

Transformando sistemas automáticos de almacenamiento: Un primer enfoque
de mantenimiento predictivo basado en algoritmos de clasificación

Presentado por:

Jonathan Andrés Acaro Imaicela,

Ingeniero Industrial

Director

Luis Alfredo Paipa Galeano, MSc., PhD

Facultad de Ingeniería

Maestría en Analítica Aplicada



Universidad de
La Sabana

Nota de aceptación

Firma del presidente del jurado

Firma del jurado

Firma del jurado

Contenido

Lista de ilustraciones	6
Lista de tabla.....	6
Palabras clave	8
Glosario.....	8
1 Resumen.....	9
2 Introducción	9
3 Pregunta de investigación	11
4 Marco conceptual	11
4.1 Mantenimiento.....	11
4.1.1 Definición y origen	11
4.1.2 ¿Cómo implementar el TPM?	13
4.1.3 Medir el impacto del TPM	14
4.1.4 Tipos de mantenimiento.....	15
4.2 Industria 4.0.....	15
4.2.1 Definición y origen	15
4.2.2 Industria 4.0 y su relevancia en la industria	16
4.3 Internet de las Cosas (IoT).....	16
4.3.1 Definición y origen	16
4.3.2 El IoT y su relevancia en la industria	17
4.3.3 Componentes y tecnologías del IoT	18
4.4 Big Data.....	19
4.4.1 Definición y origen	19
4.4.2 Desafíos del Big Data	19
4.5 Analítica de Datos	20

4.5.1	Definición y origen	20
4.5.2	La analítica de datos y su relevancia en la industria	21
4.5.3	Enfoques de analítica	21
4.6	Aprendizaje de Máquinas (ML)	22
4.6.1	Definición y Origen.....	22
4.6.2	ML y su relevancia en la Industria	22
4.6.3	Tipos de Aprendizaje de Máquinas	23
4.7	Sistemas de almacenamiento automatizados	26
4.7.1	Definición y características	26
4.7.2	Modelo VCM	26
5	Estado del arte.....	27
5.1	Enfoques de mantenimiento predictivo	27
5.2	Variables en el contexto del mantenimiento predictivo.....	28
5.3	Preprocesamiento de datos	29
5.4	Casos de estudio de mantenimiento predictivo	30
5.5	Consideraciones finales	32
6	Objetivos	33
7	Metodología	34
8	Resultados	36
8.1	Recolección de datos	36
8.2	Tratamiento de los datos.....	37
8.2.1	Extracción	37
8.2.2	Agrupación	38
8.2.3	Preprocesamiento	39
8.2.4	Balanceo del conjunto de datos.....	41

8.3 Selección de variables	43
8.4 Aprendizaje de máquinas supervisado y automatizado	43
8.5 Recolección de nuevos datos y validación final.....	45
9 Discusión	47
10 Conclusiones y Trabajo Futuro.....	49
11 Anexos	52
12 Bibliografía.....	53

Lista de ilustraciones

Ilustración 1 Beneficios del IIoT, adaptado de (Javaid et al., 2021)	17
Ilustración 2 Tipos de ML y principales modelos, adaptado de (El Morr & Ali-Hassan, 2019).....	23
Ilustración 3 Modelo VCM (Carrusel vertical)	27
Ilustración 4 ML supervisado, adaptado de Kotsiantis (2007)	34
Ilustración 5 Lista de tablas en la base de datos	37
Ilustración 6 Tabla de datos operativos de la máquina 102007125.....	37
Ilustración 7 Exploración de algunas variables	38
Ilustración 8 Ejemplos de observaciones resultantes	40
Ilustración 9 Resumen de la base de datos resultante	41
Ilustración 10 Distribución de datos de entrenamiento	41
Ilustración 11 Puntajes f1 de cada modelo durante entrenamiento.....	45
Ilustración 12 Puntajes f1 de cada modelo durante validación final	46
Ilustración 13 Matriz de confusión del conjunto de datos 6	46
Ilustración 14 Voltaje de batería promedio, varias máquinas	48
Ilustración 15 Desequilibrio promedio a lo largo del tiempo, varias máquinas ..	49
Lista de tablas	
Tabla 1 Pilares del TPM (Parikh & Mahamuni, 2015)	13
Tabla 2 Cálculo de métricas del TPM, elaboración propia.....	14
Tabla 3 Principales modelos de ML, descripción y uso (elaboración propia)	25
Tabla 4 Resumen de casos de estudio de mantenimiento predictivo (elaboración propia).....	29
Tabla 5 Lista de las variables identificadas y tipo de agregación	39
Tabla 6 Conjuntos de datos resultantes y distribución de clases	42
Tabla 7 Lista de variables seleccionadas para cada conjunto de datos.....	43

Tabla 8 Secuencias recomendadas por NaiveAutoML44

Palabras clave

Industria 4.0, big data, machine learning, sensores, mantenimiento predictivo.

Glosario

ANN: Artificial Neural Networks, Redes neuronales artificiales

CoAP: Constrained Application Protocol, Protocolo de aplicación restringida

I4.0: Industry 4.0, Industria 4.0

IIoT: Industrial Internet of Things, Internet de las Cosas Industrial

IoT: Internet of Things, Internet de las Cosas

KNN: k-means clustering, Agrupamiento por k-medias

KPI: Key Performance Indicator, Indicadores clave de negocio

MAE: Mean Absolute Error, Error absoluto medio

ML: Machine Learning, Aprendizaje de máquinas

MQTT: Message Queuing Telemetry Transport

MTBF: Mean Time Between Failure, Tiempo promedio entre fallas

OEE: Overall Equipment Efficiency, Eficiencia general de los equipos

PCA: Principal Component Analysis, Análisis de componentes principales

R²: R-squared, R-cuadrado

RFID: Radio Frequency Identification, Identificación por radiofrecuencia

RMSE: Root Mean Squared Error, Error Cuadrático Medio

SVM: Support Vector Machines, Máquinas de vectores de soporte

TEEP: Total Effective Equipment Performance, Rendimiento efectivo total de los equipos

TPM: Total Productive Maintenance, Mantenimiento Productivo Total

VCM: Vertical Carousel Module, Módulo de Carrusel Vertical

VIF: Variance Inflation Factor, Factor de inflación de la varianza

1 Resumen

La cuarta revolución industrial ya es un hecho y ahora las organizaciones buscan incorporar aspectos como el manejo de Big Data, computación en nube e internet de las cosas con el objetivo de fortalecer su oferta de valor. El presente proyecto se realizó para una empresa global que elabora y comercializa maquinaria de almacenamiento automático. Busca identificar cuál es el mejor modelo de mantenimiento predictivo a los datos recolectados por sensores en las máquinas. El mantenimiento predictivo consiste en predecir cuándo es el mejor momento para realizar un mantenimiento, de tal manera que se minimice los eventos de falla y el tiempo de reparación asociado. En los últimos años, esta organización ha transformado su oferta en un producto alineado al concepto de industria 4.0 y a través del presente proyecto se pretende seguir haciéndolo.

A partir de los datos capturados por los sensores de máquinas en el campo y la metodología de aprendizaje de máquinas supervisado automatizado, se obtuvieron distintos modelos de clasificación. El preprocesamiento de datos requirió agrupación de los datos en ventanas de tiempo y técnicas de balanceo de clases. Los modelos obtenidos fueron evaluados con un conjunto de datos nuevo, el cual se obtuvo en un periodo de tiempo posterior a la extracción inicial de datos para entrenamiento. Entre los modelos recomendados se encontraron bosques aleatorios, árboles extremadamente aleatorios y boosting. El puntaje f1 fue utilizado para medir el desempeño de los modelos y se alcanzó un valor máximo de 0.4 en la validación final.

2 Introducción

Debido a las paradas por fallas inesperada en la maquinaria industrial, mundialmente la pérdida total anual de ingresos es aproximadamente de 450 billones de dólares (Mohan et al., 2021). Una falla afecta el cronograma establecido, demora la entrega y genera una necesidad de horas extra de trabajo para compensar lo sucedido (Tortorella et al., 2021). Los tiempos muertos ocasionados pueden afectar la calidad del producto y la productividad

del sistema, generando sobre costos para dar solución al problema (Ciancio et al., 2020).

Considerando que las fallas inesperadas son eventos que ocurren todo el tiempo, el mantenimiento productivo total (TPM por sus siglas en inglés) nace como un enfoque diseñado para maximizar la eficacia de un equipo a lo largo de su ciclo de vida (Tortorella et al., 2021). Según el pionero del TPM Nakajima (1984), este concepto contempla maximizar la efectividad de los equipos a través de un sistema completo de mantenimiento e involucra a los departamentos y empleados de todos los niveles para lograr eficiencia económica.

Durante los últimos años, el mantenimiento ha sido transformado gracias a los avances tecnológicos de los sistemas de información y comunicación (Roda & Macchi, 2021). Pasó de un enfoque correctivo (cuando la falla ya ocurrió) a un enfoque predictivo, este último caracterizado por el uso de información para evaluar el estado actual y futuro de las máquinas (Ciancio et al., 2020).

Roda & Macchi (2021) sostienen que el mantenimiento basado en tecnología y la industria 4.0 están estrechamente relacionados. Según Mohan et al., (2021), la industria 4.0 se trata de un enfoque industrial que busca vincular maquinaria y personas para un intercambio de información más rápido a través de sistemas inteligentes basados en la web. Se espera que la integración de este tipo de sistemas traiga mejoras significativas en los productos, procesos y el rendimiento de la organización (Tortorella et al., 2021).

El problema de paradas de maquina no es ajeno a la organización objeto de estudio para este trabajo. Se trata de una compañía suiza líder en soluciones de almacenamiento automatizado y sistemas de manipulación de materiales (Kardex, 2021a). El modelo VCM, también conocido como módulo de carrusel vertical, es el producto estrella de la firma. Existen alrededor de 500 unidades en el mundo (Kardex, 2020).

Para esta compañía, según el reporte del servicio post-venta en 2020, se registraron un total de 191 eventos de falla entre los 56 clientes conectados y ubicados en Europa. A estos eventos de falla se asocia una estimación de 71

días de interrupción en la operación (Kardex, 2020). Es difícil estimar con exactitud el perjuicio económico asociado pero los altos directivos de la compañía consideran que el mantenimiento predictivo podría ayudar a reducir las cifras antes mencionadas.

Dado el impacto que representa las paradas por fallas inesperada para los clientes de esta empresa, el presente proyecto busca desarrollar un enfoque de mantenimiento predictivo usando las variables de los sensores de la máquina modelo VCM.

3 Pregunta de investigación

¿Cuál es el mejor modelo de aprendizaje de máquinas supervisado que permita prever fallas en el modelo VCM con los datos de los sensores disponibles?

4 Marco conceptual

En este capítulo se abordarán los conceptos relacionados con mantenimiento, Industria 4.0, Internet de las Cosas (IoT), Big Data y analítica. Desde su origen, sus características más importantes y la relación que existe entre cada uno. Así mismo, se brindará más detalles del modelo VCM, objeto de estudio del presente proyecto.

4.1 Mantenimiento

4.1.1 Definición y origen

En el ámbito industrial y según Yan et al. (2016), el mantenimiento se define como un grupo de actividades que restauran y preservan la condición de un equipo, para garantizar una operación continua y de calidad. El mismo autor establece que la degradación de una máquina, debido a su uso, es algo inminente y puede llegar a ocasionar eventos de falla inesperados, pérdidas de tiempo considerables y costos de reparación no planificados.

Ante la presencia de eventos de falla, en 1984, Seiichi Nakajima introduce el concepto de Mantenimiento Productivo Total (TPM), el cual pretende aumentar la eficiencia de los equipos, reduce la variación y disminuye costos asociados

(Nakajima, 1984). Según Tortorella et al. (2021), en la década de los 90's este concepto tuvo una gran acogida en economías emergentes en donde se buscaba alcanzar eficiencia y un uso adecuado de los equipos.

4.1.2 ¿Cómo implementar el TPM?

Según Smith & Mobley (2008), las siguientes actividades son indispensables para la implementación del TPM: manejar un sistema de mantenimiento, uso del recurso humano para inspecciones, reparaciones y calibraciones, mantenimiento diario, actividades en grupo y entrenamiento continuo. Parikh & Mahamuni, (2015) plantean que TPM puede ser estructurado en 8 pilares, como se aprecia en la Tabla 1.

Tabla 1 Pilares del TPM (Parikh & Mahamuni, 2015)

Pilar	Descripción
Mantenimiento Autónomo	Se busca desarrollar el empoderamiento del operario a través de tareas cotidianas para desarrollar habilidades y dominio sobre los equipos. Los operarios son responsables de sus equipos.
Mantenimiento Enfocado	Su pretende reducir las pérdidas en el lugar de trabajo y así mejorar la eficacia de la planta. Esto es posible gracias a pequeños cambios en distintas áreas.
Mantenimiento Planificado	Este pilar sigue el concepto de cero fallas y se asocia al aumento de la producción de la planta. Se busca aumentar la disponibilidad de las máquinas a través de distintos enfoques de mantenimiento.
Mantenimiento de Calidad	Este pilar busca una fabricación sin errores para satisfacer al cliente. El objetivo es identificar qué partes del equipo afectan la calidad y corregir esos errores de una forma sistemática antes de que empeoren.
Educación y Entrenamiento	Se busca convertir a los empleados en una fuerza de trabajo dinámica, motivada, independiente y competente en distintas áreas. Esto es posible gracias a actividades de formación de habilidades.
Seguridad, Salud y Entorno	El objetivo de este pilar es desarrollar un lugar de trabajo y entorno más seguros para los operarios. La seguridad tiene mucha importancia dentro de una planta.
TPM en la Oficina	Se tiene como objetivo mejorar la productividad y eficiencia de las funciones administrativas al identificar y eliminar ineficiencias. Está relacionado con las áreas de contabilidad, compras, ventas y marketing.
Gestión del Desarrollo	Este pilar busca aprovechar los aprendizajes del resto de pilares para desarrollar prácticas de mantenimiento bajo un nuevo sistema.

Tortorella et al. (2021) establecen muchas organizaciones enfrentan dificultades para implementar el TPM exitosamente. Para una implementación exitosa recomiendan las 5S, técnica de gestión asociada a principios de clasificación, orden, limpieza, estandarización y disciplina. Además, la comunicación constante con los empleados y la colaboración de todo el personal son actividades indispensables para implementar TPM (Nakajima, 1984).

4.1.3 Medir el impacto del TPM

Existen métricas como la eficiencia general de equipos (OEE por sus siglas en inglés) y el rendimiento efectivo total de los equipos (TEEP) que sirven para medir los resultados alcanzados por las actividades de TPM (Tortorella et al., 2021). El OEE es un porcentaje que considera la disponibilidad de un equipo, eficiencia en su rendimiento y el nivel de calidad, aspectos clave que se busca mejorar (Smith & Mobley, 2008b). El TEEP mide el desempeño de un equipo al considerar tasas de producción y pérdidas (Prayogo, 2020). Adicionalmente, el tiempo medio entre fallas (MTBF) es una métrica enfocada en estimar cuánto tiempo trabajará un equipo antes de que falle (Smith & Mobley, 2008a). En la Tabla 2 se indica el cálculo de estas métricas.

Tabla 2 Cálculo de métricas del TPM, elaboración propia

Métrica	Cálculo
OEE	$= \text{disponibilidad} \times \text{eficiencia} \times \text{calidad}$ $= \frac{\text{tiempo de operación}}{\text{tiempo disponible}}$ $\times \frac{\text{producción total}}{\text{tiempo de ciclo} \times \text{tiempo de operación}}$ $\times \frac{\text{producción total} - \text{unidades defectuosas}}{\text{producción total}}$
TEEP	$= \text{OEE} \times \text{utilización}$
MTBF	$= \frac{\text{tiempo disponible} - \text{tiempo de fallas}}{\text{número de fallas}}$

4.1.4 Tipos de mantenimiento

En la actualidad, es posible clasificar los tipos de mantenimiento de la siguiente forma: correctivo, preventivo y predictivo (Zonta et al., 2020).

El mantenimiento correctivo actúa después de un fallo y su principal desventaja es que la operación se ve interrumpida y se pueden desencadenar consecuencias considerables (Yan et al., 2016). Este enfoque implica operar normalmente hasta que una falla se presente y en cuyo caso será necesario reemplazar el subsistema o componente afectado para reanudar la operación (Erkoyuncu et al., 2017).

El mantenimiento preventivo consiste en realizar inspecciones, pruebas, ajustes y reemplazo de piezas en intervalos de tiempo o de operación que dependen del contexto (Erkoyuncu et al., 2017). Mena et al. (2021) establecen que para mantener el sistema funcionando bajo este enfoque se requiere del diseño, planificación y ejecución de un programa de mantenimiento. Sin embargo, este enfoque puede asociarse al desperdicio de recursos, que podrían ser reemplazados aun cuando tenían tiempo de vida restante (Yan et al., 2016).

Finalmente, el mantenimiento predictivo corrige las desventajas del preventivo al tratar de identificar el momento exacto en el que una falla ocurrirá (Ayvaz & Alpay, 2021). La reducción de mantenimientos no programados, paros en la producción, los tiempos improductivos asociados y los costos asociados son algunos de sus beneficios (Killeen et al., 2019). Este enfoque, se puede apreciar en entornos de Industria 4.0, tales como líneas de producción, en donde existen sensores que monitorean las condiciones de un equipo o componente a lo largo del tiempo, capturan datos y se busca generar aprendizajes (Zonta et al., 2020).

4.2 Industria 4.0

4.2.1 Definición y origen

Industria 4.0 (I4.0) se define como la automatización de sistemas de manufactura gracias a tecnologías y herramientas de vanguardia (Shet &

Pereira, 2021). Por su parte, Tay et al., (2021) la definen como un sistema de manufactura integrado que se adapta en tiempo real a las condiciones de la fábrica, la cadena de suministro y las necesidades del cliente.

Según Xu et al. (2021), este término está asociado a una cuarta revolución industrial, debido a los avances tecnológicos que trae consigo y como consecuencia de las revoluciones industriales anteriores. El concepto de Industria 4.0 fue mencionado por primera vez en el año 2011 en la feria tecnológica de Hannover, bajo una iniciativa tecnológica del gobierno alemán (Silvestri et al., 2020). Desde ese momento, se ha estimulado el debate sobre I4.0 y sus tecnologías de manufactura, en distintas esferas de la sociedad (Pfeiffer, 2017).

4.2.2 Industria 4.0 y su relevancia en la industria

Las áreas como manufactura, compras, cadena de suministro, logística y operaciones se ven beneficiadas de la I4.0 debido a la integración y el flujo de datos existente (Tay et al., 2021). Las empresas buscan migrar hacia esta revolución puesto que permite afrontar los recientes cambios económicos, sociales y culturales que el mundo atraviesa (Ruiz-sarmiento et al., 2020). Algunos de estos cambios incluyen cadenas de suministro complejas, productos muy personalizados y mercados más ágiles (Tortorella et al., 2021).

Las tecnologías que trae consigo la industria 4.0, también conocidas como pilares, incluyen Internet de las cosas (IoT), Big Data y analítica, simulación, computación en la nube, realidad aumentada, robots autónomos y manufactura aditiva (Silvestri et al., 2020). En el contexto del mantenimiento, el uso de IoT y computación en la nube permiten un monitoreo más eficiente de las condiciones operativas de un equipo (Tortorella et al., 2021).

4.3 Internet de las Cosas (IoT)

4.3.1 Definición y origen

Según Ashima et al. (2021), el IoT se define como la capacidad de comunicación de dispositivos, sensores y computadores, envío de datos y monitoreo en tiempo real. Andersen et al. (2020) la definen como un medio

tecnológico que cambia principalmente la vida de las personas, su forma de relacionarse y trabajar con su entorno, a través de actuadores, sensores, hardware y software conectados entre sí.

El IoT fue mencionado por primera vez en 1999 en el Massachusetts Institute of Technology (MIT) por Kevin Ashton, director de investigación (Ashima et al., 2021), quien también trabajaba para Procter & Gamble (P&G). Él reconoció el potencial de la tecnología RFID (identificación por radiofrecuencia) para el manejo e identificación de los componentes a lo largo de la cadena de suministro de P&G. De esta manera, estableció que el RFID era un componente esencial del IoT.

4.3.2 El IoT y su relevancia en la industria

Un concepto más reciente y similar es el IIoT (Industrial Internet of Things), presente en el ámbito industrial. Según Javaid et al. (2021), se trata de una red de dispositivos, tales como sensores y actuadores, conectados entre sí para monitorear, recolectar, analizar datos. El objetivo es arrojar aprendizajes que soporten la toma de decisiones de una organización.



Ilustración 1 Beneficios del IIoT, adaptado de (Javaid et al., 2021)

De los beneficios del IIoT mostrados en la Ilustración 1 se puede apreciar que estos permiten optimizar los procesos industriales y de manufactura para alcanzar excelencia operativa. Según Javaid et al. (2021), a través de las tecnologías involucradas en IIoT, se pretende alcanzar operaciones exitosas y se involucra a diferentes sistemas, equipos y personas. Esto es posible gracias al análisis de datos, el cual permite tomar decisiones organizacionales más rápidas y mejores.

De esta manera, áreas como salud, vivienda, granjas, ciudades y transporte inteligente se han visto beneficiadas por el IoT (Killeen & Parvizimosaed, 2018). Un ejemplo es la agricultura, donde el monitoreo de cultivos puede aumentar el rendimiento y productividad de las cosechas (Gupta & Nahar, 2022). Los sistemas de redes eléctricas son otro ejemplo, puesto que el monitoreo del estado del transformador puede reducir la ocurrencia de fallas y los apagones asociados (Memala et al., 2021). Se estima que para el 2025 el IoT podría generar entre 2,7 y 6,4 billones de dólares anuales, al conectar entre 8 y 25 billones de dispositivos y crear una red masiva de información (Andersen et al., 2020).

4.3.3 Componentes y tecnologías del IoT

Killeen & Parvizimosaed (2018) establecen que la arquitectura del IoT se puede categorizar en los siguientes componentes (también conocidos en la literatura como capas): sensorial, redes, almacenamiento, aprendizaje y aplicación. La capa sensorial captura datos del entorno e interactúa a través de sensores y actuadores. Luego, la capa de redes se encarga de conectar los flujos de datos de los dispositivos con la capa de almacenamiento, la cual está enfocada en procesar, agrupar y almacenar los datos capturados. La capa de aprendizaje hace analítica de los datos almacenados y busca aprendizajes sobre el entorno. Finalmente, la capa de aplicación permite el acceso y el control sobre el resto de las capas.

Según Killeen & Parvizimosaed (2018), cada capa del IoT puede contar con una o más tecnologías específicas, las cuales se describen a continuación. MQTT (Message Queing Telemetry Transport) y CoAP (Constrained

Application Protocol) son protocolos de comunicación entre dispositivos, basado en un modelo de suscripción y publicación. Las puertas de enlace (Gateway) permiten la conexión entre dispositivos y el internet. La computación en niebla (Fog Computing) consiste en un procesamiento de datos en la fuente antes de almacenarlos en la nube.

4.4 Big Data

4.4.1 Definición y origen

Gandomi & Haider (2015) definen al Big Data como el crecimiento masivo de datos asociados a transacciones en una cadena de suministro. Para Andersen et al. (2020), es una iniciativa clave dentro del contexto de IoT derivada de la proliferación de sensores y dispositivos inteligentes en la industria. En esto coinciden Killeen et al. (2019), quienes afirman que la presencia de Big Data es una consecuencia de tener dispositivos que capturan datos y conectados al internet.

Según Gandomi & Haider (2015), el concepto del Big Data es relativamente nuevo, su origen es incierto y pudo haber sido mencionado por primera vez en la década de los 90's en Silicon Graphics Inc. Los mismos autores afirman que el uso del término ha crecido considerablemente desde 2011. Finalmente, establecen que el revuelo sobre este concepto se debe a las iniciativas de promoción de IBM y empresas tecnológicas que han invertido en el desarrollo del mercado analítico.

4.4.2 Desafíos del Big Data

Sawalha & Al-Naymat (2021) establecen que el Big Data afronta los siguientes desafíos: variedad, volumen y velocidad. La variedad, producto de la gran cantidad de dispositivos que capturan y envían datos y que crece cada año. El volumen, ocasionado por los intervalos de captura de los dispositivos, que pueden ser hasta de 1 segundo. Y finalmente la velocidad, la cual se ve afectada al manejar y analizar enormes cantidades de datos en tiempo real.

Gandomi & Haider (2015) mencionan que la veracidad, variabilidad y valor, son desafíos que también deben considerarse al hablar de Big Data, y

complementan los antes mencionados. La primera, está asociada a la falta de fiabilidad que algunas fuentes de datos pueden tener. La segunda, relacionada con la tasa del flujo de datos, cómo varía en el tiempo, sus picos y valles periódicos. Y la tercera, la cual establece que el valor de los datos es bajo en relación con su volumen.

Ante los desafíos relacionados con el Big Data, Sawalha & Al-Naymat (2021) establecen que se requiere de nuevos enfoques para manejar datos. El almacenamiento en la nube es una opción novedosa en para almacenar grandes volúmenes de datos gracias a la escalabilidad y velocidad que ofrece (Nachiappan et al., 2017). El siguiente paso es generar valor a partir de los datos y transformarlos en conocimiento, para lo cual se requiere de un procesamiento y posterior análisis (Lee et al., 2015).

4.5 Analítica de Datos

4.5.1 Definición y origen

La analítica de datos es una disciplina que analiza grandes volúmenes de datos a través de herramientas sofisticadas y técnicas para extraer aprendizajes, principalmente en beneficio de empresas (Sivarajah et al., 2017). Una definición similar es la de Gandomi & Haider (2015), quienes establecen que la analítica de datos es un subproceso del Big Data, el cual pretende arrojar aprendizajes.

Según Li et al. (2022), su origen es consecuencia de un número cada vez mayor de empresas que han desplegado iniciativas de análisis de datos. La revisión literaria de Zhang et al. (2021) sobre analítica de datos, identificó que desde el 2006 este tema empezó ganar relevancia en la academia y a partir del 2016, las investigaciones relacionadas crecieron considerablemente. Los mismos autores también hallaron que el IoT, el Big Data y la Inteligencia de Negocios (BI por sus siglas en inglés) se relacionan estrechamente con la analítica.

4.5.2 La analítica de datos y su relevancia en la industria

Según Javaid et al. (2021), el propósito de la analítica de datos es arrojar aprendizajes que soporten la toma de decisiones organizacionales. Para Li et al. (2022), la analítica busca desarrollar una visión crítica que pueda proporcionar una ventaja competitiva a las organizaciones. Según Ayvaz & Alpay (2021), la analítica juega un rol fundamental en la innovación de una industria al proveer soluciones a problemas complejos, tales como incrementar la eficiencia, reducir costos y detectar riesgos y fallas. La analítica permite mejorar procesos internos y la eficiencia organizacional al identificar oportunidades de mejora (Awan et al., 2021).

Todo esto es posible gracias a distintas técnicas que permiten comprender patrones, generar aprendizajes e identificar tendencias en los datos (Andersen et al., 2020). Gandomi & Haider (2015) establecen que estas técnicas pueden estar centradas en los siguientes temas: análisis de texto, para extraer información de redes sociales, emails, encuestas y noticias; análisis de audio, centrado en mejorar la experiencia del cliente en áreas como salud y call centers; análisis de video, relevante en temas de seguridad y sistemas de vigilancia; y, análisis predictivo, presente varias áreas, como la industrial, al tratar de predecir el momento de falla de un motor.

4.5.3 Enfoques de analítica

El Morr & Ali-Hassan (2019) identifican dos enfoques posibles para la analítica: descriptiva y predictiva-prescriptiva, y los describen de la siguiente forma.

La descriptiva, también conocida como Inteligencia de Negocios, hace un análisis de lo que ya ha ocurrido y es reactiva. Pretende responder las preguntas “¿Qué ha pasado?”, “¿Cuándo?”, “¿Qué tanto?”. Esto lo hace a través de indicadores de desempeño (KPI's por sus siglas en inglés), métricas, alertas y tableros de control. Únicamente utiliza datos en forma de valores numéricos, que pueden estructurarse en tablas, filas y columnas.

Por otro lado, la analítica predictiva-prescriptiva mira hacia el futuro y busca dar respuesta a la pregunta “¿Qué va a pasar?”. Para ello, utiliza datos históricos, ya sean numéricos, de texto, audio, imágenes y video. Con ellos se construye

modelos para hacer predicciones. Gandomi & Haider (2015) afirman que las técnicas de analítica predictiva-prescriptiva están basadas en métodos estadísticos y también son conocidas como aprendizaje de máquinas (ML).

4.6 Aprendizaje de Máquinas (ML)

4.6.1 Definición y Origen

El aprendizaje de máquinas (ML por sus siglas en inglés) se define como una técnica de analítica que se enfoca en el uso de datos y algoritmos para imitar el cómo aprenden los humanos, mejorando gradualmente su precisión (IBM, 2020). Una definición similar es la de Bowler et al. (2022), quienes definen al ML como una técnica que utiliza algoritmos para aprender soluciones a tareas sin necesidad de instrucciones específicas. El Morr & Ali-Hassan (2019) lo definen como un proceso automático que detecta patrones ocultos en los datos y crea un modelo que permite hacer predicciones.

Según Fradkov (2020), su origen se remonta a la década de los 60's en la Universidad de Cornell, en donde se creó una máquina capaz de identificar las letras del alfabeto. En los años siguientes, creció un interés particular por desarrollar sistemas de reconocimiento similares. Las redes neuronales eran el modelo preferido para este propósito. Luego, la primera década del siglo XXI fue un punto de inflexión para el ML, gracias a las tendencias del Big Data y la creciente capacidad de procesamiento de los computadores. El mismo autor afirma que en ese punto el uso del ML se convirtió de una necesidad más que curiosidad para los científicos.

4.6.2 ML y su relevancia en la Industria

Gouveia & Costa, (2022) afirman que las empresas buscan mejorar el valor agregado que entregan a sus clientes. Para ello, recurren al uso de recursos tecnológicos como IoT, Big Data, computación en la nube y ML. Zhang et al. (2021) destacan este último, el ML, y afirman que ayuda a percibir el entorno al analizar los datos disponibles, para encontrar solución a un problema en específico al mismo tiempo que se aprende de él.

Según Zhang et al., (2021), el ML ha apoyado a distintos campos de la industria, tales como operaciones, finanzas, marketing y manejo de recursos humanos, en la toma de decisiones precisas. Shankarrao Patange & Bharatkumar Pandya (2022) coinciden y afirman que esta técnica ha ganado relevancia en negocios de manufactura. Particularmente, los problemas que se busca resolver son diagnóstico de fallas, aseguramiento de la calidad y seguridad en el trabajo. Bowler et al. (2022) destaca el uso de ML en aplicaciones más específicas, relacionadas al monitoreo de procesos para resolver problemas asociados a limpieza, mezcla, extrusión y moldeo por inyección.

4.6.3 Tipos de Aprendizaje de Máquinas

Bowler et al. (2022), plantean dos tipos principales de ML, supervisado y no supervisado como se presenta en la Ilustración 2.

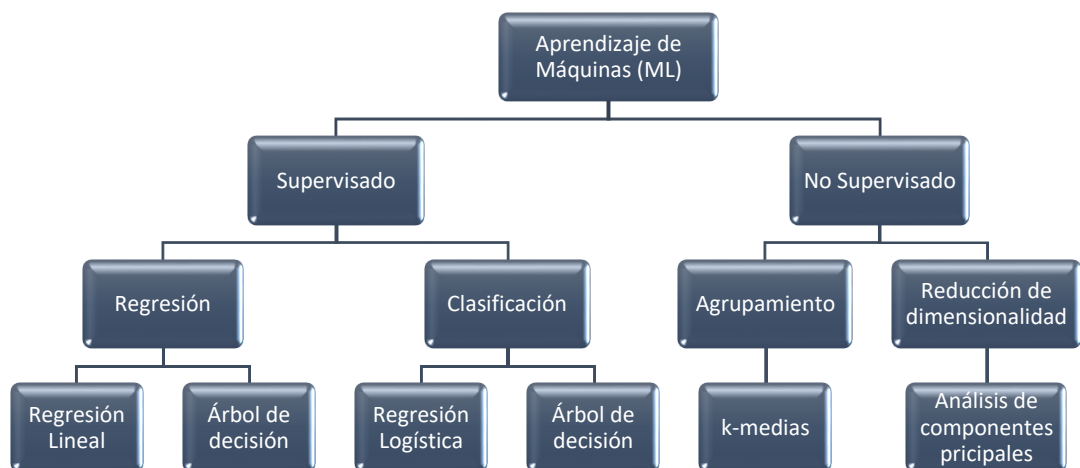


Ilustración 2 Tipos de ML y principales modelos, adaptado de (El Morr & Ali-Hassan, 2019)

El ML supervisado identifica una función entre variables de entrada asociadas a una variable de salida (Bowler et al., 2022). Para cada observación, se conoce su valor de salida. Según Zaki & Meira (2020), el ML supervisado se divide en dos enfoques, uno basado en regresión y otro basado en clasificación. En el primero, se pretende predecir un valor numérico a partir de una o más variables de entrada. Mientras que, en el segundo, se busca clasificar las observaciones según las variables de entrada y se asigna una variable categórica de salida

(conocida como etiqueta o clase), que identifica observaciones de un mismo grupo.

En ambos casos existen métricas para evaluar qué tan buenos son los modelos de ML supervisado. En el caso de regresión las métricas más comunes son: R^2 , que es la proporción de la variabilidad explicado por el modelo; el error absoluto medio (MAE por sus siglas en inglés), que mide la diferencia media entre las predicciones y los valores verdaderos; y el error cuadrático medio (RMSE por sus siglas en inglés), el cual mide la desviación estándar de los errores en la predicción (Ayvaz & Alpay, 2021).

En el caso de la clasificación, las métricas más comunes son: exactitud, fracción del total de observaciones clasificadas correctamente; precisión, la fracción de casos verdaderamente positivos de todos los casos que el modelo predijo como positivos; exhaustividad (conocida en inglés como *recall*), fracción de casos positivos clasificados como positivos; y el valor F (conocido en inglés como *F1-score*), el cual combina las métricas de precisión y exhaustividad (Erickson & Kitamura, 2021).

Por otro lado, el ML no supervisado utiliza únicamente variables de entrada para buscar patrones en los datos o reducir su dimensionalidad (Bowler et al., 2022). Esto sucede porque las observaciones no tienen una etiqueta asignada. El propósito de los modelos es predecir una posible etiqueta para observaciones nuevas (El Morr & Ali-Hassan, 2019). Existen dos enfoques principales, el agrupamiento (conocido como *clustering* en inglés) y la reducción de dimensionalidad. El primero, tiene como objetivo agrupar observaciones similares. En el segundo, se busca reducir la cantidad de variables de entrada y simplificar el conjunto de datos.

Como se puede apreciar en la Tabla 3, dentro del ML supervisado la mayoría de los modelos pueden ser utilizados tanto para regresión como clasificación. Además, se indica un resumen de los principales modelos de ML supervisado y no supervisado, una breve explicación de su funcionamiento y cuál es su propósito.

Tabla 3 Principales modelos de ML, descripción y uso (elaboración propia)

Tipo	Modelo	Descripción	Objetivo	Referencia
Supervisado	Regresión lineal	Predecir un valor numérico a partir de una o más variables de entrada.	Predicción	(Zaki & Meira, 2020)
	Regresión logística	Predecir un valor binario a partir de una o más variables de entrada.	Clasificación	(Tripepi et al., 2008)
	Árbol de decisión	Los datos son divididos en subgrupos de forma recursiva a través de un conjunto de reglas. Se usa cuando se pretende segmentar miembros de una clase o estratificar distintas categorías.	Predicción y Clasificación	(IBM, 2021b)
	Bosques aleatorios	Extensión del modelo de árbol de decisión. Se construyen varios árboles con distintas variables y se ponderan los resultados de cada árbol.	Predicción y Clasificación	(Medina-Merino & Ñique-Chacón, 2017)
	Máquina de vectores de soporte (SVM)	Los datos se proyectan en un espacio dimensional, se detecta un separador entre categorías y se transforman las observaciones. Nuevas observaciones son clasificadas de acuerdo al separador identificado.	Predicción y Clasificación	(IBM, 2021)
	K vecinos cercanos (KNN)	Se utiliza la distancia entre las observaciones en el espacio para determinar la clase o el valor de nuevas observaciones a través de un procedimiento de votación.	Predicción y Clasificación	(Bowler et al., 2022)
No supervisado	Redes neuronales artificiales (ANN)	Las observaciones fluyen desde una capa de entrada, pasan por capas ocultas y llegan a una capa de salida. En este trayecto se ajustan parámetros de forma iterativa con el objetivo de reducir el error de la predicción.	Predicción y Clasificación	(Bowler et al., 2022)
	Agrupamiento k-medias	Agrupar observaciones con características en común. Se define un número determinado de grupos y cada iteración busca reducir la distancia del centro de cada grupo hacia cada observación.	Clustering	(Bowler et al., 2022)
	Análisis de componentes principales (PCA)	Simplifica conjuntos de datos con gran cantidad de variables de entrada en componentes principales. Esto se logra al proyectar las observaciones en nuevas dimensiones en el espacio.	Reducción de Dimensionalidad	(Bowler et al., 2022)

En este subcapítulo se han descrito los principales tipos de ML, supervisado y no supervisado. La diferencia principal consiste en conocer o no el valor de la variable de salida de las observaciones (El Morr & Ali-Hassan, 2019). En el caso del ML supervisado, se busca predecir un valor de salida a partir de variables de entrada y este valor puede ser numérico o categórico (etiqueta). Mientras que en el ML no supervisado no se conoce el valor de salida de las observaciones, y los modelos serán quienes asignen este valor.

4.7 Sistemas de almacenamiento automatizados

4.7.1 Definición y características

Los sistemas automatizados de almacenamiento y recuperación se definen como una máquina de operaciones intralogísticas que almacena y entrega elementos de forma automática (Kardex, 2021b). Son considerados una solución innovadora porque optimizan el espacio ocupado al crecer hacia lo alto, llevan un mejor control en inventarios, maximizan la productividad y optimizan las operaciones de almacenamiento en un almacén.

Existen distintos modelos de este tipo de maquinaria y el principio de optimización de espacio se mantiene. Las características principales de los modelos son capacidad de almacenamiento (número de posiciones) y el nivel de desempeño que se espera alcanzar (recuperación de materiales por hora) (Kardex, 2021b). Estas características dependerán del segmento industrial.

4.7.2 Modelo VCM

Uno de los productos estrella de la organización es el sistema de almacenamiento automático vertical modelo VCM (Kardex, 2021b). En la Ilustración 3, se aprecia una imagen de este modelo. Es usado en sectores como la producción, distribución y operaciones de almacén. Cuenta con una capacidad de carga de entre 180 hasta 650 kg por estantería. Su capacidad de almacenamiento es de 1.000 posiciones y un desempeño de 150 materiales por hora. Los distintos niveles, ubicados tanto al frente como en la parte posterior, giran verticalmente como si se tratara de un carrusel.

Al girar todo el tiempo, uno de los problemas más comunes en este equipo es el desajuste de la cadena, lo cual impacta en la velocidad del equipo y el desgaste de otros componentes. Para corregir este y otros errores más específicos, se recomienda al cliente programar una cita de mantenimiento por lo menos una vez al año.

De acuerdo con la filosofía de Industria 4.0 que tiene la organización, las nuevas versiones de este modelo ahora vienen incorporadas con sensores. Estos sensores capturan y envían datos relacionados a la productividad, estado

y condición de la máquina. Han sido almacenados en la nube desde hace más de un año y aún no se ha generado ningún valor con ellos.



Ilustración 3 Modelo VCM (Carrusel vertical)

5 Estado del arte

En este capítulo se abordan los enfoques utilizados para dar solución al problema del mantenimiento predictivo. Industrias en donde se llevaron a cabo casos de estudio, las variables medidas más comunes, técnicas de preprocesamiento de datos, así como los modelos específicos que se usaron, son algunos detalles que serán descritos.

5.1 Enfoques de mantenimiento predictivo

La revisión literaria de Zonta et al. (2020) identificó tres enfoques que se le puede dar al problema de mantenimiento predictivo, físico, de conocimiento y basado en datos. A continuación, se describe cada uno.

El enfoque físico se basa en el monitoreo de la condición de un componente o equipo y requiere un registro de las fallas registradas (Zonta et al., 2020). Según Tinga & Loendersloot (2019), se busca modelar el mecanismo de falla, ya sea el nivel de fatiga o desgaste, a través de un modelo matemático. Los autores reconocen que el desarrollo de un modelo de este tipo requiere de

mucho tiempo. Destacan además que no se requiere de un conjunto de datos grande para determinar el modelo.

El enfoque basado en conocimiento reduce la complejidad del enfoque anterior y es considerado una solución híbrida. Es común en contextos de monitoreo de alertas y puede contar con modelos estadísticos o estar enfocado en monitorear las condiciones de un componente particular. Se menciona el uso del conocimiento de expertos quienes formalizan qué componentes deben analizarse (Dalzochio et al., 2020) o bien recomiendan métodos para mejorar las actividades de mantenimiento de un equipo o componente (Tinga & Loendersloot, 2019).

Finalmente, el enfoque, basado en datos, generalmente recolectados por sensores y dispositivos en el contexto de la Industria 4.0. Dalzochio et al. (2020) plantea que el enfoque basado en datos es preferido al desarrollar un modelo de mantenimiento predictivo. Según Zonta et al. (2020), este enfoque puede tener dos objetivos. Estimar la vida útil restante de un equipo o componente, o bien monitorear su condición. Así mismo, Ayvaz & Alpay (2021) destacan el enfoque basado en datos y su relevancia. Estos autores establecen que las técnicas más comunes son modelos de ML supervisada y ML no supervisado.

5.2 Variables en el contexto del mantenimiento predictivo

En términos de variables usadas en los modelos de mantenimiento predictivo, la literatura muestra que el tipo de sensores utilizados y las variables medidas dependen tanto del tipo de industria como del contexto de cada problema, como se aprecia en la Tabla 4.

Tabla 4 Resumen de casos de estudio de mantenimiento predictivo (elaboración propia)

Detalles		Autor		(Killeen et al., 2019)	(Betti et al., 2021)	(Ruiz-sarmiento et al., 2020)	(Abdalla et al., 2022)	(Nguyen & Medjaher, 2019)	(Einabady et al., 2019)	(Ayvaz & Alpay, 2021)	(Cerrada et al., 2016)
		Industria	Automotriz	Energía	Metalurgia	Petrolera	Aeroespacial	Automotriz	Manufactura	Manufactura	
Monitoreo		Motor	Planta hidroeléctrica	Laminadora	Bombas	Turbinas	Línea de producción	Línea de producción	Engranajes		
Enfoque		Combinado	No supervisado	Supervisado	Supervisado	Supervisado	Supervisado	Supervisado	Supervisado		
Mediciones	Vibración		x				x		x		
	Temperatura	x	x	x	x		x	x			
	Presión	x	x	x	x			x			
	Velocidad	x						x			
	Fuerza			x				x			
	Electricidad				x		x	x			
Modelos	Árbol de decisión			x							
	Bosques aleatorios							x	x		
	SVM			x							
	Red neuronal		x	x		x	x				
	XGBoost				x			x			
	Otros	x		x							

Así como existen variables comunes para la mayoría de los casos identificados, hay otras específicas y con menor ocurrencia. La fuerza, presente en los estudios de Ruiz-sarmiento et al. (2020) y Ayvaz & Alpay (2021) es un ejemplo de variable poco común. La variable velocidad es otro ejemplo, al solo encontrarse en los estudios de Killeen et al. (2019) y Ayvaz & Alpay (2021). Por otro lado, variables como temperatura, presión y electricidad son más comunes, y se encuentran presentes en la mayoría de los estudios identificados.

5.3 Preprocesamiento de datos

Según Tornede et al. (2020), los datos capturados en el tiempo por sensores requieren ser preprocesados antes de probar modelos de ML y una práctica común es agruparlos en intervalos de tiempo.

Esto se puede apreciar en algunos estudios, como el de Liu et al. (2021), quienes analizaron patrones de consumo eléctrico de edificios. Para ello reconfiguraron los datos de medidores en intervalos de tiempo de entre 3 y 6

horas a lo largo del día, creando medidas como el promedio o diferencia entre máximos y mínimos.

Algo similar se aprecia en el estudio de Tornede et al., (2020), enfocado en la estimación de vida restante de los componentes de un motor de turbina. En este estudio los datos fueron agrupados en periodos de tiempo y se extrajeron características para describir dicho periodo con medidas como mínimo, máximo, número de picos, varianza, entre otros.

El estudio de Amihai et al. (2018), enfocado en predecir el estado de salud de bombas industriales a partir de datos de vibración y temperatura, tuvo un enfoque similar. Los datos capturados fueron resumidos en ventanas de tiempo específicas (1 día, 3 días, 7 días) antes de utilizar algoritmos de ML.

Por otro lado, un problema común en el caso de modelos de clasificación es el desbalanceo en las clases o etiquetas, que ocurre cuando existe una cantidad considerablemente mayor o menor de una etiqueta en específico (Betti et al., 2021). Para corregir este problema se puede recurrir a técnicas de submuestreo aleatorio (He & Ma, 2013) y re-muestro *Synthetic Minority Over-sampling Technique* (SMOTE por sus siglas en inglés), propuesta por Chawla et al. (2002).

El submuestreo reduce aleatoriamente la cantidad de observaciones de la clase mayoritaria, mientras que el re-muestro crea nuevos ejemplos de la clase minoritaria derivados de las observaciones originales. En ambos casos se trabaja con la proporción de la clase minoritaria que se desea luego del submuestreo o re-muestreo. Una vez los datos hayan sido preprocesados, el siguiente paso es seleccionar un modelo.

5.4 Casos de estudio de mantenimiento predictivo

La Tabla 4 indica un resumen de algunos de los estudios más significativos del mantenimiento predictivo basado en datos haciendo uso modelos de ML. Se incluye el tipo de industria en donde fueron realizados, qué componentes o equipos fueron monitoreados, qué enfoque de ML hubo, las variables que

fueron medidas y los modelos que mejor resultados les brindaron. A continuación, se incluye un resumen de algunos de estos estudios.

Ayvaz & Alpay (2021) ocuparon un modelo de bosques aleatorios, el cual arrojó valores R^2 por encima de 0.9 y error cuadrático medio mínimo en cuanto a la predicción de tiempo de vida útil restante antes de una falla. Utilizaron los datos en series de tiempo capturados por sensores de movimiento, velocidad, peso, temperatura y corriente eléctrica existentes en una línea de manufactura. Ante la poca existencia de eventos de falla, presentaron un desbalanceo de clases. Gracias a un análisis de correlación, el conocimiento de expertos y un análisis de componentes principales (PCA), redujeron la alta dimensionalidad que tenían y pasaron de 94 variables a 19. Esto debido a que algunas variables eran derivadas de un mismo sensor. Otros algoritmos utilizados fueron Gradient Boosting y SVM que no reportaron resultados igual de positivos.

La investigación de Cerrada et al. (2016) estuvo enfocada en el monitoreo de estado de los engranajes de una caja de cambios y uso de modelos de aprendizaje de máquinas para detección de fallas. Fue desarrollada en un entorno de laboratorio. Registraron las vibraciones bajo distintos escenarios de falla simulados y obtuvieron un total de 359 atributos, por lo que fue necesario descartar una parte de ellos antes de usar el modelo de bosques aleatorios. Los distintos grupos de atributos seleccionados, así como los parámetros de dichos modelos permitieron alcanzar valores de precisión y f1-score superiores a 0.9. Fue posible clasificar fallas basándose en las lecturas de vibración.

Por otro lado, existen también investigaciones cuyo enfoque fue el uso de técnicas de aprendizaje profundo, un tipo de modelo de ML más específico. Nguyen & Medjaher (2019) propusieron el uso de una red neuronal tipo Long Short-Term Memory (LSTM) para estimar la probabilidad de falla del sistema en una ventana de tiempo en el futuro. Se trata de un problema conocido como estimación de vida útil restante (RUL). Utilizaron el set de datos *Turbofan Engine Degradation Simulation* de la Nasa, el cual detalla los modos de operación de los motores, los datos recolectados por 21 sensores y cómo se degrada en el tiempo hasta llegar a un evento de falla. Su enfoque destaca la

importancia de la ventana de tiempo ocupada. Para algunos motores y ventanas de tiempo en específico lograron estimar probabilidades del 100%.

Por su parte, Einabady et al. (2019) propusieron una metodología basada en monitoreo de datos en tiempo real para la industria automotriz y determinaron el mejor momento para hacer estos mantenimientos. Utilizaron un modelo de redes neuronales artificiales (ANN) y el resultado arrojado fue utilizado como entrada para un problema de optimización. Este estudio buscaba minimizar el costo de mantenimiento asociado a reemplazar o no un componente en un periodo dado. Exitosamente estimaron la vida útil restante para 24 partes de vehículos con los datos recolectados por sensores de temperatura, vibración y consumo energético.

Las investigaciones sobre mantenimiento predictivo en sistemas de almacenamiento automático (AS/RS), objeto de estudio del presente proyecto, son casi nulas. Salah et al. (2015) desarrollaron un Análisis de Modos de Fallas y Efectos (FMEA) para un prototipo de AS/RS que estaba siendo desarrollado en la Universidad de Duisburg-Essen. Este equipo es totalmente distinto al modelo VCM, pero mantiene el mismo principio de manipulación, movimiento y almacenamiento de la carga. Dentro de su análisis encontraron que los componentes que suponen un riesgo para la maquinaria son los motores, la plataforma elevadora y los cables que sostienen la estructura.

5.5 Consideraciones finales

Finalmente, la literatura consultada también menciona que existen oportunidades de mejora para el campo del mantenimiento predictivo. Un ejemplo es la integración del flujo de información de una manera escalable como un reto considerable para las organizaciones (Dalzochio et al., 2020). Esto implica alinear la captura de datos, su procesamiento, la creación de modelos de inteligencia artificial, la integración con sistemas ERP y la oportuna comunicación de las alertas. Además, Nguyen & Medjaher (2019) destacan que existe una brecha entre los pronósticos basados en datos y las decisiones de mantenimiento que se deben tomar.

En el contexto de la industria 4.0, la disponibilidad de sistemas ciber-físicos que permiten la comunicación entre sensores y equipos, hace posible la generación y captura de grandes volúmenes de datos. Lee et al. (2015) y Javaid et al. (2021) destacan la importancia de transformar datos en conocimientos. El mantenimiento predictivo es un caso de estudio que ha sido abordado en la literatura a través del uso de datos recolectados por sensores y técnicas de analítica como el ML supervisado, tal y como se aprecia en la Tabla 4.

Las industrias en donde se ha llevado a cabo estos proyectos son diversas. Un aspecto que destaca en cuanto al preprocesamiento de datos es su agrupación en intervalos de tiempo. Indistintamente de los modelos utilizados, el objetivo fue predecir el momento exacto en que una falla ocurrirá para evitarla. Las redes neuronales y los de árboles de decisión, así como sus derivados (bosques aleatorios, XGBoost) son los modelos más comunes.

El presente proyecto no es algo nuevo en este campo. Se pretende explorar los datos disponibles capturados por los sensores de las máquinas modelo VCM, preprocesarlos y aplicar la metodología de ML supervisado automatizado, con el objetivo de encontrar un modelo capaz de identificar eventos de falla antes de que ocurran.

6 Objetivos

Determinar el mejor modelo de mantenimiento predictivo para los sistemas de almacenamiento VCM tomando como referencia la métrica de evaluación f1-score. Para lo cual, se deberá cumplir los siguientes objetivos específicos:

1. Construir una rutina de extracción y procesamiento de datos.
2. Identificar las variables disponibles y seleccionar las más importantes.
3. A través del ML supervisado automatizado, determinar los mejores modelos de clasificación de fallas y evaluarlos con los datos disponibles.
4. Validar la capacidad de predicción de fallas de los modelos encontrados con un conjunto de datos nuevo.

A cada objetivo se le puede asociar correspondientemente los siguientes criterios de éxito.

1. Descargar la totalidad de datos disponibles asociados al modelo VCM para al menos 10 máquinas.
2. Identificar al menos 5 variables independientes con las que se pueda trabajar.
3. Obtener por lo menos 3 modelos de ML.
4. Lograr clasificar por lo menos un evento de falla del nuevo conjunto de datos correctamente.

7 Metodología

Se toma como referencia el marco de ML supervisado propuesto por Kotsiantis (2007) que se aprecia en la Ilustración 4 y que se explica a continuación.

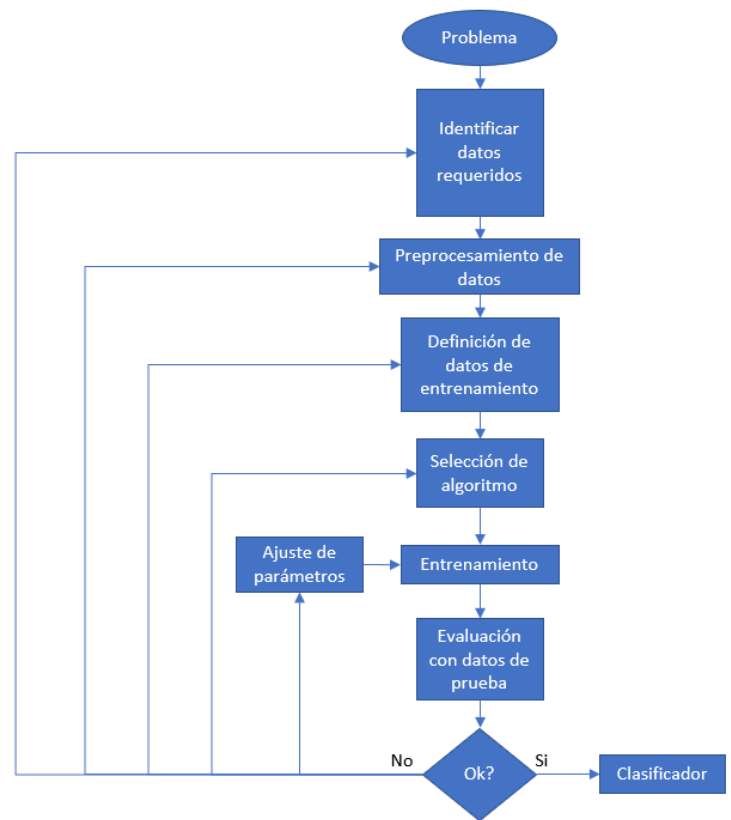


Ilustración 4 ML supervisado, adaptado de Kotsiantis (2007)

Se comienza con identificar un problema y recolectar todos los datos disponibles. Luego, se procesa el conjunto de datos inicial y valida aspectos como valores faltantes, atípicos y selección de variables. El conjunto de datos resultante será dividido en subconjuntos de entrenamiento y prueba.

Después, se selecciona un modelo, el cual será evaluado y sus parámetros ajustados de manera iterativa hasta alcanzar resultados satisfactorios. El modelo resultante será evaluado con los datos de prueba, para validar su comportamiento con datos nuevos. Otra forma de evaluar los modelos es con la técnica de la validación cruzada. Consiste en dividir el conjunto de datos inicial en subconjuntos mutuamente exclusivos y de igual tamaño, probar el modelo y obtener un promedio de la precisión alcanzada.

En esta misma línea, Tornede et al. (2020), establece un concepto más avanzado, el aprendizaje de máquinas supervisado automatizado. Este enfoque procede según la metodología antes descrita con la diferencia de que el proceso es automático. Las diferentes combinaciones posibles de preprocesamiento de datos, selección de variables, modelos y parámetros, son probadas automáticamente hasta encontrar una secuencia óptima para un determinado conjunto de datos.

A continuación, se describe cómo se aplicó la metodología del ML supervisado automatizado en el presente proyecto.

Los datos capturados por sensores de distintas máquinas modelo VCM fueron descargados desde la nube. Este tipo de datos usualmente son de alta dimensionalidad porque se cuenta con muchas variables y observaciones, lo que ocasiona que trabajar directamente con ellos sea costoso en términos de procesamiento y almacenamiento (Rani & Sikka, 2012).

El procesamiento consistió en agruparlos en periodos de tiempo de 6 horas para generar nuevas características y tenerlos en formato tabular (Tornede et al., 2020). En cuanto al tamaño de muestra, Ramezan et al., (2021), identificaron que a partir de 315 observaciones se alcanza alta precisión en algunos modelos, mientras que Figueroa et al., (2012) colocan esta cifra entre 80 y 560. Se consideró un mínimo de 560 observaciones.

Para seleccionar las variables más importantes, se calculó el factor de inflación de la varianza (*VIF*), el cual determina el grado de multicolinealidad entre variables e indica cuáles son redundantes (Gupta & Nahar, 2022). Luego, se ocuparon técnicas de submuestreo aleatorio (He & Ma, 2013) y remuestreo

SMOTE para balancear el conjunto de datos resultante (Chawla et al., 2002) pues se conoce de antemano que existe una limitada cantidad de eventos de falla.

En la etapa de selección de modelos, se hizo uso del ML supervisado automatizado con la ayuda del paquete de Python NaiveAutoML (Mohr & Wever, 2021). Los modelos que este paquete arrojó fueron evaluados con la técnica de validación cruzada. La métrica de evaluación fue el f1-score debido al desbalanceo existente en el problema. Finalmente, se utilizó un conjunto de datos nuevo, para evaluar la capacidad de los modelos previamente obtenidos para identificar fallas.

8 Resultados

8.1 Recolección de datos

Se identificaron 15 máquinas de modelo VCM pertenecientes a distintos clientes en Europa que están conectadas y han capturado datos durante el último año. Se extrajeron todos los datos disponibles desde el 1 de enero de 2021 hasta el 1 de marzo de 2022. Estos datos se encontraban en una base de datos PostgreSQL almacenada en Azure. Todo el proceso de descarga, procesamiento de datos y generación de modelos se realizó en Python 3 a través de Jupyter.

Las máquinas cuentan con sensores de movimiento, voltaje del controlador principal, equilibrio de la carga y contadores de ciclos y horas para la máquina, puerta y extractor. El intervalo de captura varía entre 1 segundo y 10 minutos y 30 minutos dependiendo de cada variable. Finalmente son enviadas a la base de datos. La organización captura estos valores con el propósito de crear tableros de control a los cuales el cliente pueda acceder.

También se cuenta con un registro de eventos de falla asociados a cada una de las máquinas. Similar a los sensores, cada falla se registra y almacena en la base de datos con fecha, hora y código de error asociado. Cada uno de estos códigos de error detalla el grado de severidad y el componente al cual está asociado. Se consideraron todos los errores catalogados como prioridad 1.

8.2 Tratamiento de los datos

8.2.1 Extracción

La Ilustración 5 muestra una parte del contenido de la base de datos. Se compone de varias tablas para cada máquina. El nombre de cada tabla consiste en el serial de la máquina, y el tipo de datos que almacena. Una tabla para cada tipo de datos recolectados tales como producción, operativos, estadísticas, desequilibrio de la máquina y las fallas. A través del paquete de Python Psycopg2, se creó una conexión entre Python y la base de datos PostgreSQL para descargar masivamente cada una de las tablas a través de una rutina de extracción y preprocesamiento.






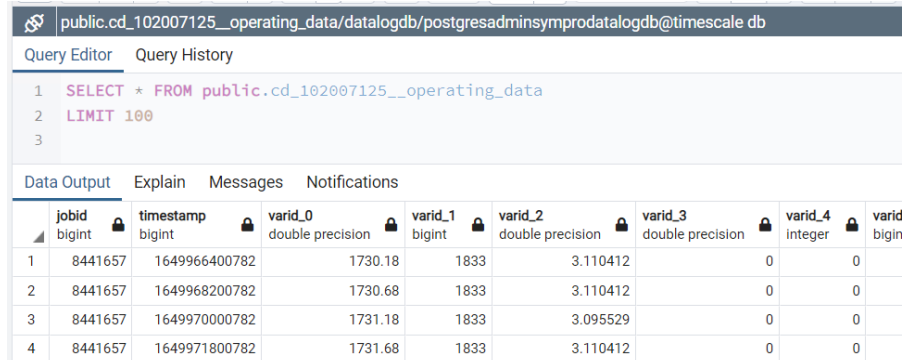
- >  cd_102006865__operating_data
- >  cd_102006865__stat_vcm
- >  cd_102006865__statusmessages
- >  cd_102006865__system_messages
- >  cd_102006865__unbalance

Ilustración 5 Lista de tablas en la base de datos



The screenshot shows a PostgreSQL query editor interface. At the top, the database name is 'public.cd_102007125_operating_data/datalogdb/postgresadminsprodatalogdb@timescale db'. Below that, there are tabs for 'Query Editor' and 'Query History'. The query editor contains the following SQL query:

```
1 SELECT * FROM public.cd_102007125_operating_data
2 LIMIT 100
3
```

Below the query editor, there are tabs for 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with the following columns and data:

	jobid bigint	timestamp bigint	varid_0 double precision	varid_1 bigint	varid_2 double precision	varid_3 double precision	varid_4 integer	varid bigint
1	8441657	1649966400782	1730.18	1833	3.110412		0	0
2	8441657	1649968200782	1730.68	1833	3.110412		0	0
3	8441657	1649970000782	1731.18	1833	3.095529		0	0
4	8441657	1649971800782	1731.68	1833	3.110412		0	0

Ilustración 6 Tabla de datos operativos de la máquina 102007125

El contenido de una tabla en particular se aprecia en la Ilustración 6. La columna “timestamp” indica el momento en que el registro fue capturado (tiempo en formato unix), mientras que el resto de las columnas indican diferentes variables. Estas columnas se encuentran codificadas como “varid_0”, “varid_1” y así sucesivamente. El significado de esta codificación estaba almacenado en otra tabla. Así, por ejemplo, para esta tabla de datos operativos, “varid_0” representa la cantidad de horas operativas, “varid_1”

representa los ciclos operativos y “varid_2” representa el voltaje de la batería del controlador.

En la Ilustración 7 se muestra el comportamiento las variables desequilibrio y ciclos operativos para las máquinas con serial 102008388 y 102006615. En el eje x se encuentra el tiempo, y el eje y el valor registrado. La variable desequilibrio indica hacia dónde está inclinada la carga de la máquina, se mide en % y oscila entre 100% y -100%. La línea roja indica el momento en el que se presentó una falla. La variable de ciclos operativos cuenta cada vez que la máquina ha realizado una operación de almacenamiento.

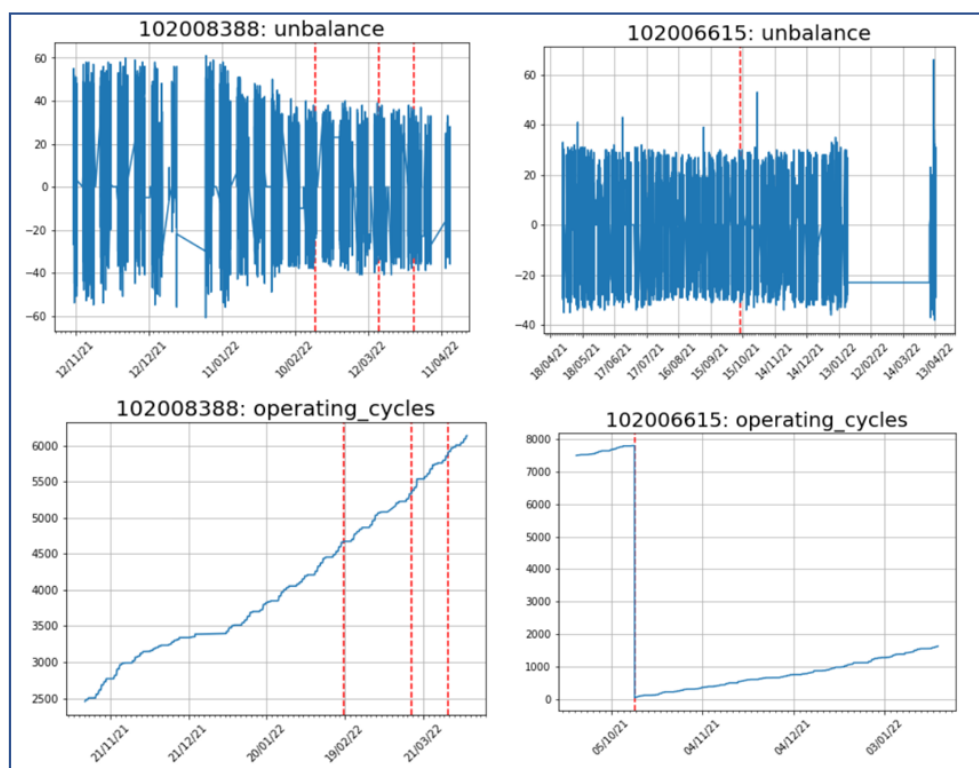


Ilustración 7 Exploración de algunas variables

8.2.2 Agrupación

Las tablas de cada máquina fueron combinadas entre sí. De esta manera, todo lo relacionado a una máquina se encuentra en una sola gran tabla en lugar de tablas separadas. Se convirtió el tiempo unix a formato tradicional (dd/mm/yyyy hh:mm:ss).

Luego, la tabla fue agrupada en intervalos de 6 horas. Por ejemplo, el intervalo de tiempo desde las 12:00AM hasta las 5:59AM será catalogado como 6:00AM,

6:00AM hasta las 11:59AM será catalogado como 12:00PM. En cada intervalo se resumió el comportamiento de las variables a través de un tipo de agrupación, así como la presencia o ausencia de una falla en el periodo siguiente. En la Tabla 5 se resumen las variables identificadas y el tipo de característica generada. Inicialmente se identificaron 37 variables que al ser agrupadas pasan a un total de 42.

Tabla 5 Lista de las variables identificadas y tipo de agrupación

Variable	Descripción	Agrupación				
		Mínimo	Promedio	Máximo	Desv. Est.	Variación
battery_voltage	Voltaje de la batería	x	x	x		
unbalance	Desequilibrio de la máquina	x	x	x	x	
busyTime	Tiempo ocupado (en h)					x
idleTime	Tiempo desocupado (en h)					x
oh_door_1	Horas operativas de las puertas 1 - 6					x
oc_door_1	Ciclos operativos de las puertas 1 - 6					x
oc_tray_extractor_1	Ciclos operativos de las bandejas extractoras 1 - 6					x
oh_tray_extractor_1	Horas operativas de las bandejas extractoras 1 - 6					x
oc_drive	Ciclos operativos del motor					x
oh_drive	Horas operativas del motor					x
oc_extractor	Ciclos operativos del extractor					x
oh_extractor	Horas operativas del extractor					x
operating_cycles	Ciclos operativos de la máquina					x
operating_hours	Horas operativas de la máquina					x
picking_cycles_off	Ciclos operativos desde el último apagado					x
picking_cycles_on	Ciclos operativos desde el último encendido					x
remaining_cycles	Ciclos restantes antes de un mantenimiento					x

8.2.3 Preprocesamiento

Cada variable fue explorada para identificar ceros, nulos, o valores atípicos y removerlos en caso de que sea justificable. Producto de fallas en la lectura de los sensores, se identificaron valores negativos o valores extremadamente grandes en algunas variables. En estos casos, toda la observación fue eliminada.

Las variables vacías, llenas de ceros o con registros insuficientes en comparación al resto, fueron eliminadas. En particular, todas las variables asociadas a las puertas y bandejas extractoras 2, 3, 4, 5 y 6 no registran valor alguno. Esto sucede porque las máquinas analizadas solo tienen una puerta. Otras variables, como los ciclos restantes, el tiempo ocupado y tiempo ocioso, se asocian a sensores que no están configurados correctamente y no han capturado suficientes datos. Finalmente, periodos de tiempo en donde no hubo actividad (fines de semana, horarios nocturnos, de madrugada o días festivos), también fueron eliminados.

En la Ilustración 8 que se muestra a continuación, se indican dos observaciones asociadas a periodos de tiempo de 6 horas en donde se evidencia el comportamiento de las variables y si hubo una falla (derecha) o no (izquierda) en el periodo siguiente.

battery_voltage_min	3.095529	battery_voltage_min	3.080647
battery_voltage_mean	3.109172	battery_voltage_mean	3.09801
battery_voltage_max	3.110412	battery_voltage_max	3.110412
unbalance_min	-37.0	unbalance_min	-75.0
unbalance_mean	-12.80303	unbalance_mean	12.121359
unbalance_max	14.0	unbalance_max	73.0
unbalance_std	10.425147	unbalance_std	30.991703
oh_door_1_hourly	0.0	oh_door_1_hourly	0.02
oc_door_1_hourly	0.0	oc_door_1_hourly	29.0
oh_drive_hourly	0.03	oh_drive_hourly	0.15
operating_cycles_hourly	4.0	operating_cycles_hourly	29.0
picking_cycles_on_hourly	4.0	picking_cycles_on_hourly	33.0
breakdown	False	breakdown	True

Ilustración 8 Ejemplos de observaciones resultantes

Luego de procesar los datos, el conjunto cuenta con 13 variables, 1784 observaciones sin falla y 10 con falla. Existe un desbalanceo en las etiquetas de clase. La Ilustración 9 indica las variables identificadas. Las primeras 11 serán consideradas independientes, y la variable 12 como dependiente.


```

0  battery_voltage_min
1  battery_voltage_mean
2  battery_voltage_max
3  unbalance_min
4  unbalance_mean
5  unbalance_max
6  unbalance_std
7  oh_door_1_hourly
8  oc_door_1_hourly
9  oh_drive_hourly
10 operating_cycles_hourly
11 picking_cycles_on_hourly
12 breakdown

```

Ilustración 9 Resumen de la base de datos resultante

En la Ilustración 10 se aprecian histogramas con la distribución de las variables identificadas. No se aprecian valores atípicos y se puede notar que cada variable sigue una distribución en particular.

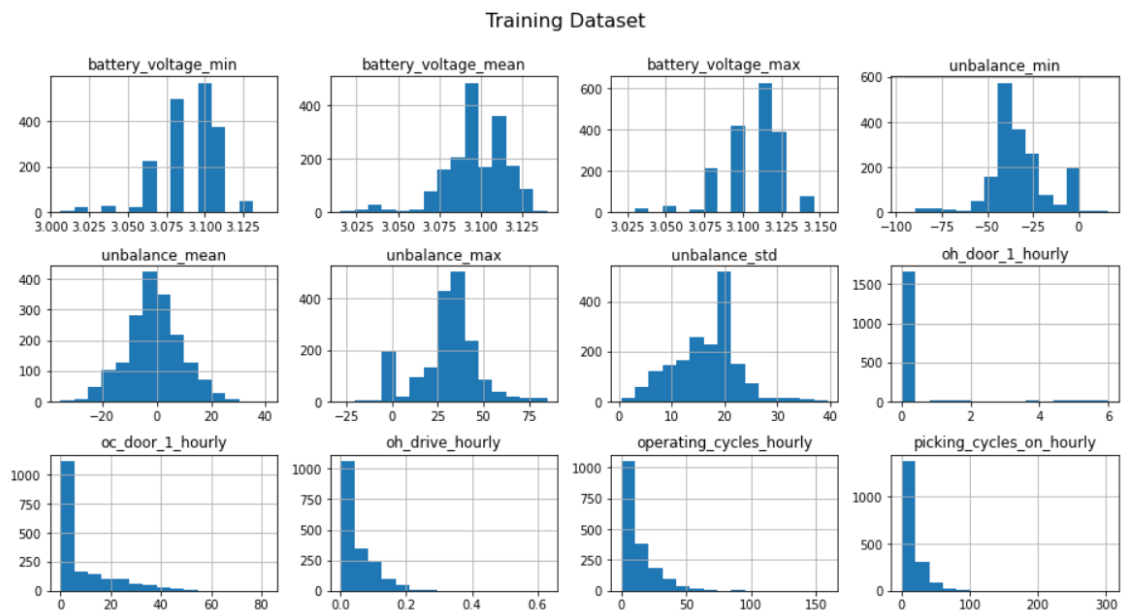


Ilustración 10 Distribución de datos de entrenamiento

Las actividades de extracción, agrupación y preprocesamiento descritas fueron automatizadas en un código de Python que se puede correr en cualquier momento. Al contar con esta rutina de extracción y procesamiento de datos se cumple el primer objetivo específico.

8.2.4 Balanceo del conjunto de datos

Se combinaron las técnicas de submuestreo aleatorio de la clase mayoritaria (sin falla) y remuestreo SMOTE de la clase minoritaria (falla), combinando

distintas proporciones arbitrarias para cada técnica. De esta manera, para cada conjunto de datos generado se obtendrá un modelo de ML.

Para el submuestreo aleatorio se utilizaron distintas proporciones (0.0098, 0.0137, 0.0175). Esta proporción reduce la cantidad de observaciones de la clase mayoritaria. Por ejemplo, si se tiene 1784 observaciones asociadas a la clase mayoritaria y 10 observaciones a la minoritaria, una proporción de 0.0098 reduce a 1020 las observaciones de la clase mayoritaria. Este valor se obtiene al dividir las 10 observaciones minoritarias entre la proporción de 0.0098.

En el caso del remuestreo, también se usaron proporciones (0.08, 0.12, 0.16). Este valor aumenta la cantidad de observaciones de la clase minoritaria. En este caso, con el resultado del ejemplo anterior, una proporción de 0.08 aumentaría las 10 observaciones minoritarias a 81. Este número se obtiene de multiplicar la proporción por la cantidad de observaciones en la clase mayoritaria.

Se crearon conjuntos de datos para cada combinación posibles entre las proporciones. Es decir, un submuestreo con una proporción de 0.0098 combinado con un remuestreo con proporción 0.08, otro submuestreo con la misma proporción de 0.0098 combinado con un remuestreo con proporción 0.12. En la Tabla 6, se muestran las características de los 9 conjuntos de datos resultantes y se evidencia cómo se redujo la clase mayoritaria a la vez que se aumentó la clase minoritaria.

Tabla 6 Conjuntos de datos resultantes y distribución de clases

Conjunto de datos	No fallas	Fallas	Observaciones totales
Dataset #0	1020	81	1101
Dataset #1	1020	122	1142
Dataset #2	1020	163	1183
Dataset #3	729	58	787
Dataset #4	729	87	816
Dataset #5	729	116	845
Dataset #6	571	45	616
Dataset #7	571	68	639
Dataset #8	571	91	662

8.3 Selección de variables

Para cada conjunto de datos, se calculó el coeficiente de inflación de la varianza (VIF) y de forma iterativa se descartaron variables. En la Tabla 7 se muestran las variables resultantes de cada conjunto de datos.

Tabla 7 Lista de variables seleccionadas para cada conjunto de datos

	dataset_0	dataset_1	dataset_2	dataset_3	dataset_4	dataset_5	dataset_6	dataset_7	dataset_8
battery_voltage_min	x	x	x	x	x	x		x	x
battery_voltage_mean									
battery_voltage_max							x		
unbalance_min				x					
unbalance_mean	x	x	x	x	x	x	x	x	x
unbalance_max	x	x	x		x	x	x	x	x
unbalance_std									
oh_door_1_hourly	x	x	x	x	x	x	x	x	x
oc_door_1_hourly	x	x	x	x	x	x	x	x	x
oh_drive_hourly		x	x	x	x	x		x	
operating_cycles_hourly									
picking_cycles_hourly	x	x	x	x	x	x	x	x	x

De esta manera, se ha cumplido el segundo objetivo específico. Se identificaron las variables no relacionadas entre sí según el criterio VIF. Estas cambian según el conjunto de datos y las más comunes son el voltaje mínimo, el desequilibrio promedio, las horas y ciclos operativos de la máquina.

8.4 Aprendizaje de máquinas supervisado y automatizado

Los conjuntos de datos obtenidos pretenden resolver un problema de clasificación. Fueron divididos en variables independientes (X) (según cada conjunto de datos, ver Tabla 7) y dependientes (y) (la presencia de una falla o no en el periodo siguiente). También se consideró el conjunto de datos inicial.

Con el paquete AutoNaiveML, se realiza el ajuste de cada conjunto de datos. Se retorna la secuencia recomendada y el puntaje f1 obtenido. Esta secuencia consiste en un método de procesamiento de datos, un criterio de selección de variables, y los modelos evaluados con sus parámetros. Dado el desbalanceo existente en los conjuntos de datos, se consideró la métrica de evaluación “f1”.

En la Tabla 8 se indican las secuencias obtenidas para cada conjunto de datos. Los modelos obtenidos fueron Extra Trees, Random Forest, Histogram

Gradient Boosting Classification Tree y Gaussian Naive Bayes. En el Anexo 1 se detallan los parámetros específicos de cada secuencia.

Tabla 8 Secuencias recomendadas por NaiveAutoML

Conjunto de datos	Pre-procesamiento	Selección de variables	Clasificador	F1
dataset_0	QuantileTransformer	NA	ExtraTrees	0.8
dataset_1	QuantileTransformer	NA	ExtraTrees	0.9
dataset_2	QuantileTransformer	NA	RandomForest	0.94
dataset_3	PowerTransformer	NA	ExtraTrees	0.83
dataset_4	PowerTransformer	NA	ExtraTrees	0.88
dataset_5	NA	NA	HistGradientBoosting	0.94
dataset_6	NA	NA	HistGradientBoosting	0.86
dataset_7	NA	NA	HistGradientBoosting	0.88
dataset_8	NA	NA	HistGradientBoosting	0.89
dataset_original	StandardScaler	SelectPercentile	GaussianNB	0.17

Finalmente, las secuencias fueron evaluadas para cada conjunto de datos con el objetivo de verificar la veracidad de los resultados arrojados por AutoNaiveML. A través de una validación cruzada con 5 particiones se calculó el puntaje f1. Estos valores coinciden con lo arrojado por el paquete y oscilan entre el 78% y 92%. El conjunto de datos original, sin ningún preprocesamiento, muestra un desempeño inferior al resto. En la Ilustración 11 se muestran los resultados obtenidos. Es posible apreciar que los modelos más recomendados por NaiveAutoML son *histogram based gradient boosting* y *Extra trees*. En este punto se ha cumplido el tercer objetivo específico.



Ilustración 11 Puntajes f1 de cada modelo durante entrenamiento

8.5 Recolección de nuevos datos y validación final

Con la misma rutina asociada al objetivo específico 1, se extrajeron datos generados desde el 1 de marzo de 2022 hasta el 15 de abril de 2022. Se identificaron un total de 301 observaciones, de las cuales 297 se asocian a periodos sin falla y 4 en donde sí hubo falla. En este punto se está validando la capacidad de los modelos para identificar fallas al usar datos nunca vistos.

Se consideraron las secuencias obtenidas en la etapa de entrenamiento y que se muestran en la Tabla 8. Cada una fue evaluada con el conjunto de datos nuevo, de manera que se cumple el cuarto objetivo específico. En la Ilustración 12 se indican los puntajes f1 obtenidos.

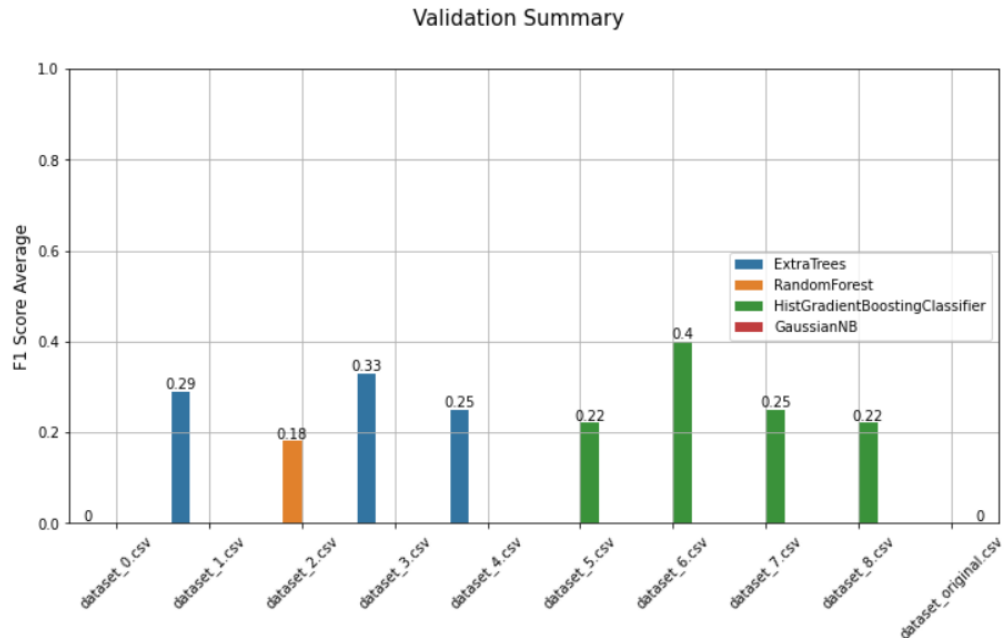


Ilustración 12 Puntajes f1 de cada modelo durante validación final

Los puntajes f1 obtenidos en la etapa de validación final son inferiores a los obtenidos durante el entrenamiento. En la Ilustración 12, se evidencia que ahora bordean el 30%, cuando en el entrenamiento alcanzaba 80%, según la Ilustración 11. La secuencia de datos asociada al conjunto de datos 6 es la que presenta los mejores resultados. El modelo asociado a este conjunto de datos es un *Histogram-based gradient boosting classifier*. En la gráfica de confusión mostrada en la Ilustración 13 se evidencia que, de los 4 nuevos eventos de falla identificados, solo 1 fue clasificado correctamente. Los periodos de tiempo donde no hubo ninguna falla fueron clasificados correctamente.

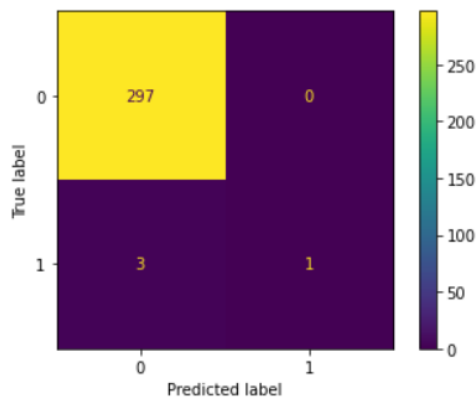


Ilustración 13 Matriz de confusión del conjunto de datos 6

9 Discusión

La organización objeto de estudio del presente proyecto cuenta con la infraestructura para capturar y almacenar datos recolectados por sensores, así como las fallas que sus máquinas presenten. Por tanto, el objetivo del presente proyecto fue encontrar un modelo de ML supervisado que permita clasificar los periodos de tiempo en función de si una falla se va a presentar o no en las 6 horas siguientes.

A través del uso del paquete de Python Naive AutoML y los datos de entrenamiento los mejores modelos encontrados fueron Extra Trees, Random Forest, Histogram Gradient Boosting Classification Tree. Estos modelos alcanzaron puntajes f1 alrededor de 0.8. Sin embargo, al utilizar estos modelos obtenidos bajo un conjunto de datos nuevo, los resultados obtenidos cambiaron y los puntajes f1 bajaron a un promedio de 0.3.

De acuerdo con la metodología propuesta por Kotsiantis (2007), cuando los resultados alcanzados no son satisfactorios, es necesario retroceder en el proceso e identificar qué aspectos se podría mejorar. En el presente proyecto existen oportunidades de mejora que son importantes mencionar:

- El conjunto de datos disponible tiene una limitada cantidad de fallas. Esto se puede atribuir a la naturaleza del problema de estudio. Cerrada et al. (2016) afirman que contar con muchas muestras para entrenar modelos de clasificación no es algo usual en entornos industriales reales. La ocurrencia de una falla en máquinas relativamente nuevas, como las del presente proyecto, es un caso fortuito. Por este motivo, fueron considerados todos los errores de prioridad 1. Se esperaría que con el pasar de los años, la cantidad de máquinas conectadas aumente, así como los datos y las fallas disponibles.
- De las 42 variables identificadas al inicio, esta cantidad se redujo a 13. Esto ocurrió porque existen sensores que no están debidamente configurados. Más variables podrían suponer una mejor imagen de qué sucedió antes de una falla. De tal manera que sería importante habilitar dichos sensores.

- Además, las variables identificadas en este proyecto difieren de aquellas de la literatura. En la Tabla 4 se evidencia que las más comunes están asociadas a vibración, temperatura y presión. En este caso no se cuenta con ninguna de ese tipo. Los resultados obtenidos podrían indicar que las variables disponibles no son las adecuadas para un enfoque de mantenimiento predictivo. Por lo tanto, debería considerarse incorporar otro tipo de mediciones para este propósito.
- Si bien los datos de las 15 máquinas contempladas pertenecen al mismo modelo (VCM), cada una de ellas tiene un contexto distinto. Las máquinas pertenecen a distintos clientes, en países distintos y manejan líneas de productos distintos. El nivel de uso que cada una tiene también cambia. Es decir, se trata de condiciones operativas distintas que podrían estar introduciendo ruido.

En cuanto a este último punto, un análisis gráfico reveló dos aspectos importantes. Primero, en la Ilustración 14 se aprecia el voltaje promedio de cada máquina y la variación entre cada una es evidente. En la Ilustración 15 se aprecia la variable desequilibrio promedio de tres máquinas. Dos de ellas, las máquinas 102006548 y 102006865, dejaron de registrar valores a partir de noviembre 2021. Esto afecta la capacidad de clasificación de los modelos puesto que las condiciones iniciales han cambiado.

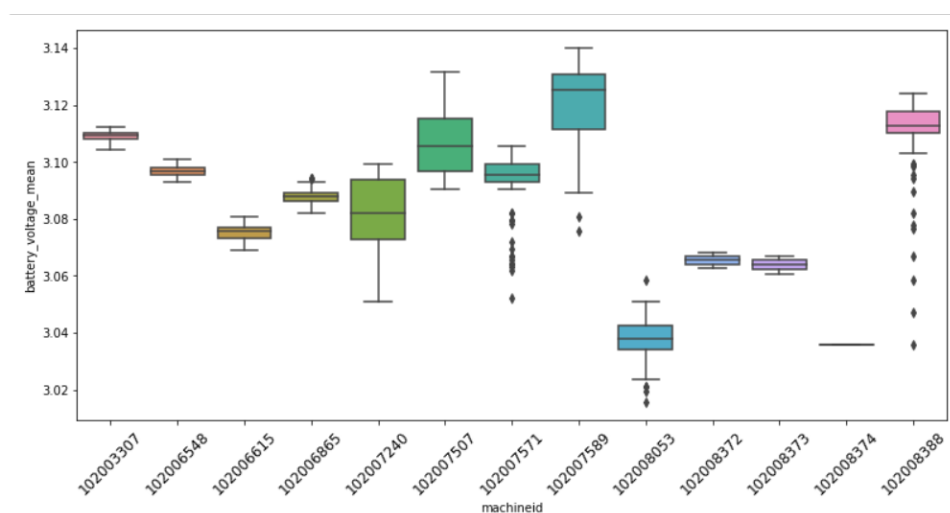


Ilustración 14 Voltaje de batería promedio, varias máquinas

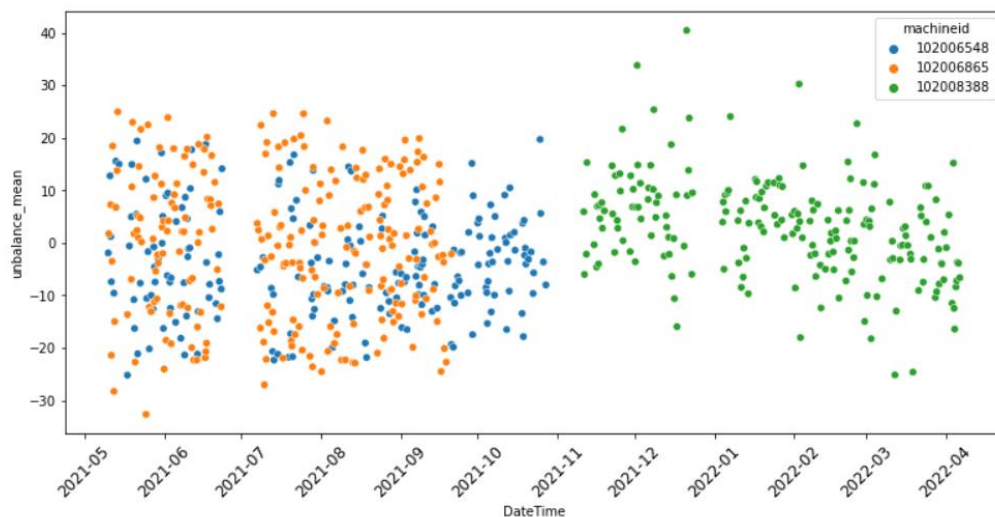


Ilustración 15 Desequilibrio promedio a lo largo del tiempo, varias máquinas

Sin embargo, de haber considerado únicamente una o dos máquinas en específico, el conjunto de datos resultante hubiese sido más pequeño aún. En particular, la cantidad de periodos en los cuales hubo una falla hubiese sido menor. Fue necesario considerar todos los datos existentes asociados a máquinas VCM para lograr obtener un conjunto de datos lo suficientemente grande. De manera que el dilema era centrarse en un análisis específico con pocas máquinas y un conjunto de datos reducido, o un análisis más general con todos los datos disponibles.

Como aclaración final, el motivo principal por el que la organización captura los datos utilizados en el presente proyecto es para crear tableros de control e indicadores. La idea es estos puedan venderse como un servicio adicional a los clientes. La organización quería evaluar si los datos actualmente disponibles pudieran tener otros usos. El mantenimiento predictivo es uno de esos propósitos la organización esperaba alcanzar.

10 Conclusiones y Trabajo Futuro

En el contexto de la I4.0, el mantenimiento predictivo es un tema que ha ganado relevancia durante los últimos años. Esto debido a la gran cantidad de sensores y sistemas ciber-físicos que permiten la captura, envío y almacenamiento de datos. La organización objeto de estudio cuenta esta infraestructura para sus máquinas modelo VCM. Aunque el propósito principal

de estos datos es crear tableros de control con métricas e indicadores, que puedan venderse a los clientes como un servicio adicional, también se buscaba generar valor y dar una posible solución al problema de fallas inesperadas en sus equipos a través de un mantenimiento predictivo.

Un enfoque preferido para resolver el problema del mantenimiento predictivo es aquel basado en datos. Estos datos generalmente provienen de sensores y se hace uso de técnicas de ML supervisado. Tal es el caso del presente proyecto, en donde los equipos cuentan con distintos tipos de sensores que capturan mediciones en distintos intervalos de tiempo. Algunas de las variables identificadas fueron desequilibrio de la carga que el equipo almacena, voltaje del tablero principal y la medición de horas y ciclos operativos.

El primer paso fue la descarga masiva de datos desde la nube, los cuales fueron capturados desde enero de 2021 hasta el primero de marzo de 2022. Las actividades de procesamiento más importantes fueron la agrupación en intervalos de tiempo de 6 horas y el balanceo de clases debido a la limitada cantidad de fallas registradas. A través de las técnicas de sub-muestreo y re-muestreo SMOTE, se generaron 10 conjuntos de datos distintos.

El siguiente paso fue buscar modelos con la ayuda del paquete de Python NaiveAutoML. El modelo que se buscaba permitiría alertar si en las siguientes 6 horas se va a registrar una falla o no. Este paquete, evaluó automáticamente varios modelos, así como sus parámetros. Durante esta etapa de entrenamiento, se alcanzaron puntajes f1-score de alrededor del 80% para los conjuntos de datos generados.

Luego, para la validación final, se extrajo un conjunto de datos nuevo generado a partir del 1 de marzo 2022 hasta el 15 de abril 2022. En esta etapa se puso a prueba la capacidad de predicción de los modelos previamente identificados para anticipar fallas. Se alcanzaron puntajes f1-score cercanos al 30%. Los modelos identificados no fueron capaces de clasificar correctamente las nuevas fallas registradas.

El impacto esperado con el presente proyecto era obtener un modelo de alta capacidad predictiva que permita anticiparse a la ocurrencia de fallas antes de

que ocurran. Sin embargo, si en este momento los modelos obtenidos fueran puestos a ejecución en tiempo real, no serían capaces de identificar la mayoría de las fallas que pudieran presentarse.. El mejor modelo de mantenimiento predictivo es un *Histogram-based gradient boosting classifier* identificado a partir del conjunto de datos “dataset_6”, al alcanzar un puntaje f1-score de 40% en la etapa de validación.

Se identificaron posibles causas para la baja capacidad de predicción de fallas en la etapa de validación. Primero, se cuenta con un conjunto de datos desbalanceado debido a la naturaleza del problema. Luego, hay una limitada cantidad de variables independientes. Después, al considerar máquinas con condiciones operativas distintas, se podría estar introduciendo ruido. Finalmente, al contrastar con la literatura, se identificó que las variables con las que se cuenta podrían no ser las mejores para un enfoque de mantenimiento predictivo. En los casos de estudio identificados se mencionan otro tipo de variables, asociadas a temperatura, humedad o presión, motivo por el cual la organización debería considerarlas. Internamente se debería evaluar la factibilidad de incorporar sensores para este propósito en las máquinas. Esto puede darse a través de proyectos piloto con máquinas puntuales antes de replicar a gran escala. El impacto obtenido con el presente proyecto puede resumirse de la siguiente forma. Se ha determinado que la organización se encuentra en una etapa muy temprana para alcanzar un enfoque de mantenimiento predictivo. El motivo principal es la limitada cantidad de datos existentes y las condiciones anteriormente descritas. Además, se ha realizado una primera exploración de los datos disponibles y se ha evaluado un primer acercamiento para resolver el problema en cuestión.

Aunque los resultados de este primer enfoque de mantenimiento predictivo no fueron positivos, el proyecto aún tiene mucha importancia para la organización. El ejercicio realizado se podría repetir dentro de algunos meses, en donde más datos hayan sido capturados. Cuando se logren suficientes datos, el análisis podría ser más específico y centrarse en una sola máquina. Además, se podría replicar este ejercicio para otros modelos de máquinas como el VBM y VLM. También se podría cambiar el enfoque del proyecto a uno de ML no

supervisado, definir comportamientos operativos normales e identificar anomalías.

Finalmente, otro aspecto que debería considerarse y que no fue contemplado en el alcance de este proyecto, es la integración de los modelos al flujo de la organización. Al ser códigos de Python su ejecución puede programarse para descargar datos, correr los modelos y tomar una acción en el caso de identificar una falla. Algunas preguntas por resolver serían: ¿dónde debe estar almacenado este código? ¿con qué frecuencia debería ejecutarse? Y la más importante de todas, ¿qué hacer si se detecta una potencial falla?

11 Anexos

Anexo 1. Secuencias recomendadas y parámetros sugeridos por NaiveAutoML para cada conjunto de datos

```

dataset_0.csv ("Pipeline(steps=[('data-pre-processor', QuantileTransformer()), ('classifier', ExtraTreesClassifier())]", 0.8657)
dataset_1.csv ("Pipeline(steps=[('data-pre-processor', QuantileTransformer()), ('classifier', ExtraTreesClassifier())]", 0.9048)
dataset_2.csv ("Pipeline(steps=[('data-pre-processor', QuantileTransformer()), ('classifier', RandomForestClassifier())]", 0.938)
dataset_3.csv ("Pipeline(steps=[('data-pre-processor', PowerTransformer()), ('classifier', ExtraTreesClassifier())]", 0.8263)
dataset_4.csv ("Pipeline(steps=[('data-pre-processor', PowerTransformer()), ('classifier', ExtraTreesClassifier())]", 0.8802)
dataset_5.csv ("Pipeline(steps=[('classifier', HistGradientBoostingClassifier(early_stopping=True, l2_regularization=3.0400794152507e-07, learning_rate=0.12376923152168709, max_iter=512, max_leaf_nodes=500, min_samples_leaf=9, n_iter_no_change=8, validation_fraction=None, warm_start=True))]", 0.9426)
dataset_6.csv ("Pipeline(steps=[('classifier', HistGradientBoostingClassifier(early_stopping=False, l2_regularization=0.004190392461688923, learning_rate=0.15942653799649206, max_iter=512, max_leaf_nodes=4, min_samples_leaf=1, n_iter_no_change=0, validation_fraction=None, warm_start=True))]", 0.8628)
dataset_7.csv ("Pipeline(steps=[('classifier', HistGradientBoostingClassifier(early_stopping=True, l2_regularization=1.3019522827837508e-05, learning_rate=0.2495513404567697, max_iter=512, max_leaf_nodes=42, min_samples_leaf=59, n_iter_no_change=8, validation_fraction=None, warm_start=True))]", 0.8833)
dataset_8.csv ("Pipeline(steps=[('classifier', HistGradientBoostingClassifier(early_stopping=False, l2_regularization=6.228545247627197e-06, learning_rate=0.11593471702316656, max_iter=512, max_leaf_nodes=16, min_samples_leaf=13, n_iter_no_change=0, validation_fraction=None, warm_start=True))]", 0.8962)
dataset_original.csv ("Pipeline(steps=[('data-pre-processor', StandardScaler()), ('feature-pre-processor', SelectPercentile()), ('classifier', GaussianNB())]", 0.1733)

```

12 Bibliografía

Amihai, I., Gitzel, R., Kotriwala, A. M., Pareschi, D., Subbiah, S., & Sosale, G. (2018). An industrial case study using vibration data and machine learning to predict asset health. *Proceeding - 2018 20th IEEE International*

Conference on Business Informatics, CBI 2018, 1, 178–185.
<https://doi.org/10.1109/CBI.2018.00028>

- Andersen, D. L., Ashbrook, C. S. A., & Karlborg, N. B. (2020). Significance of big data analytics and the internet of things (IoT) aspects in industrial development, governance and sustainability. *International Journal of Intelligent Networks*, 1, 107–111. <https://doi.org/10.1016/j.ijin.2020.12.003>
- Ashima, R., Haleem, A., Bahl, S., Javaid, M., Mahla, S. K., & Singh, S. (2021). Automation and manufacturing of smart materials in additive manufacturing technologies using Internet of Things towards the adoption of industry 4.0. *Materials Today: Proceedings*, 45, 5081–5088. <https://doi.org/10.1016/j.matpr.2021.01.583>
- Awan, U., Shamim, S., Khan, Z., Zia, N. U., Shariq, S. M., & Khan, M. N. (2021). Big data analytics capability and decision-making: The role of data-driven insight on circular economy performance. *Technological Forecasting and Social Change*, 168. <https://doi.org/10.1016/j.techfore.2021.120766>
- 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(September 2020), 114598. <https://doi.org/10.1016/j.eswa.2021.114598>
- Betti, A., Crisostomi, E., Paolinelli, G., Piazzzi, A., Ruffini, F., & Tucci, M. (2021). Condition monitoring and predictive maintenance methodologies for hydropower plants equipment. *Renewable Energy*, 171, 246–253. <https://doi.org/10.1016/j.renene.2021.02.102>
- Bowler, A. L., Pound, M. P., & Watson, N. J. (2022). A review of ultrasonic sensing and machine learning methods to monitor industrial processes. In *Ultrasonics* (Vol. 124). Elsevier B.V. <https://doi.org/10.1016/j.ultras.2022.106776>
- Cerrada, M., Zurita, G., Cabrera, D., Sánchez, R. V., Artés, M., & Li, C. (2016). Fault diagnosis in spur gears based on genetic algorithm and random

- forest. *Mechanical Systems and Signal Processing*, 70–71, 87–103.
<https://doi.org/10.1016/j.ymssp.2015.08.030>
- Chawla, N. v, Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. In *Journal of Artificial Intelligence Research* (Vol. 16).
- Ciancio, V., Homri, L., Dantan, J. Y., & Siadat, A. (2020). Towards prediction of machine failures: Overview and first attempt on specific automotive industry application. *IFAC-PapersOnLine*, 53(3), 289–294.
<https://doi.org/10.1016/j.ifacol.2020.11.047>
- Dalzochio, J., Kunst, R., Pignaton, E., Binotto, A., Sanyal, S., Favilla, J., & Barbosa, J. (2020). Machine learning and reasoning for predictive maintenance in Industry 4.0: Current status and challenges. *Computers in Industry*, 123, 103298. <https://doi.org/10.1016/j.compind.2020.103298>
- Einabady, B., Baboli, A., & Ebrahimi, M. (2019). Dynamic Predictive Maintenance in industry based on real time information: Case study automotive industries. *IFAC PapersOnLine*, 52(13), 1069–1074.
<https://doi.org/10.1016/j.ifacol.2019.11.337>
- el Morr, C., & Ali-Hassan, H. (2019). *Descriptive, Predictive, and Prescriptive Analytics* (pp. 31–55). https://doi.org/10.1007/978-3-030-04506-7_3
- Erickson, B. J., & Kitamura, F. (2021). Magician's corner: 9. performance metrics for machine learning models. In *Radiology: Artificial Intelligence* (Vol. 3, Issue 3). Radiological Society of North America Inc.
<https://doi.org/10.1148/ryai.2021200126>
- Erkoyuncu, J. A., Khan, S., Eiroa, A. L., Butler, N., Rushton, K., & Brocklebank, S. (2017). Perspectives on trading cost and availability for corrective maintenance at the equipment type level. *Reliability Engineering and System Safety*, 168, 53–69. <https://doi.org/10.1016/j.ress.2017.05.041>
- Figuroa, R. L., Zeng-Treitler, Q., Kandula, S., & Ngo, L. H. (2012). Predicting sample size required for classification performance. *BMC Medical*

- Informatics and Decision Making*, 12(1). <https://doi.org/10.1186/1472-6947-12-8>
- Fradkov, A. L. (2020). Early history of machine learning. *IFAC-PapersOnLine*, 53(2), 1385–1390. <https://doi.org/10.1016/j.ifacol.2020.12.1888>
- Gandomi, A., & Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*, 35(2), 137–144. <https://doi.org/10.1016/j.ijinfomgt.2014.10.007>
- Gouveia, B., & Costa, O. (2022). Industry 4.0: Predicting lead conversion opportunities with machine learning in small and medium sized enterprises. *Procedia Computer Science*, 204, 54–64. <https://doi.org/10.1016/j.procs.2022.08.007>
- Gupta, A., & Nahar, P. (2022). Classification and yield prediction in smart agriculture system using IoT. *Journal of Ambient Intelligence and Humanized Computing*. <https://doi.org/10.1007/s12652-021-03685-w>
- He, H., & Ma, Y. (2013). *Imbalanced Learning: Foundations, Algorithms, and Applications* (1st ed.). Wiley-IEEE Press.
- IBM. (2020). *Machine Learning*. <https://www.ibm.com/cloud/learn/machine-learning>
- Javaid, M., Abid Haleem, Pratap Singh, R., Rab, S., & Suman, R. (2021). Upgrading the manufacturing sector via applications of Industrial Internet of Things (IIoT). *Sensors International*, 2, 100129. <https://doi.org/10.1016/j.sintl.2021.100129>
- Kardex. (2020). *Remote Report 2020*.
- Kardex. (2021a). *Kardex Corporate Profile*. Kardex. <https://www.kardex.com/en/company/corporate-profile>
- Kardex. (2021b). *Sistemas automatizados de almacenamiento y recuperación*. Kardex. <https://www.kardex.com/en/load-unit/small-parts-piece-picking>
- Killeen, P., Ding, B., Kiringa, I., Yeap, T., & Edi, I. (2019). IoT-based predictive maintenance maintenance for fleet management. *Procedia*

- Computer Science*, 151(2018), 607–613.
<https://doi.org/10.1016/j.procs.2019.04.184>
- Killeen, P., & Parvizimosaed, A. (2018). *An AHP-Based Evaluation of Real-Time Stream Processing Technologies in IoT*.
- Kotsiantis, S. (2007). Supervised Machine Learning: A Review of Classification Techniques. In *Informatica* (Vol. 31).
- Lee, J., Ardakani, H. D., Yang, S., & Bagheri, B. (2015). Industrial Big Data Analytics and Cyber-physical Systems for Future Maintenance & Service Innovation. *Procedia CIRP*, 38, 3–7.
<https://doi.org/10.1016/j.procir.2015.08.026>
- Li, L., Lin, J., Ouyang, Y., & Luo, X. (Robert). (2022). Evaluating the impact of big data analytics usage on the decision-making quality of organizations. *Technological Forecasting and Social Change*, 175.
<https://doi.org/10.1016/j.techfore.2021.121355>
- Liu, X., Ding, Y., Tang, H., & Xiao, F. (2021). A data mining-based framework for the identification of daily electricity usage patterns and anomaly detection in building electricity consumption data. *Energy and Buildings*, 231. <https://doi.org/10.1016/j.enbuild.2020.110601>
- Memala, W. A., Bhuvaneshwari, C., Mana, S. C., Selvan, M. P., Maniraj, M., & Kishore, S. (2021). An approach to remote condition monitoring of electrical machines based on IOT. *Journal of Physics: Conference Series*, 1770(1).
<https://doi.org/10.1088/1742-6596/1770/1/012023>
- Mena, R., Viveros, P., Zio, E., & Campos, S. (2021). An optimization framework for opportunistic planning of preventive maintenance activities. *Reliability Engineering and System Safety*, 215.
<https://doi.org/10.1016/j.ress.2021.107801>
- Mohan, R., Roselyn, P., Uthra, A., Devaraj, D., & Umachandran, K. (2021). Intelligent machine learning based total productive maintenance approach for achieving zero downtime in industrial machinery. *Computers and*

- Industrial Engineering*, 157(March), 107267.
<https://doi.org/10.1016/j.cie.2021.107267>
- Mohr, F., & Wever, M. (2021). *Naive Automated Machine Learning -- A Late Baseline for AutoML*. <http://arxiv.org/abs/2103.10496>
- Nachiappan, R., Javadi, B., Calheiros, R. N., & Matawie, K. M. (2017). Cloud storage reliability for Big Data applications: A state of the art survey. In *Journal of Network and Computer Applications* (Vol. 97, pp. 35–47). Academic Press. <https://doi.org/10.1016/j.jnca.2017.08.011>
- Nakajima, S. (1984). *Introducción al TPM*. Tecnologías de Gerencia y Producción S.A.
- Nguyen, K. T. P., & Medjaher, K. (2019). A new dynamic predictive maintenance framework using deep learning for failure prognostics. *Reliability Engineering and System Safety*, 188(February), 251–262. <https://doi.org/10.1016/j.ress.2019.03.018>
- Parikh, Y., & Mahamuni, P. (2015). Total Productive Maintenance: Need & Framework. In *International Journal of Innovative Research in Advanced Engineering (IJIRAE)* (Vol. 2). www.ijirae.com
- Pfeiffer, S. (2017). The Vision of “Industrie 4.0” in the Making—a Case of Future Told, Tamed, and Traded. *NanoEthics*, 11(1), 107–121. <https://doi.org/10.1007/s11569-016-0280-3>
- Prayogo, A. (2020). *Analysis of Total Effective Equipment Performance SD5 Machine on Krosok Production Line, Primary Manufacturing Department. (Case Study: PT NT)*. 1(1), 0–10. <http://journal.UMK.ac.id/index.php/jointech>
- Ramezan, C. A., Warner, T. A., Maxwell, A. E., & Price, B. S. (2021). Effects of training set size on supervised machine-learning land-cover classification of large-area high-resolution remotely sensed data. *Remote Sensing*, 13(3), 1–27. <https://doi.org/10.3390/rs13030368>

- Rani, S., & Sikka, G. (2012). Recent Techniques of Clustering of Time Series Data: A Survey. In *International Journal of Computer Applications* (Vol. 52, Issue 15).
- Roda, I., & Macchi, M. (2021). Maintenance concepts evolution: a comparative review towards Advanced Maintenance conceptualization. In *Computers in Industry* (Vol. 133). Elsevier B.V. <https://doi.org/10.1016/j.compind.2021.103531>
- Ruiz-sarmiento, J., Monroy, J., Moreno, F., & Galindo, C. (2020). A predictive model for the maintenance of industrial machinery in the context of industry 4.0. *Engineering Applications of Artificial Intelligence*, 87(October 2019), 103289. <https://doi.org/10.1016/j.engappai.2019.103289>
- Salah, B., Janeh, O., Bruckmann, T., & Noche, B. (2015). Improving the performance of a new storage and retrieval machine based on a parallel manipulator using FMEA analysis. *IFAC-PapersOnLine*, 28(3), 1658–1663. <https://doi.org/10.1016/j.ifacol.2015.06.324>
- Sawalha, S., & Al-Naymat, G. (2021). Towards an efficient big data management schema for IoT. *Journal of King Saud University - Computer and Information Sciences*. <https://doi.org/10.1016/j.jksuci.2021.09.013>
- Shankarrao Patange, G., & Bharatkumar Pandya, A. (2022). How artificial intelligence and machine learning assist in industry 4.0 for mechanical engineers. *Materials Today: Proceedings*. <https://doi.org/10.1016/j.matpr.2022.08.201>
- Shet, S. v., & Pereira, V. (2021). Proposed managerial competencies for Industry 4.0 – Implications for social sustainability. *Technological Forecasting and Social Change*, 173. <https://doi.org/10.1016/j.techfore.2021.121080>
- Silvestri, L., Forcina, A., Introna, V., Santolamazza, A., & Cesarotti, V. (2020). Maintenance transformation through Industry 4.0 technologies: A systematic literature review. *Computers in Industry*, 123. <https://doi.org/10.1016/j.compind.2020.103335>

- Sivarajah, U., Kamal, M. M., Irani, Z., & Weerakkody, V. (2017). Critical analysis of Big Data challenges and analytical methods. *Journal of Business Research*, 70, 263–286. <https://doi.org/10.1016/j.jbusres.2016.08.001>
- Smith, R., & Mobley, R. K. (2008a). MTBF User Guide: Measuring Mean Time between Failures. *Rules of Thumb for Maintenance and Reliability Engineers*, 283–284. <https://doi.org/10.1016/B978-075067862-9.50018-6>
- Smith, R., & Mobley, R. K. (2008b). Total Productive Maintenance. *Rules of Thumb for Maintenance and Reliability Engineers*, 107–120. <https://doi.org/10.1016/B978-075067862-9.50008-3>
- Tay, S. I., Alipal, J., & Lee, T. C. (2021). Industry 4.0: Current practice and challenges in Malaysian manufacturing firms. *Technology in Society*, 67. <https://doi.org/10.1016/j.techsoc.2021.101749>
- Tinga, T., & Loendersloot, R. (2019). Physical model-based prognostics and health monitoring to enable predictive maintenance. In *Predictive Maintenance in Dynamic Systems: Advanced Methods, Decision Support Tools and Real-World Applications* (pp. 313–353). Springer International Publishing. https://doi.org/10.1007/978-3-030-05645-2_11
- Tornede, T., Tornede, A., Wever, M., Mohr, F., & Hüllermeier, E. (2020). AutoML for Predictive Maintenance: One Tool to RUL Them All. *Communications in Computer and Information Science*, 1325, 106–118. https://doi.org/10.1007/978-3-030-66770-2_8
- Tortorella, G. L., Fogliatto, F. S., Cauchick-Miguel, P. A., Kurnia, S., & Jurburg, D. (2021). Integration of Industry 4.0 technologies into Total Productive Maintenance practices. *International Journal of Production Economics*, 240. <https://doi.org/10.1016/j.ijpe.2021.108224>
- Xu, X., Lu, Y., Vogel-Heuser, B., & Wang, L. (2021). Industry 4.0 and Industry 5.0—Inception, conception and perception. *Journal of Manufacturing Systems*, 61, 530–535. <https://doi.org/10.1016/j.jmsy.2021.10.006>
- Yan, H. C., Zhou, J. H., & Pang, C. K. (2016). Machinery degradation inspection and maintenance using a cost-optimal non-fixed periodic strategy. *IEEE*

Transactions on Instrumentation and Measurement, 65(9), 2067–2077.
<https://doi.org/10.1109/TIM.2016.2563998>

Zaki, M. J., & Meira, W. Jr. (2020). *Data Mining and Machine Learning: Fundamental Concepts and Algorithms* (Second Edition). Cambridge University Press.

Zhang, J. Z., Srivastava, P. R., Sharma, D., & Eachempati, P. (2021). Big data analytics and machine learning: A retrospective overview and bibliometric analysis. In *Expert Systems with Applications* (Vol. 184). Elsevier Ltd.
<https://doi.org/10.1016/j.eswa.2021.115561>

Zonta, T., André da Costa, C., da Rosa Righi, R., José de Lima, M., Silveira da Trindade, E., & Pyng Li, G. (2020). Predictive maintenance in the Industry 4.0: A systematic literature review. *Computers & Industrial Engineering*, 150(October), 106889. <https://doi.org/10.1016/j.cie.2020.106889>