeño permitieron identificar ventajas, desventajas y limitaciones de cada algoritmo utilizado. Random Forest presentó un rendimiento superior, con una precisión de 99.80% y una capacidad de detección de fallas y reducción de falsas alarmas. Naive Bayes, aunque alcanzó una precisión ligeramente inferior (99.10%), destacó por su rapidez de entrenamiento y su alta sensibilidad en la identificación de fallas.

La comparación entre ambos modelos evidencia que la inteligencia artificial es una herramienta efectiva para la detección temprana de fallas en el mantenimiento industrial, permitiendo mejorar la eficiencia operativa, prevenir averías y reducir costos. La elección entre Random Forest y Naive Bayes dependerá del contexto industrial, considerando factores como la necesidad de

precisión, la tolerancia a falsas alarmas, y la rapidez en la implementación.

El estudio confirma que el uso de algoritmos de aprendizaje automático en mantenimiento predictivo tiene un potencial significativo para optimizar las operaciones y prolongar la vida útil de los equipos industriales.

Referencias

Achouch, M. D. (2022). On predictive maintenance in Industry 4.0: Overview, models, and challenges. *Applied Sciences, 12*, pág. 8081. doi:10.3390/app12168081

Ahmed, S. &. (2023). An Industry 4.0 implementation of a condition monitoring system and IoT-enabled predictive maintenance scheme for diesel generators. *Alexandria Engineering Journal, 76*, págs. 525-541. doi:10.1016/j.aej.2023.06.026

Chen, Y. &. (2024). Logistic-ELM: A Novel Fault Diagnosis Method for Rolling Bearings. *Journal of Mechanical Engineering Science, 56*(2), 145-158. Obtenido de https://doi.org/10.1177/0954406224

Darshan, K. G. (2024). Machine learning and IoT–Based predictive maintenance approach for industrial applications. *Alexandria Engineering Journal, 88*, págs. 298-309. doi:10.1016/j.aej.2023.11.030

Garcia, L. &. (2023). Predictive maintenance for smart industrial systems: A roadmap. *Procedia Computer Science, 220*, págs. 645-650. doi:DOI: 10.1016/j.procs.2023.03.082

Gupta, P. K. (2024). Improved Fault Classification for Predictive Maintenance in Industrial Applications. *MDPI Sensors, 24*(5), 1387. Obtenido de https://doi.org/10.3390/s24051387

Gupta, V. S. (2025). Predictive Maintenance of Armoured Vehicles using Machine Learning Approaches. *Military Technology and Engineering, 12*(1), 90-110. Obtenido de https://doi.org/10.1016/j.miltecheng.2024.12.001

Kumar, R. &. (2024). Paradigm shift for predictive maintenance and condition monitoring from Industry 4.0 to Industry 5.0: A systematic review, challenges and case study. *Results in Engineering, 24*, pág. 102935. doi:10.1016/j.rineng.2024.102935

Labañino Urbina, S. V. (2019). Algoritmo Random Forest para la detección de fallos en redes de computadoras. *Serie Científica de la Universidad de las Ciencias Informáticas, 12*(8), págs. 27-41.

Lazaroiu, G. A. (2023). Predictive Maintenance Algorithms, Artificial Intelligence Digital Twin Technologies, and Internet of Robotic Things in Big Data-Driven Industry 4.0 Manufacturing Systems. *Mathematics, 11*(6), pág. 981. doi:10.3390/math11060981

Liao, L. K. (2023). Challenges in predictive maintenance – A review. *CIRP Journal of Manufacturing Science and Technology, 40*, págs. 53-67. doi:10.1016/j.cirpj.2022.11.003

Lopez, M. &. (2024). Predictive maintenance in Industry 4.0: A systematic multi-sector mapping. *CIRP Journal of Manufacturing Science and Technology, 50*, págs. 80-103. doi:10.1016/j.cirpj.2024.02.003

Maheshwari, S. T. (2024). Comprehensive Study Of Predictive Maintenance In Industries Using Classification Models And LSTM Model. *arXiv*.

Molina Salgado, J. L. (2024). Preprocesamiento de datos en el pronóstico de fallos de rodamientos para el mantenimiento predictivo. *Computacion, 28*(4), págs. 1811-1821. doi:10.13053/CyS-28-4-4913

Shivam. (2019). *Machine Predictive Maintenance Classification*. Obtenido de Kaggle: https://www.kaggle.com/datasets/shivamb/machine-predictive-maintenance-classification

Smith, J. &. (2024). Improve predictive maintenance through the application of artificial intelligence: A systematic review. *Results in Engineering, 21*, pág. 101645. doi:10.1016/j.rineng.2023.101645

Technology, L. U. (April de 2023). Bearing Fault Detection Scheme Using Machine Learning for Condition Monitoring Applications. *Proc. of the International Conference on Mechanical, Automotive and Mechatronics Engineering*, 29-30.

Ucar, A. K. (2024). Artificial Intelligence for Predictive Maintenance Applications: Key Components, Trustworthiness, and Future Trends. *Applied Sciences, 14*, pág. 898. doi:10.3390/app14020898

Wu, X. &. (2024). Predictive Machine Failure Using Binary Classification with Deep Learning. *Artificial Intelligence and Machine Learning Journal, 20*(3), 287-301. Obtenido de https://doi.org/10.1016/j.aimlj.2024.04.003

Zhang, H. (2004). The Optimality of Naive Bayes. AAAI Conference on Artificial Intelligence. *3*, págs. 562-567.