Conciencia Digital

ISSN: **2600-5859**
Vol. 5 No 2.1, pp. 45 – 68 , mayo 2022

www.concienciadigital.org

Aprendizaje de máquina para mantenimiento predictivo: un problema de clasificación binaria

Machine learning for predictive maintenance: a binary classification problem

[1] Pablo Hernán Vilema Lara https://orcid.org/0000-0002-3983-4606
Escuela Superior Politécnica de Chimborazo (ESPOCH), Facultad de Mecánica. Riobamba, Ecuador. pablo.vilema@espoch.edu.ec
[2] Félix Antonio García Mora https://orcid.org/0000-0001-5814-3694
Escuela Superior Politécnica de Chimborazo (ESPOCH), Facultad de Mecánica. Riobamba, Ecuador. felix.garcia@espoch.edu.ec
[3] César Marcelo Gallegos Londoño https://orcid.org/0000-0002-8685-7501
Escuela Superior Politécnica de Chimborazo (ESPOCH), Facultad de Mecánica. Riobamba, Ecuador. edhernandez@espoch.edu.ec

Artículo de Investigación Científica y Tecnológica
Enviado: 11/03/2022
Revisado: 16/04/2022
Aceptado: 03/05/2022
Publicado:11/05/2022
DOI: **https://doi.org/10.33262/concienciadigital.v5i2.1.2150**

Cítese: Lamilla Vera , L. P., Álvarez Santana, C. L., & Tapia Segarra, J. I. (2022). Los emprendimientos y su aporte al desarrollo socioeconómico de las mujeres de la Feria Ruku Kawsay en Orellana Ecuador. ConcienciaDigital, 5(2.1), 21-44. https://doi.org/10.33262/concienciadigital.v5i2.1.2147

CONCIENCIA DIGITAL, *es* una revista multidisciplinar, t**rimestral,** que se publicará en soporte electrónico tiene como **misión** contribuir a la formación de profesionales competentes con visión humanística y crítica que sean capaces de exponer sus resultados investigativos y científicos en la misma medida que se promueva mediante su intervención cambios positivos en la sociedad. https://concienciadigital.org
La revista es editada por la Editorial Ciencia Digital (Editorial de prestigio registrada en la Cámara Ecuatoriana de Libro con No de Afiliación 663) www.celibro.org.ec

Conciencia Digital

ISSN: **2600-5859**
Vol. 5 No 2.1, pp. 45 – 68 , mayo 2022

www.concienciadigital.org

Palabras claves:
Aprendizaje de máquina, mantenimiento predictivo, detección de fallos, clasificación binaria.

Resumen

Introducción. Con el auge de la industria 4.0, se están extrayendo de las máquinas y procesos una gran cantidad de datos, los cuales pueden ser analizados mediante enfoques de aprendizaje de máquina, permitiendo una toma de decisiones más confiable dentro del área de mantenimiento; realizar análisis de datos de mantenimiento predictivo se vuelve un verdadero reto para un ser humano debido a la gran cantidad de datos. **Objetivo.** Por esta razón en el presente estudio, se plantea como objetivo crear un modelo predictivo de aprendizaje de máquina para detectar fallos. **Metodología.** Para la creación del modelo se utilizó los datos de mantenimiento predictivo ai4i2020 disponibles en el repositorio de *Machine Learning* de la Universidad de California y el software libre Python. Se probó 4 algoritmos de clasificación, con la finalidad de compararlos en función de las métricas de rendimiento. **Resultados.** Dando como resultado que SVM es el mejor algoritmo con una exactitud del 98,95% y una precisión de 98,88% (optimizados los hiperparámetros). **Conclusiones.** Se concluye que el modelo funciona con un elevado rendimiento y una buena generalización de los patrones aprendidos durante el entrenamiento, en datos de prueba o datos no vistos por el algoritmo.

Keywords:
Machine learning, predictive maintenance, fault detection, binary classification

Abstract

Introduction. With the rise of Industry 4.0, a large amount of data is being extracted from machines and processes, which can be analyzed using machine learning approaches, allowing for more reliable decision making within the maintenance area; performing predictive maintenance data analysis becomes a challenge for a human being due to the large amount of data. **Objective.** For this reason, the objective of this study is to create a predictive machine learning model to detect failures. **Methodology.** The ai4i2020 predictive maintenance data available in the Machine Learning repository of the University of California and the free Python software were used to create the model. Four classification algorithms were evaluated to compare them based on performance metrics. **Results.** As a result, SVM is the best algorithm with an accuracy of 98.95% and a precision of 98.88% (optimized hyperparameters). **Conclusions.** It is concluded that the model works with high performance and good generalization of patterns

learned during training, on test data or data not seen by the algorithm.

Introducción

En toda industria, el mantenimiento de máquinas y equipos es un aspecto muy importante a tener en cuenta, ya que se encuentra relacionado directamente con la eficiencia y el tiempo de operación de los equipos, debido a ello es fundamental y necesario detectar y solucionar las fallas en los equipos antes que estas ocurran, evitando paradas inesperadas en los procesos productivos (Wan et al., 2017). Uno de los principales problemas dentro del área de mantenimiento son los paros imprevistos en las máquinas, por ende, se evidenciará un impacto negativo en los costos asociados a la productividad y al mantenimiento, generando cuantiosas pérdidas económicas a nivel global de toda una empresa.

El tiempo de inoperatividad de las máquinas y los costos asociados a la producción obedecen en gran medida a la estrategia de mantenimiento aplicada en una empresa, para maximizar el tiempo de operatividad de la máquina y elevar la disponibilidad, los fallos deben detectarse y corregirse antes que las máquinas lleguen al punto de fallo (Fernandes et al., 2020). Una empresa necesita de estrategias de mantenimiento óptimas que ayuden a garantizar la confiabilidad de los sistemas, disminuir los costos, evadir tiempos de inactividad y maximizar el tiempo de vida útil de un elemento (Lee et al., 2019). Entre las principales estrategias de mantenimiento se tiene: correctiva, preventiva y predictiva; la estrategia correctiva se caracteriza porque no se realiza ninguna actividad de mantenimiento hasta que la máquina presente una avería (Kang et al., 2016), por otro lado, la estrategia preventiva la cual es la más utilizada en la industria (Lee et al., 2019), la cual se caracteriza porque las actividades de mantenimiento se ejecutan de acuerdo a intervalos periódicos de tiempo preestablecidos, es decir sin importar la condición del componente, y finalmente se tiene la estrategia de mantenimiento predictiva, donde las actividades se ejecutan únicamente cuando son necesarias y antes que ocurran los fallos (Carvalho et al., 2019) basándose en un análisis del monitoreo continuo del estado de salud de la máquina o elemento, de entre las 3 estrategias la que más destaca es la predictiva debido a que brinda ventajas como: maximizar el tiempo de operación de los equipos, retrasar o reducir la ejecución de actividades de mantenimiento y disminuir notablemente los costos de materiales, repuestos y mano de obra (Carvalho et al., 2019).

En la actualidad, la industria atraviesa la cuarta revolución industrial o también conocida como industria 4.0, misma que implica la utilización de tecnologías IoT con la finalidad de permitir el intercambio de información entre sensores, máquinas y usuarios finales (Fernandes et al., 2020). Gracias a estas tecnologías y entornos ciber físicos se puede recopilar de las máquinas y procesos una gran cantidad de datos, que generalmente se

añaden y almacenan en la nube (Kanawaday & Sane, 2018). El procesamiento de estos datos se vuelve una tarea compleja para un ser humado, debido al gran volumen y a la velocidad que se generan (Fernandes et al., 2020), es por ello que para realizar este análisis surge la necesidad de implementar el aprendizaje de máquina el cual es una de las disciplinas principales de la inteligencia artificial y se fundamenta en extraer conocimiento de los datos para posteriormente aplicarlo en la toma de decisiones. El aprendizaje de máquina se ha convertido en la actualidad en una herramienta muy valiosa para desarrollar modelos predictivos inteligentes en muchas aplicaciones (Carvalho et al., 2019), aprendizaje de máquina tiene la capacidad de manejar datos multivariados y de gran dimensión y de extraer relaciones ocultas dentro de los datos en entornos complejos y dinámicos, como por ejemplo un entorno industrial (Wuest et al., 2016), por lo tanto el aprendizaje de máquina brinda enfoques predictivos poderosos para aplicaciones de mantenimiento predictivo (Carvalho et al., 2019).

La presente investigación tiene por objetivo desarrollar un modelo de aprendizaje de máquina supervisado, utilizando los datos de mantenimiento predictivo del repositorio de machine *learning* de la Universidad de California, aplicando varios algoritmos de clasificación con la finalidad de realizar una comparación en función de las métricas de rendimiento y encontrar el que mejor rendimiento proporcione al momento de detectar fallas en una máquina.

Trabajos relacionados

Varias son las aplicaciones y estudios que se han desarrollado con enfoque de aprendizaje de máquina aplicado al mantenimiento predictivo, por ejemplo, en la investigación realizada por Canizo et al. (2017), presenta la evolución de una aplicación de mantenimiento predictivo a un entorno de Big Data, donde el objetivo fue realizar la predicción de fallas en turbinas eólicas empleando una solución basada en datos alojada en la nube la cual se compone de tres módulos principales: el primer módulo hace referencia a un generador de modelo predictivos para cada turbina eólica utilizando el algoritmo de *Random Forest*, el segundo se refiere a un agente de monitoreo que se encarga de realizar predicciones cada 10 minutos sobre fallas de las turbinas eólicas, y finalmente el tercer módulo que hace referencia a un tablero donde se pueden observar las predicciones realizadas.

Bien se conoce de la importancia de las funciones y misiones que cumplen una infraestructura nuclear, en beneficio de un país, empresas y la sociedad en general; es por ello que Gohel et al. (2020), proponen diseñar y desarrollar un modelo de aprendizaje de máquina con la finalidad de realizar el mantenimiento predictivo de una infraestructura nuclear, para lo cual emplearon algoritmos como la máquina de soporte vectorial (SVM) y la regresión logística para la realización de las predicciones. Los datos son adquiridos de sensores de temperatura, presión, vibración y acelerómetros los cuales se encuentran

instalados en sistemas y subsistemas para monitorear las máquinas y procesos de la infraestructura nuclear.

Otra de las investigaciones novedosas de aprendizaje de máquina aplicadas al mantenimiento predictivo es la desarrollada por Ayvaz & Alpay (2021), en la cual desarrollaron un sistema de mantenimiento predictivo basado en datos para líneas de producción en manufactura, el cual tuvo por objetivo detectar señales de posibles fallas antes que esas sucedan mediante técnicas de aprendizaje de máquina. Los datos utilizados fueron generados por sensores IoT en tiempo real y los resultados mostraron que el sistema de mantenimiento predictivo pudo identificar los patrones de fallas potenciales y con ello ayudar a prevenir paradas inesperadas en los procesos productivos.

De acuerdo al estudio realizado por Schwendemann et al. (2021), mencionan que en la última década se han desarrollado muchas investigaciones centradas en la detección de fallas en rodamientos, por lo tanto, en su investigación dan a conocer una descripción general de los enfoques más significativos para el estudio de fallas de rodamientos en máquinas herramientas como la rectificadora. El documento presenta dos partes principales del análisis; la primera parte está orientada a la clasificación de las fallas de rodamientos, la cual incluye la detección de condiciones no saludables, la posición del error y la gravedad, la segunda parte hace referencia a la predicción de la vida útil remanente la cual ayuda a optimizar los costos de recambio y a minimizar el tiempo de inactividad de la máquina.

Algoritmos de aprendizaje de máquina para clasificación

Los algoritmos que se consideraron para realizar la comparación en el presente estudio son: *Random Forest, Gradient Boosting, XGBoost* y la Máquina de vectores de soporte (SVM); estos algoritmos serán descritos brevemente a continuación.

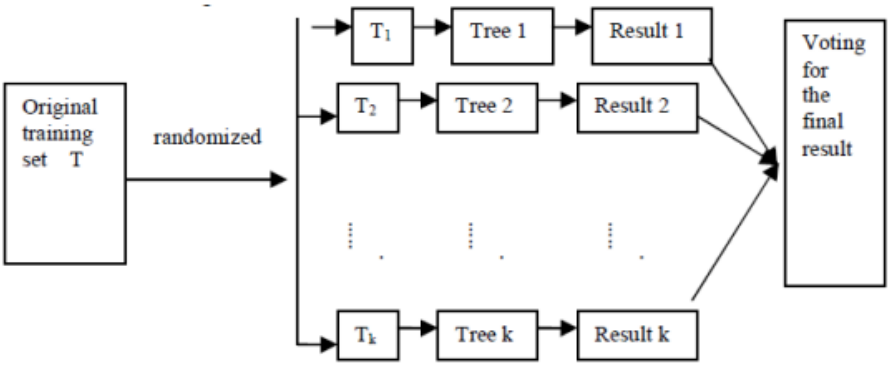
Random Forest

Random Forest o por su traducción en español como bosque aleatorio es un clasificador que consta de una colección de clasificadores estructurados en forma de árbol $\{h(x, \theta_k), k = 1, \dots\}$ donde $\{\theta_k\}$ son vectores aleatorios independientes e idénticamente distribuidos y cada árbol de decisión emite un voto unitario para la clase más popular en la entrada x (Breiman, 2001).

ISSN: **2600-5859**
Vol. 5 No 2.1, pp. 45 – 68 , mayo 2022

www.concienciadigital.org

Figura 1
Esquema de Random Forest



Fuente: Liu et al. (2012)

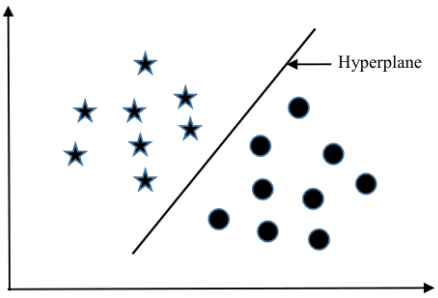
En el modelo de *Random Forest* propuesto por Breiman, cada árbol de decisión se planta sobre la base de un conjunto de muestras de entrenamiento y una variable aleatoria, la variable aleatoria correspondiente al k-ésimo árbol se denota como θ_k, entre dos cualesquiera de estas dos variables son independientes e idénticamente distribuidas, dando como resultado un clasificador $h(x, \theta_k)$ donde x es el vector de entrada. Luego de k ejecuciones se obtiene clasificadores en secuencia $\{h_1(x), h_2(x), \dots h_k(x)\}$, mismos que son utilizados para constituir más de un sistema modelo de clasificación, el resultado final del sistema es elegido por mayoría ordinaria de votos (Liu et al., 2012), este proceso se muestra en la figura 1.

Máquina de vectores de soporte (SVM)

La máquina de vectores de soporte (SVM), es un algoritmo capaz de clasificar casos linealmente separables y no linealmente separables. En primer lugar, mapea cada componente de datos en un espacio de características n-dimensional donde n es el número de características. Posteriormente identifica el hiperplano que separa los datos en dos clases a medida que maximiza la distancia marginal para las dos clases minimizando de esta forma los errores de clasificación. La distancia marginal para una determinada clase está determinada por la distancia entre el hiperplano de decisión y su instancia más cercana perteneciente a esa clase. Para ejecutar la clasificación se requiere encontrar el hiperplano capaz de diferenciar las dos clases por el margen máximo (Uddin et al., 2019), la figura 2 muestra una ilustración simplificada de SVM.

Conciencia Digital
ISSN: **2600-5859**
Vol. 5 No 2.1, pp. 45 – 68 , mayo 2022
www.concienciadigital.org

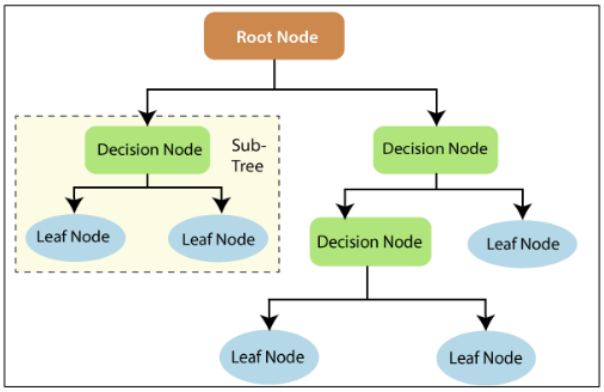
Figura 2
Funcionamiento simplificado de la máquina de vectores de soporte



Fuente: Uddin et al. (2019)

Árbol de decisión

Figura 3
Árbol de decisión



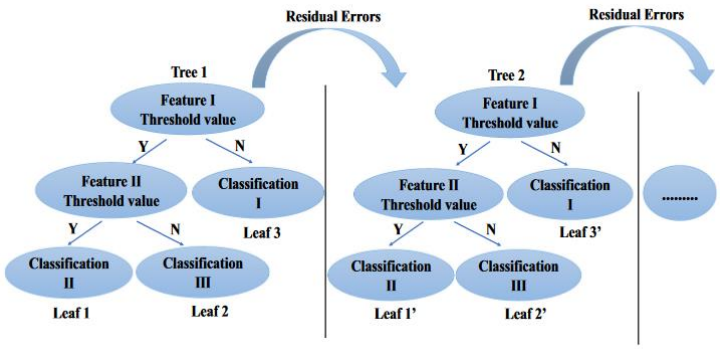
Fuente: Jijo & Abdulazeez (2021)

Árbol de decisión es una técnica de aprendizaje de máquina basada en árboles, los cuales agrupan atributos ordenándolos según sus valores. La técnica de árbol de decisión es utilizada primordialmente para fines de clasificación, cada árbol está compuesto de nodos, ramas y hojas; el nodo representa atributos o características en un grupo a clasificar, la rama el valor que el nodo puede tomar y las hojas representan el final de los datos o resultado (Dey, 2016). En clasificación el algoritmo de árbol de decisión emplea la entropía y la ganancia de información para construir un árbol de decisión en lugar del método de reducción de la desviación estándar. La entropía es empleada para calcular la homogeneidad de la muestra y para que esta sea cero tiene que ser completamente homogénea algo que se lo consigue únicamente si la muestra se divide en partes iguales (Gianey & Choudhary, 2018). La figura 3 muestra la estructura un árbol de decisión.

XGBoost

El *Boosting* o impulso en español, es un método de ensamble que crea miembros de conjunto de manera secuencial, donde el miembro más nuevo se genera con la finalidad de compensar las instancias que fueron etiquetadas de manera incorrecta por los alumnos anteriores. Por otra parte, el aumento de gradiente es una variación que representa el problema de aprendizaje como un descenso de gradiente en alguna función de pérdida diferenciable arbitraria que mide el rendimiento del modelo en el conjunto de entrenamiento. De allí se puede mencionar que *XGBoost* es en esencia un algoritmo de aprendizaje de máquina de impulso de árboles de decisión, donde cada nuevo modelo que se genera intenta corregir las deficiencias del modelo anterior. A su vez puede mencionar que *XGBoost* es una implementación generalizada de aumento de gradiente que incluye un término de regularización, el cual es empleado para combatir los problemas de sobreajuste, así como soporte para funciones de pérdida diferenciables arbitrarias (Mitchell & Frank, 2017), la figura 4 muestra el esquema de árboles *XGBoost*.

Figura 4
Esquema de árboles XGBoost



Fuente: Dong et al. (2020)

Datos desequilibrados

El desequilibrio de los datos en aprendizaje de máquina se refiere a una distribución no uniforme de las clases dentro de un *dataset*. Dicho problema principalmente se presenta en tareas de clasificación donde la distribución de etiquetas en un conjunto de datos es desigual. Existen principalmente dos formas de solucionar este problema, se lo puede llevar a cabo añadiendo registros u observaciones a la clase minoritaria (*oversampling*) o a su vez eliminando registros de la clase mayoritaria (*undersampling*). Algunas de las técnicas de *undersampling* son: enlaces *tomeks*, centroides *cluster*, entre otras; y por otro lado una de las técnicas de *oversampling* mayormente empleada es el método SMOTE (técnica de sobre muestreo de minoría sintética). El trabajar con datos desequilibrados impacta directamente de manera negativa en el rendimiento del modelo, ya que el algoritmo con el cual se esté ejecutando el modelo pondrá mayor atención en la clase

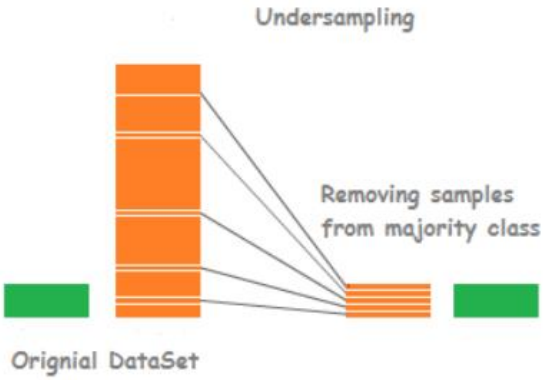
ISSN: **2600-5859**
Vol. 5 No 2.1, pp. 45 – 68 , mayo 2022

mayoritaria e ignorará la clase minoritaria, lo cual desencadena en obtener predicciones incorrectas, un ejemplo típico de desequilibrio de datos es la detección de fraude, donde se tendrán registros abundantes de una clase (no fraude) y muy pocos registros de la otra (fraude) (Mohammed et al., 2020). A continuación, la figura 5 y la figura 6 muestran el proceso de *oversampling* y *undersampling* respectivamente.

Figura 5
Proceso de oversampling

Fuente: Mohammed et al. (2020)

Figura 6
Proceso de undersampling



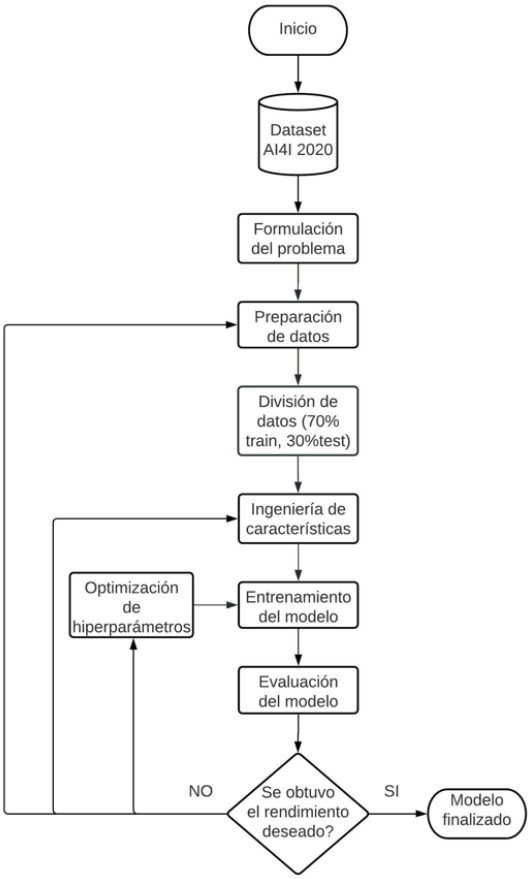
Fuente: Mohammed et al. (2020)

Metodología

La metodología para crear el modelo predictivo de detección de fallos fundamentado en aprendizaje de máquina sigue los pasos del diagrama de flujo que se muestra en la figura 7.

Conciencia Digital
ISSN: **2600-5859**
Vol. 5 No 2.1, pp. 45 – 68 , mayo 2022
www.concienciadigital.org

Figura 7
Metodología para crear el modelo



Descripción del conjunto de datos

Para la realización del modelo de aprendizaje de máquina, se utilizó la base de datos de mantenimiento predictivo ai4i2020 que se encuentra disponible en el repositorio de machine *learning* de la Universidad de California (Dua & Graff, 2019). La base de datos está conformada de 10000 observaciones almacenadas como filas con 14 características: UDI (identificador único), Product_ID (número de serie y letra que representa la variante de calidad del producto), *Type* (letra de la variante de calidad del producto), *Air_temperature_K* (temperatura del aire en grados Kelvin), *Process_temperature_K* (temperatura de proceso en grados Kelvin), *Rotational_speed_rpm* (velocidad rotacional en rpm), *Torque_Nm* (torque en Nm), *Tool_wear_min* (desgaste de la herramienta en minutos), *Machine failure* (falla de la máquina); a la etiqueta falla de máquina se encuentra asociados 5 modos de fallo independientes: TWF (falla por desgaste de herramienta), HDF (falla por disipación de calor), PWF (falla debido a la potencia), OSF (falla debido al sobre esfuerzo), RNF (fallos aleatorios). Si al menos uno de los modos de

fallo mencionados anteriormente es verdadero, la etiqueta falla de la máquina se establece en 1, de no ser así se establece en 0 (Dua & Graff, 2019).

Formulación del problema

La base de datos de mantenimiento predictivo ai4i2020 puede ser analizada de dos formas, la primera está enfocada en realizar predicciones de la falla de la máquina (clasificación binaria), y en la segunda se puede realizar predicciones de cuál fue el modo de fallo que ocasionó la falla de la máquina (clasificación multiclase), el presente estudio únicamente se enfocará en la clasificación binaria; por lo que como conjunto de características se empleará las variables de proceso y no las de modo de fallo, ya que puede existir fuga de datos, y como variable objetivo se utiliza la variable referente a la falla de la máquina.

Preparación de datos

Dentro del paso de preparación de los datos, se realizará: limpieza de los datos, análisis exploratorio, y sobre muestreo debido al desequilibrio.

Limpieza de datos

De la base de datos original se procede a eliminar todas las variables referentes a modos de fallos, a su vez se eliminan las variables UDI, *Product_ID* y *Type* ya que se considera que no contribuyen a la predicción. Una vez eliminadas las variables mencionadas, la base de datos únicamente cuenta con 5 variables de proceso y la variable objetivo, las cuales serán empleadas para crear el modelo.

Análisis exploratorio

El análisis exploratorio de los datos se lo realiza con la finalidad de encontrar posibles errores en la base de datos para posteriormente ser corregidos. Se comprueba que la base de datos no cuenta con valores faltantes por lo que no se hará uso de métodos de imputación (un paso muy necesario en caso de existirlos), a su vez tampoco se cuenta con filas duplicadas.

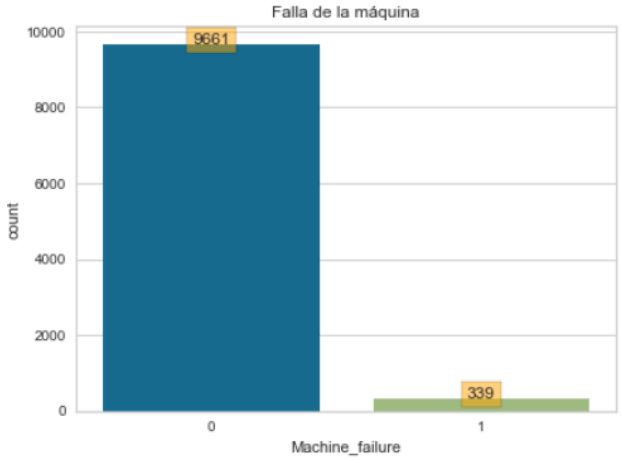
Mediante análisis gráfico se concluye que las variables de proceso *Rotational_speed_rpm* y *Torque_Nm* cuentan con valores atípicos, pero no serán eliminados ya que la base de datos no es muy extensa y se podría perder información valiosa para el análisis, a su vez en el proceso de ingeniería de características se utilizará para la normalización la función *RobustScaler*, la cual ayuda a minimizar el impacto de valores atípicos.

Al analizar gráficamente la variable objetivo se determina que existe desequilibrio en la base de datos, ya que existen más registros para la clase 0 (9661) y muy pocos registros para la clase 1 (339), tal y como lo muestra la Figura 8. Este problema es recurrente para

Conciencia Digital
ISSN: **2600-5859**
Vol. 5 No 2.1, pp. 45 – 68 , mayo 2022
www.concienciadigital.org

algunos casos, como por ejemplo el mantenimiento predictivo, detección de fraudes, entre otros.

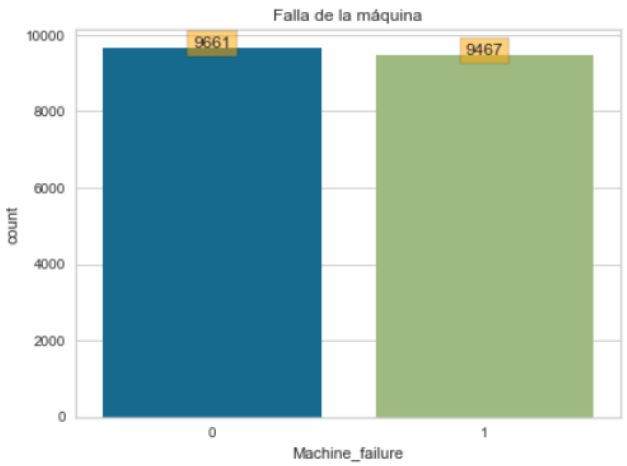
Figura 8
Análisis gráfico de la variable objetivo



Sobre muestreo

Debido a que crear un modelo con una base de datos desequilibrada reduce notablemente el rendimiento del modelo, se procede a emplear la función *Smote*, la cual añade muestras sintéticas a la clase minoritaria, de esta forma se puede apreciar una base de datos equilibrada tal y como muestra la figura 9.

Figura 9
Variable objetivo-equilibrada



Se aprecia que la base de datos pasa de 339 muestras en la clase 1 a 9467, es decir se añadieron 9218 muestras a la base de datos original, quedando finalmente el *dataset* con 19128 observaciones

División de datos

La división de datos se estableció en 70% para entrenamiento y 30% para la prueba. Del total de observaciones (19128), se distribuye el 70% para entrenamiento dando un total de 13389 observaciones y para la prueba se emplean los datos restantes, es decir 5739 observaciones. Cabe acotar que esta no es la cantidad de datos que se ingresarán en los clasificadores, ya que más adelante se reducirán notablemente debido a la extracción de características en el dominio del tiempo.

Ingeniería de características

Este paso contempla: extracción, selección de características, y estandarización de estas.

Extracción de características

Tabla 1
Características extraídas

Característica	Fórmula
Energía absoluta	$t1 = \sum_{n=1}^{N} (x(n))^2$
Media	$t2 = \frac{1}{N} \sum_{n=1}^{N} x(n)$
Raíz media cuadrada	$t3 = \sqrt{\frac{1}{N} \sum_{n=1}^{N} (x(n))^2}$
Valor máximo	$t4 = \max(x(n))$
Valor mínimo	$t5 = \min(x(n))$
Varianza	$t6 = \frac{1}{N} \sum_{n=1}^{N} (x_n - t2)^2$
Desviación estándar	$t7 = \sqrt{\frac{1}{N} \sum_{n=1}^{N} (x_n - t2)^2}$
Curtosis	$t8 = \frac{N \sum_{n=1}^{N} (x_n - t2)^4}{[\sum_{n=1}^{N} (x_n - t2)^2]^2}$
Asimetría	$t9 = \frac{N \sum_{n=1}^{N} (x_n - t2)^3}{(t7)^3}$
Rango Inter cuartil	$t10 = Q_3 - Q_1$
Distancia pico-pico	$t11 = \max(x_n) - min(x_n)$
Mediana	$t12 = \frac{1}{2} (x_{\frac{N}{2}} + x_{\frac{N}{2}+1})$

Donde x(n) es una serie de tiempo con N puntos

Conciencia Digital
ISSN: **2600-5859**
Vol. 5 No 2.1, pp. 45 – 68 , mayo 2022
www.concienciadigital.org

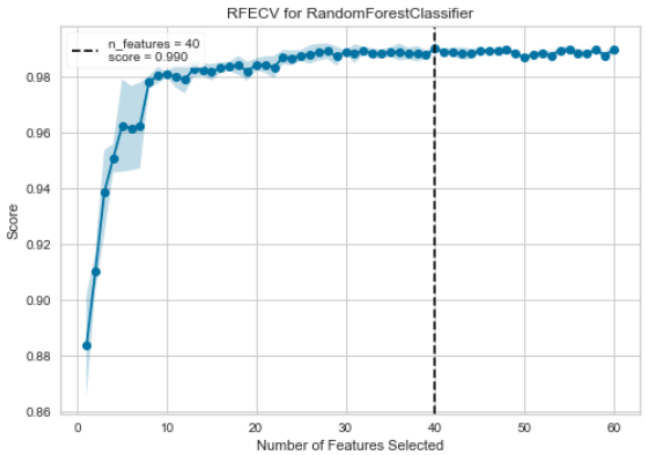
La base de datos de mantenimiento predictivo ai4i2020 cuenta únicamente con series de tiempo, por lo que mediante la ayuda de la librería TSFEL se extraen 12 características en el dominio del tiempo por cada variable de proceso; es decir se extraen un total de 60 características tanto para el conjunto de entrenamiento como para el conjunto de prueba. El tamaño de ventana es uno de los parámetros a modificar para la extracción automática de características con TSFEL, por lo que en este caso se definió un tamaño de ventana de 4, esto quiere decir que del total de datos de entrenamiento (13389) así como de prueba (5739) se dividen para 4. De esta forma la base de datos se reduce a 3346 observaciones para el entrenamiento y 1434 para la prueba. Las características que fueron extraídas, así como su fórmula se denotan en la tabla 1.

Selección de características

El proceso de selección de características se lo realiza mediante el método de envoltura RFECV (eliminación recursiva de características mediante validación cruzada), el algoritmo utilizado fue *Random Forest* dando como resultado el número óptimo de características que proporcionan el mejor rendimiento del modelo, las características seleccionadas fueron 40 por lo que 20 se procedieron a eliminar ya que son las que menos contribuyen a la predicción, la Figura 10 muestra el proceso de selección de características.

Figura 10

Número óptimo de características seleccionadas por el método RFECV



Escala y estandarización

La mayoría de los métodos de aprendizaje de máquina necesitan que los datos se encuentren en la misma escala y que se encuentren estandarizados, con la finalidad de mejorar el rendimiento. Se utiliza la función *RobustScaler* debido a que esta ayuda a reducir el impacto de valores atípicos, los cuales se evidenciaron al realizar el análisis

exploratorio, de esta forma los datos se encuentran listos para ser ingresados en el clasificador.

Entrenamiento del modelo

El entrenamiento del modelo se lo realiza con las características seleccionadas por el método RFECV que mejor rendimiento proporcionaron, a su vez se emplea los 4 algoritmos siguientes: *Random Forest*, Máquina de soporte vectorial, *XGBoost* y árbol de decisión. Se emplea 4 algoritmos con la finalidad de saber cuál es el que mejor rendimiento proporciona en la detección de fallos de la máquina. Todos los algoritmos obtienen un 100% tanto en exactitud y precisión excepto la máquina se soporte vectorial que obtiene 99,37 en exactitud y 99,27 en precisión.

Evaluación del modelo

Para evaluar el rendimiento del modelo y determinar si existe una buena generalización en datos no vistos, se utiliza las siguientes métricas de evaluación: matriz de confusión, exactitud, precisión y curva ROC-AUC.

La matriz de confusión tiene una forma de tabla, la cual define las instancias de datos que están correcta e incorrectamente clasificadas. Las columnas representan las instancias predichas por el clasificador y las filas representan los valores reales o las etiquetas de clase a las que pertenece realmente el objeto de datos (Gianey & Choudhary, 2018). A continuación, la Figura 11 muestra la matriz de confusión.

Figura 11
Matriz de confusión

		Predicted	
		0	1
Actual	0	TN	FP
	1	FN	TP

Donde cada valor de la celda de la matriz de confusión significa:

- TN: verdadero negativo, representa las instancias negativas que se clasificaron correctamente.
- TP: verdadero positivo, representa las instancias positivas que se clasificaron correctamente.
- FN: falso negativo, representa las instancias negativas que se clasificaron incorrectamente.

Conciencia Digital
ISSN: **2600-5859**
Vol. 5 No 2.1, pp. 45 – 68 , mayo 2022
www.concienciadigital.org

- FP: falso positivo, representa las instancias positivas que se clasificaron incorrectamente (Gianey & Choudhary, 2018).

Por otra parte, la exactitud y la precisión son métricas de evaluación que se derivan de la matriz de confusión, las mismas se dan a conocer a continuación en la tabla 2.

Tabla 2
Exactitud y precisión

Métrica	Fórmula
Exactitud: denota el número de predicciones correctas realizadas por el clasificador, tanto positivas como negativas.	$\frac{TN + TP}{TN + FP + FN + TP}$
Precisión: denota la tasa de positivos que se predijeron como positivos y en realidad fueron positivos.	$\frac{TP}{FP + TP}$

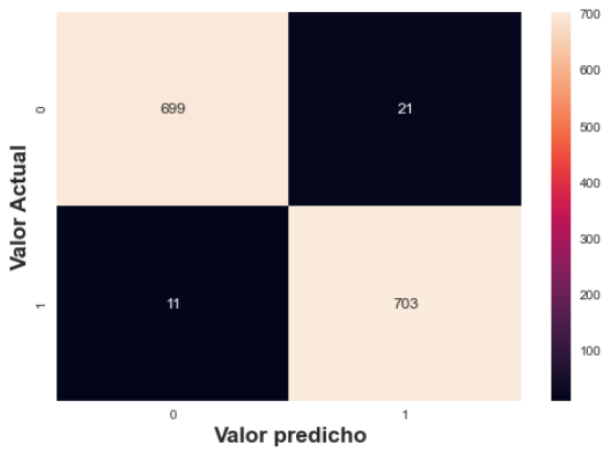
Fuente: Gianey & Choudhary (2018)

La curva de ROC representa en el eje x la tasa de falsos positivos y en el eje y la tasa de verdaderos positivos, el gráfico muestra la probabilidad de pertenecer a una determinada clase o grupo de acuerdo con cada valor de umbral posible (0 a 1). Para estimar el rendimiento del clasificador mediante la curva ROC, se determina el área bajo la curva obteniendo un puntaje (AUC-ROC). Donde se dice que un AUC-ROC de 0,5 significa que no hay discriminación, 0,7-0,8 se considera aceptable, 0,8-0,9 excelente y mayor a 0,9 sobresaliente (Vieira et al., 2019).

Resultados

La figura 12 muestra la matriz de confusión resultado de evaluar el modelo con el algoritmo de *Random* Forest, la figura 13 el algoritmo de SVM, la figura 14 el algoritmo de *XGBoost* y la figura 15 el algoritmo de árbol de decisión.

Figura 12
Matriz de confusión Random Forest



ISSN: **2600-5859**
Vol. 5 No 2.1, pp. 45 – 68 , mayo 2022

Figura 13
Matriz de confusión SVM

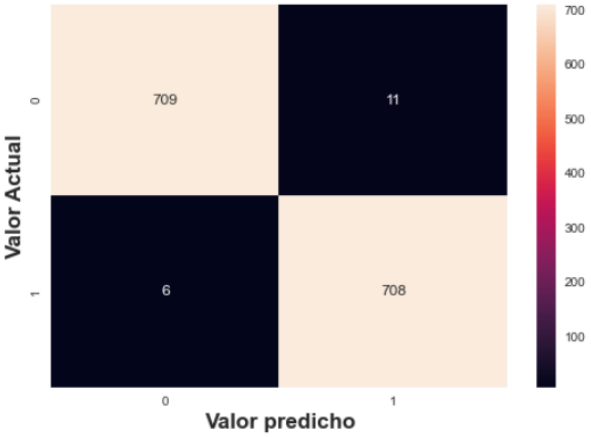


Figura 14
Matriz de confusión XGBoost

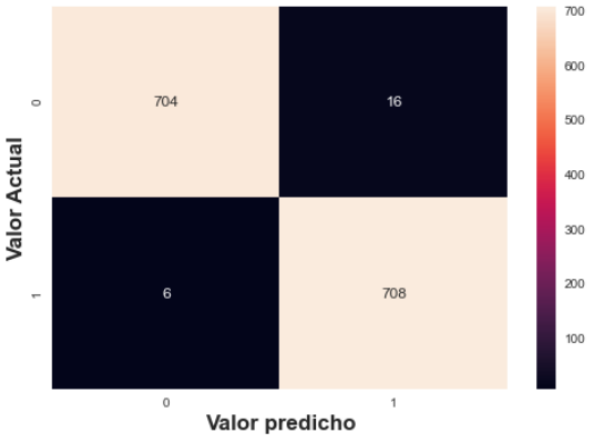
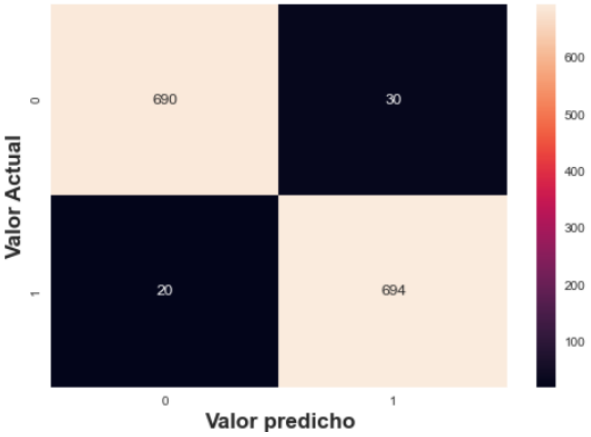


Figura 15
Matriz de confusión árbol de decisión



Conciencia Digital
ISSN: **2600-5859**
Vol. 5 No 2.1, pp. 45 – 68 , mayo 2022
www.concienciadigital.org

Los resultados de las matrices de confusión muestran que el algoritmo que mejor rendimiento proporciona con los datos de mantenimiento predictivo es SVM, esto se contrastará más adelante apreciando las métricas de exactitud, precisión y curva ROC-AUC.

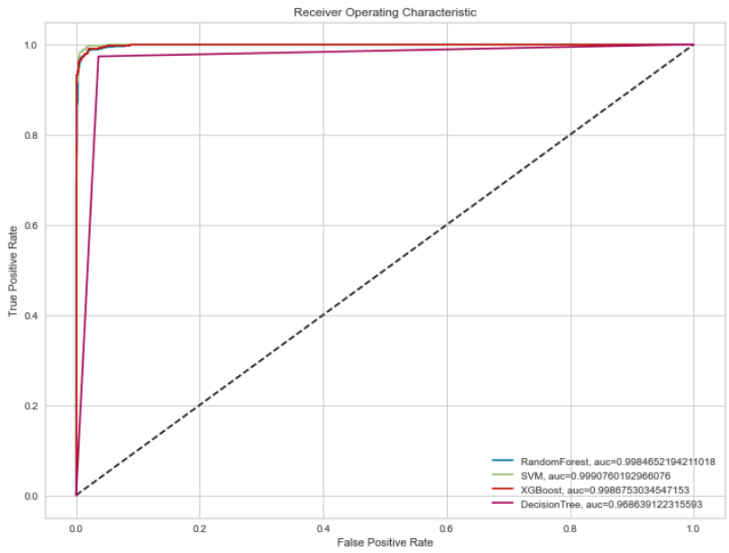
A continuación, la tabla 3 muestra el valor obtenido en exactitud y precisión por cada uno de los algoritmos analizados.

Tabla 3
Exactitud y precisión para cada algoritmo

Algoritmo	Exactitud (%)	Precisión (%)
Random Forest	97,76	97,09
SVM	98,81	98,47
XGBoost	98,46	97,79
Árbol de decisión	96,51	95,85

Apreciando los porcentajes de la exactitud y precisión se puede comprobar que efectivamente el algoritmo que mejor rendimiento proporciona es SVM, cabe acotar que estos porcentajes obtenidos son resultado de trabajar con hiperparámetros predeterminados, por lo que más adelante se realizará la optimización de estos con la finalidad de averiguar si se puede lograr un mejor rendimiento y en consecuencia notar una mejora en las métricas de evaluación de exactitud y precisión.

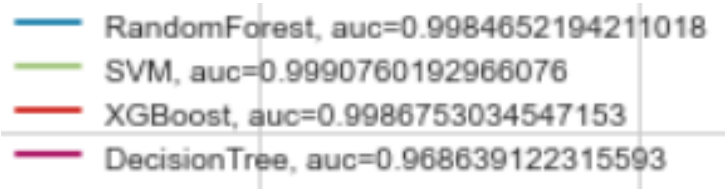
Figura 16
Curva ROC



Conciencia Digital
ISSN: **2600-5859**
Vol. 5 No 2.1, pp. 45 – 68 , mayo 2022
www.concienciadigital.org

Otra de las métricas que se emplea para evaluar el rendimiento del modelo es la curva ROC, la cual se muestra a continuación en la figura 16. Donde se puede apreciar que la curva de color verde (algoritmo SVM), es la que más se acerca a un valor de 1, a continuación, la figura 17 muestra los valores AUC-ROC parra cada algoritmo.

Figura 17
Valores AUC-ROC



La figura 17 muestra que SVM alcanza el puntaje AUC más alto con 0,9990, seguido de *XGBoost* con 0,9986, *Random Forest* con 0,9984 y finalmente árbol de decisión con 0,9686. Los puntajes AUC de todos los algoritmos muestran un rendimiento elevado, pero siempre se elegirá el que más se acerque a un valor de 1.

Optimización de hiperparámetros

Con la finalidad de mejorar el rendimiento del modelo se procedió a optimizar los hiperparámetros de los 4 algoritmos utilizados en la comparación, para ello se creó una cuadrícula de hiperparámetros para posteriormente emplear una optimización de búsqueda aleatoria mediante validación cruzada utilizando la función de *Python RandomizedSearchCV*. Para todos los casos se utilizó 3 pliegues de validación cruzada y 100 iteraciones; los resultados de la optimización de hiperparámetros se muestran en la tabla 4.

Tabla 4
Exactitud y precisión con hiperparámetros optimizados

Algoritmo	Exactitud (%)	Precisión (%)
Random Forest	98,25	97,91
SVM	98,95	98,88
XGBoost	98,67	98,60
Árbol de decisión	97,35	96,81

Visualizando la tabla 4 se puede apreciar que, para todos los algoritmos, la exactitud y la precisión elevan su porcentaje una vez optimizados los hiperparámetros, a continuación, la tabla 5 muestra el porcentaje de mejora que se alcanza al trabajar el modelo con la

optimización de hiperparámetros versus los hiperparámetros predeterminados de cada algoritmo de aprendizaje de máquina.

Tabla 5

Porcentaje de mejora con optimización de hiperparámetros vs hiperparámetros predeterminados

Algoritmo	% de mejora Exactitud	% de mejora precisión
Random Forest	0,49	0,82
SVM	0,14	0,41
XGBoost	0,21	0,81
Árbol de decisión	0,84	0,96

Notoriamente se puede apreciar en la tabla 5 que el optimizar los hiperparámetros permite mejorar el porcentaje de las métricas de evaluación, si bien es cierto que el porcentaje de mejora no es tan elevado esto contribuye a mejorar las predicciones correctas y reducir las incorrectas.

Conclusiones

- El aprendizaje de máquina resulta ser una herramienta muy útil que permite potenciar el mantenimiento predictivo; debido a que la cantidad de datos que regularmente se extrae de los procesos y máquinas es muy extensa, dificulta el análisis de los mismos el cual es regularmente llevado a cabo de manera tradicional, gracias al enfoque de aprendizaje de máquina se pueden utilizar los datos de mantenimiento predictivo para la predicción de fallas de una máquina o un proceso, de esa forma el personal encargado del mantenimiento se puede anticipar al fallo y así evitar paros inesperados lo cuales conllevan a pérdidas económicas muy cuantiosas, pérdidas que implican la no producción, daño de artículos que se estén produciendo, horas hombre, repuestos y materiales, entro otros.
- En la presente investigación se desarrolló un modelo de aprendizaje de máquina para la predicción de fallos, para lo cual se empleó 4 algoritmos con la finalidad de determinar cuál es el que mejor rendimiento proporciona, concluyendo que el algoritmo que mejor desempeño muestra con los datos de mantenimiento predictivo ai4i2020 es la máquina de soporte vectorial (SVM), esto se lo puede evidenciar en todas las métricas de evaluación analizadas anteriormente.
- Tanto la selección de características por el método de envoltura RFECV así como el sobre muestreo de datos, permitieron mejorar el rendimiento del modelo, ya que al trabajar con datos desequilibrados se corre riesgo de obtener un rendimiento

muy deficiente; a su vez la selección de características permitió elegir las que de mayor forma contribuyen a la predicción, reduciendo así de alguna manera el costo computacional, el cual se ve notablemente afectado cuando se trabaja con bases de datos de gran dimensión.

- La optimización de hiperparámetros es un paso que debe ser llevado a cabo cuando se desee tratar de mejorar el rendimiento de un modelo, para este estudio se puede concluir que el optimizar los hiperparámetros de todos los algoritmos ayudó a mejorar el rendimiento del modelo.

Agradecimientos

En primer lugar, agradecer a la Escuela Superior Politécnica de Chimborazo ESPOCH y especialmente a la carrera de Mantenimiento Industrial, la cual ha contribuido de gran forma mediante el conocimiento, para llevar a cabo de forma exitosa la presente investigación.

Referencias Bibliográficas

Ayvaz, S., & Alpay, K. (2021). Predictive maintenance system for production lines in manufacturing: A machine learning approach using IoT data in real-time. *Expert Systems with Applications, 173,* 114598.

Breiman, L. (2001). Random Forests. *Machine Learning.*

Canizo, M., Onieva, E., Conde, A., Charramendieta, S., & Trujillo, S. (2017). Real-time predictive maintenance for wind turbines using Big Data frameworks. *2017 IEEE International Conference on Prognostics and Health Management, ICPHM 2017,* 70-77.

Carvalho, T.P., Soares, F., Vita, R., Francisco, R., Basto, J.P., & Alcalá, S.G. (2019). A systematic literature review of machine learning methods applied to predictive maintenance. *Computers and Industrial Engineering, 137,* 106024.

Dey, A. (2016). Machine Learning Algorithms: A Review. *International Journal of Computer Science and Information Technologies, 7* (3), 1174-1179.

Dong, W., Huang, Y., Lehane, B., & Ma, G. (2020). XGBoost algorithm-based prediction of concrete electrical resistivity for structural health monitoring. *Automation in Construction, 114,* 103155.

Dua, D., & Graff, C. (2019). UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets/AI4I+2020+Predictive+Maintenance+Dataset

Fernandes, M., Canito, A., Corchado, J.M., & Marreiros, G. (2020). Fault Detection Mechanism of a Predictive Maintenance System Based on Autoregressive Integrated Moving Average Models. *Distributed Computing and Artificial Intelligence, 16th International Conference. DCAI 2019. Advances in Intelligent Systems and Computing, 1003,* 171-180.

Gianey, H.K, & Choudhary, R. (2018). Comprehensive Review on Supervised Machine Learning Algorithms. *Proceedings - 2017 International Conference on Machine Learning and Data Science, MLDS 2017,* 37-43.

Gohel, H.A., Upadhyay, H., Lagos, L., Cooper, K., & Sanzetenea, A. (2020). Predictive maintenance architecture development for nuclear infrastructure using machine learning. *Nuclear Engineering and Technology*, *52* (7), 1436-1442.

Jijo, B., & Abdulazeez, A. (2021). Classification Based on Decision Tree Algorithm for Machine Learning. *Journal of Applied Science and Technology Trends, 2* (1), 20-28.

Kanawaday, S., & Sane, A. (2018). Machine learning for predictive maintenance of industrial machines using IoT sensor data. *Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS,* 87-90.

Kang, H.S., Lee, J.Y., Choi, S., Kim, H., Park, J.H., Son, J.Y., Kim, B.H., & Noh, S.D. (2016). Smart manufacturing: Past research, present findings, and future directions. *International Journal of Precision Engineering and Manufacturing Green Technology*, *3* (1), 111-128.

Lee, W.J., Wu, H., Yun, H., Kim, H., Jun, M., & Sutherland, J.W. (2019). Predictive maintenance of machine tool systems using artificial intelligence techniques applied to machine condition data. *Procedia CIRP, 80,* 506-511.

Liu, Y., Wang, Y., & Zhang, J. (2012). New machine learning algorithm: Random Forests. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 743* LNCS, 246-252.

Mitchell, R., & Frank, E. (2017). Accelerating the XGBoost algorithm using GPU computing. *PeerJ Computer Science 3: e127.*

Mohammed, R., Rawashdeh, J., & Abdullah, M. (2020). Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results. *2020 11th International Conference on Information and Communication Systems, ICICS 2020,* 243-248.

Schwendemann, S., Amjad, Z., & Sikora, A. (2021). A survey of machine-learning techniques for condition monitoring and predictive maintenance of bearings in grinding machines. *Computers in Industry, 125,* 103380.

Uddin, S., Khan, A., Hossain, M.E., & Moni, M.L. (2019). Comparing different supervised machine learning algorithms for disease Prediction. *BMC Medical Informatics and Decision Making, 19* (1), 1-16.

Vieira, S., Lopez Pinaya, W.H., & Mechelli, A. (2019). Main concepts in machine learning. *Machine Learning: Methods and Applications to Brain Disorders*, 21-44.

Wan, J., Tang, S., Li, D., Wang, S., Liu, C., Abbas, H., & Vasilakos, A. V. (2017). A Manufacturing Big Data Solution for Active Preventive Maintenance. *IEEE Transactions on Industrial Informatics, 13* (4), 2039-2047.

Wuest, T., Weimer, D., Irgens, C., & Thoben, K.D. (2016). Machine learning in manufacturing: Advantages, challenges, and applications. *Production and Manufacturing Research, 4* (1), 23-45.

ISSN: **2600-5859**
Vol. 5 No 2.1, pp. 45 – 68 , mayo 2022

El artículo que se publica es de exclusiva responsabilidad de los autores y no necesariamente reflejan el pensamiento de la **Revista Conciencia Digital.**

Indexaciones