Universidad Nacional de Ingeniería

Facultad de Ingeniería Mecánica

TRABAJO DE SUFICIENCIA PROFESIONAL

Diseño de un sistema de predicción de fallas de motor de camión minero basado en algoritmos de aprendizaje

Para obtener el Título Profesional de Ingeniero Mecatrónico

Elaborado por

Diego Cris Ashly Rey Tapia

0000-0001-7705-2034

Asesor

Mg. Alcides Guillermo Joo Aguayo

0000-0002-8459-8489

LIMA – PERU

2025

Citar/How to cite	(Rey, 2025)
Referencia/Reference Estilo/Style: APA (7ma ed.)	Rey, D. (2025). *Diseño de un sistema de predicción de fallas de motor de camión minero basado en algoritmos de aprendizaje.* [Tesis, Universidad Nacional de Ingeniería]. Repositorio institucional Cybertesis UNI.

Dedicatoria

A mi valiente mamá. Esta tesis es el resultado de tu amor, apoyo y sacrificio en mi viaje educativo. Las palabras de aliento y de amor que me brindaste cada día han sido mi inspiración. Cada día que trabajaste incansablemente y cada vez que me brindaste tu cariño son tesoros que valoro profundamente, ya puedes cerrar el ciclo conmigo.

Agradecimientos

Este trabajo realizado ha sido gracias al apoyo recibido. Debo un especial reconocimiento a mi asesor, Guillermo Joo, cuyo conocimiento y asesoría han sido cruciales en mi recorrido investigativo. A Violeta Campos, le agradezco por apoyarme en el desarrollo de esta tesis con su tiempo y dedicación, además de creer siempre en mí. A mis compañeros de trabajo, Adolfo Elescano y Leandro Cortez, por sus recomendaciones académicas y técnicas durante el proceso de la elaboración. A todos los mencionados, mi mas sincera gratitud.

Resumen

El siguiente trabajo de investigación tiene como foco el crear un sistema de predicción de fallas de motor de camión minero basado en algoritmos de aprendizaje. Su importancia radica en que la falla imprevista de un motor tiene un alto impacto operacional y económico en la mina, por lo que es necesario implementar un modelo predictivo capaz de poder anticipar esas fallas. Para lograrlo, se realizó un análisis de las principales fuentes de información presentes en la mina y se escogieron dos principales: el análisis de aceite de motor, que permite detectar partículas contaminantes en el motor, y el controlador VIMS que recolecta diferentes contadores del estado del camión. Uniendo esta base de datos, se realizó un preprocesamiento de datos, se separó la muestra de datos a utilizar y se realizó un resumen estadístico. Luego, se procedió con la estandarización de datos y luego se separó la muestra de datos en dos conjuntos, uno de entrenamiento y otro de validación. Para los datos de entrenamiento se evaluaron algoritmos lineales y no lineales, así como algoritmos ensamblados con la finalidad de encontrar el que muestre un mejor desempeño y utilizarlo en los datos de validación.

El resultado esperado es la creación de un sistema capaz de predecir fallas en los motores de camiones mineros con una precisión mayor que la realizada actualmente a través de estadísticas. Esto sería una alternativa que contribuirá a mejorar la disponibilidad de los camiones mineros y reducir los costos de mantenimiento.

Palabras clave — algoritmo de aprendizaje, camiones mineros, estandarización, mantenimiento predictivo, minería, motor de combustión.

Abstract

This research project aims to develop a prediction system for mining truck engine failures based on learning algorithms. Given the high criticality of engines in mining operations and their economic impact, the goal is to optimize equipment availability by implementing a predictive model that can anticipate failures and reduce the costs associated with unscheduled maintenance.

To achieve this, an analysis was conducted on the main information sources available at the mine, and two key ones were selected: the oil analysis, which detects contaminant particles in the engine, and the VIMS controller, which collects various pressure, temperature, and truck status data. These datasets were merged, and a data preprocessing phase was carried out. The data sample was then split, and a statistical summary was performed. Next, the data was standardized to better apply the proposed algorithms. Additionally, the dataset was divided into two subsets: one for training and another for validation. For the training data, both linear and nonlinear algorithms, as well as ensemble algorithms, were evaluated to identify the one with the best performance for use on the validation set.

The expected outcome is the development of a system capable of predicting failures in mining truck engines with greater accuracy than current statistical methods. This would offer an alternative that helps improve truck availability, reduce maintenance costs, and thereby increase the operational efficiency of mining operations.

Keywords — learning algorithm, mining trucks, standardization, predictive maintenance, mining, combustion engine.

Tabla de Contenido

Lista de Tablas

Pág.

Lista de Figuras

Introducción

El presente trabajo de suficiencia profesional tiene como objetivo poder predecir con cierto grado de exactitud la falla del motor de camiones mineros a través del uso del aprendizaje supervisado basados en algoritmos lineales, no lineales y ensamblados. Este trabajo abarca los siguientes capítulos:

En el Capítulo I, se presentan las generalidades del estudio y los conceptos fundamentales. Se describe la identificación del problema, así como el objetivo general y los objetivos específicos, que servirán como guía durante el desarrollo del trabajo. Asimismo, se formulan la hipótesis general y las hipótesis específicas, junto con la operacionalización de variables dependientes e independientes, incluyendo sus respectivos indicadores. También, se describe la metodología explicando el tipo, diseño y técnicas de e instrumentos para la investigación. Finalmente se mostrará el paso a paso del análisis y procesamiento de los datos.

En el Capítulo II, se ilustra el marco teórico y conceptual, en el marco teórico muestra los principales conceptos necesarios para comprender el trabajo, tales como, la definición del aprendizaje supervisado y los distintos tipos de algoritmos que comprende. Además, se revisarán definiciones de transformación de datos y selección de variables. Por otro lado, en el marco conceptual se especificará los principios de la gestión de mantenimiento de maquinaria pesada, así como, el análisis de aceite y sistemas de información vital del motor de camión minero CAT 797F.

En el Capítulo III, se sustenta la elaboración del trabajo de investigación. Inicialmente, se explican las fuentes de donde se han adquirido los datos para el trabajo, los cuales son: el análisis de aceite de motor a través del sistema SOS y la descarga de datos del controlador VIMS. De esos datos, se toma una muestra la cual es procesada a

través del uso de librerías en el software Python. Esta muestra es evaluada utilizando diversos algoritmos (lineales, no lineales y ensamblados) con el fin de predecir, a través del aprendizaje supervisado, posibles fallas en los motores. Para ello, los datos se dividen en dos conjuntos: uno de entrenamiento y otro de validación. Finalmente, se compara el rendimiento de cada algoritmo y se selecciona el de mejor desempeño para su aplicación en el conjunto de validación.

El Capítulo IV expone los resultados obtenidos con el algoritmo que mostró mejor rendimiento en el conjunto de validación. Además, se valida la comprobación de las hipótesis planteadas y una discusión de los resultados encontrado. Para cerrar, se presentan las conclusiones, recomendaciones, referencias bibliográficas y anexos del trabajo de suficiencia profesional.

Capítulo I. Parte introductoria del trabajo

1.1 Generalidades

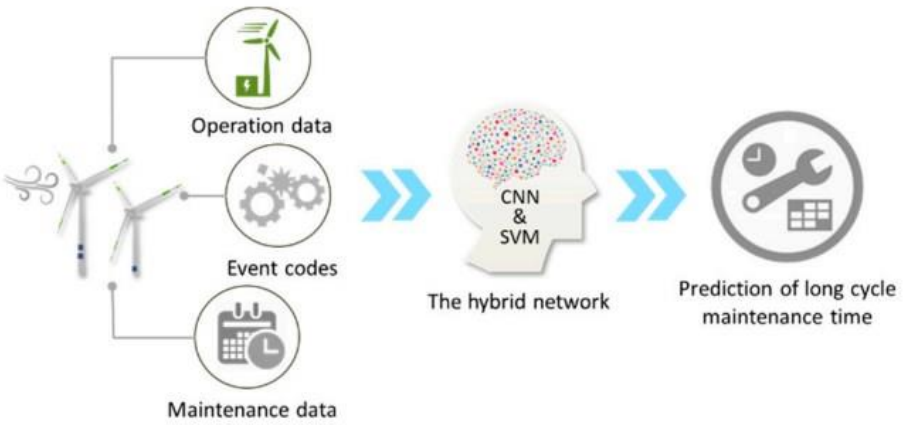
Carvalho, Pereira y Cardoso (2019), en su trabajo de investigación sobre la aplicabilidad de sistemas automáticos realiza una visión exhaustiva sobre los principales métodos y métricas. Esta investigación se enmarca en la creciente preocupación por el uso de modelos predictivos altamente precisos pero opacos, conocidos como *modelos caja negra*, los cuales dificultan la comprensión de sus decisiones tanto por parte de usuarios técnicos como no técnicos. Carvalho et al. (2019), proponen una clasificación de los enfoques de interpretabilidad que permite comprender las diversas formas en que se pueden explicar los modelos, según el momento en que se aplican (antes, durante o después del entrenamiento del modelo), el tipo de información generada (explicaciones globales o locales), y su nivel de dependencia del modelo base (agnósticos o específicos). Esta clasificación proporciona un marco robusto en la forma de interpretación de los modelos, además su correcto uso dependiendo el tipo de problema, el público objetivo y su aplicación. También, esta investigación revisa la aplicación de machine learning en contextos industriales donde no solo se busca una buena precisión sino ser un apoyo para justificar decisiones del proceso. Finalmente, refuerza la pertinencia de investigar metodologías que permitan integrar transparencia algorítmica en procesos industriales automatizados.

Yeh, Lin, Lin, Yu y Chen (2019) desarrollaron un modelo de predicción de mantenimiento prolongado en turbinas eólicas utilizando técnicas de *machine learning*, específicamente una red híbrida basada en redes neuronales convolucionales (CNN) y máquinas de soporte vectorial (SVM). Su investigación, se baseo en recopilar la data a través de sensores de 31 turbinas durante 941 días, con esto logró identificar variables importantes como la velocidad del viento y su potencia generada, la temperatura del aceite en los rodamientos y la cantidad de eventos críticos reportados. Después de un análisis y

selección de datos, el modelo propuesto alcanzo una precisión mayor al 70% en el periodo de mantenimiento de largo ciclo, lo que ayudara a mejorar la planificación de paradas programadas y reducir costos operativos. Este trabajo es excelente como base para la presente investigación, ya que su modelo se orienta a aumentar la disponibilidad de equipos a través de la predicción de fallas, con la finalidad de mejorar la eficiencia en operaciones mineras. La metodología aplicada por Yeh et al. (2019), brinda un ejemplo de aprendizaje automaticoen contextos industriales complejos, y que la identificación temprana de fallas permite tomar mejores decisiones en el mantenimiento.

Figura 1

La arquitectura de la predicción del tiempo de mantenimiento de ciclo largo (Yeh, 2019)

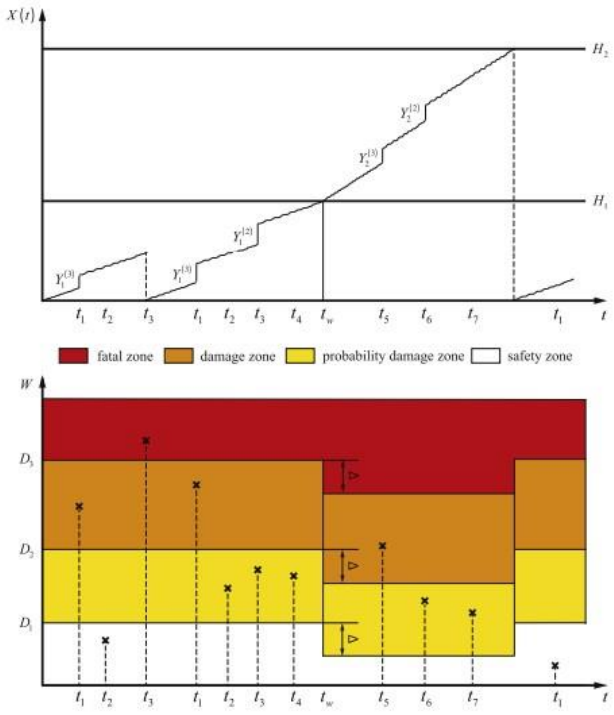


Wei, Zhao, He y He (2018), proponen un modelo de confiabilidad para sistemas mecánicos degradables en ambientes industriales, integrando un enfoque de mantenimiento basado en condición (condition-based maintenance, CBM) que considera efectos de impactos externos categorizados por zonas. En este modelo, el sistema es tratado como una entidad con dos estados (normal y debilitado) cuya transición está determinada por un proceso de degradación estocástico (proceso de Wiener en dos fases) y por la exposición a eventos de choque (shocks) con distintos niveles de severidad. Estos choques son clasificados en zonas de seguridad, daño parcial, daño crítico y fallas catastróficas, dependiendo de su magnitud y del estado actual del sistema. El modelo

incorpora estrategias de mantenimiento que consideran reparaciones imperfectas, reemplazos preventivos y correctivos, optimizadas con base en la minimización del costo promedio a largo plazo. A través de simulaciones numéricas, los autores demuestran cómo esta metodología puede mejorar la disponibilidad y confiabilidad de equipos industriales sujetos a entornos altamente dinámicos, como micro-motores o sistemas mecánicos en condiciones de operación extremas. Este antecedente es particularmente relevante para el presente trabajo de investigación, cuyo objetivo es diseñar un sistema predictivo para anticipar fallas en motores de camiones mineros. Al igual que en el modelo de Wei et al. (2018), los sistemas mineros operan bajo condiciones extremas que aceleran su deterioro y los exponen a fallas tanto suaves (por desgaste) como críticas (por eventos repentinos).

Figura 2

Shocks zonificados bajo el sistema de estado binario (Wei, 2018)

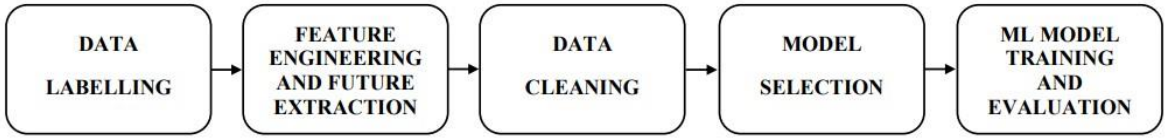


El estudio de Orrù y Zoccheddu. (2020), representa un importante antecedente para el presente trabajo de investigación. En dicho estudio, los autores desarrollaron un sistema de mantenimiento predictivo aplicando algoritmos de aprendizaje automático,

específicamente Multilayer Perceptron (MLP) y Support Vector Machine (SVM), con el objetivo de anticipar fallas en bombas centrífugas utilizadas en la industria del petróleo y gas. El enfoque metodológico empleado es altamente pertinente para el proyecto actual, ya que también se basa en el análisis de datos operativos obtenidos de sensores (temperatura, presión y vibración), el preprocesamiento y la ingeniería de características, así como en la división del conjunto de datos en muestras de entrenamiento y validación. En adición, se muestra técnica para poder balancear las clases, un ejemplo es el algoritmo SMOTE; asimismo, se presenta el rendimiento de los modelos a través de métricas como precisión, recall y F1-score. Su importancia radica en la viabilidad de usar modelos basado en machine learning para el mantenimiento predictivo, el cual brinda una base metodológica para el desarrollo del sistema predictivo en la presenta investigación.

Figura 3

Esquema de entrenamiento y validación de un modelo (Orrú,2020)

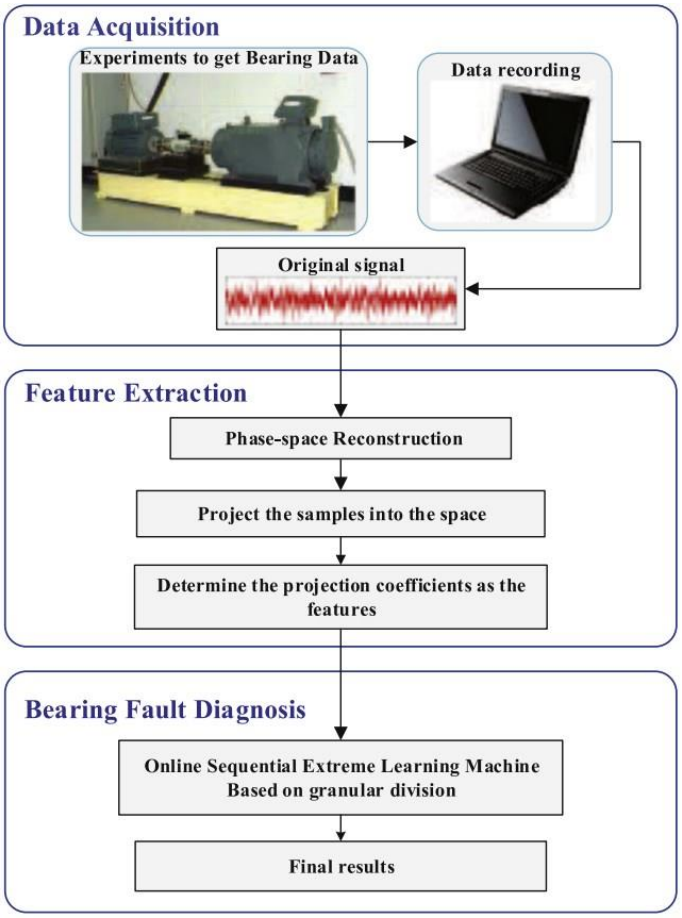


El trabajo de Mao (2016), es un significativo antecedente para la investigación actual. En su trabajo, se propone un enfoque basado en algoritmos de machine learning de tipo extremo en línea (OS-ELM), para diagnosticar fallas en rodamientos en el sector industrial. La base de datos trabajada era altamente desbalanceada y de forma secuencial, lo cual es común en ambientes industriales. El procedimiento utilizado combina técnicas de pre-procesamiento de datos, tales como la extracción de características a partir de señales de vibración mediante EMD y WPT, también se aplico algoritmos de balance como SMOTE y una arquitectura secuencial que actualiza dinámicamente el modelo predictivo. Esto guarda una relación con el presente trabajo, debido a que también realiza predicción de fallas en componentes mecánicos, mediante el análisis de datos electrónicos como los sensores y se aplica machine learning. Finalmente el uso del enfoque OS-ELM y la

estratégica de granulación de datos brinda un soporte metodológico importante, además, la atención en el desbalanceo de clases refuerza la relevancia de tener métricas mas sensibles para la clase minoritaria.

Figura 4

Diagrama de flujo total del método de diagnóstico propuesto (Mao, 2016)

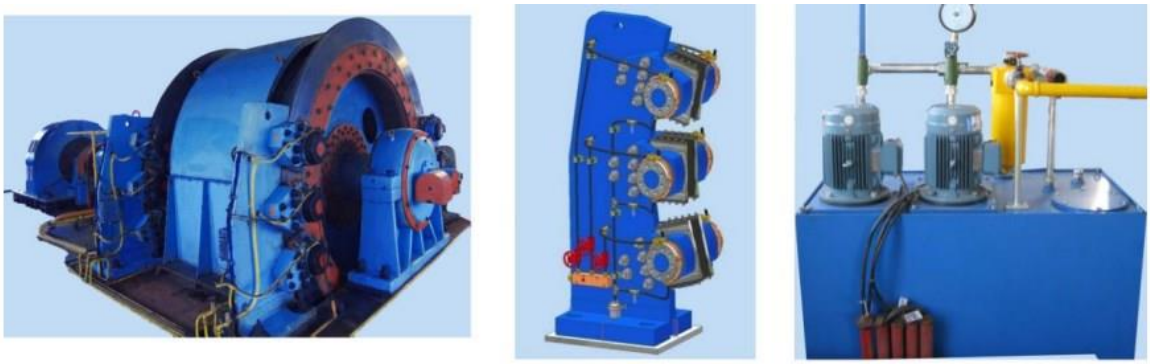


El estudio desarrollado por Li, Jiang, Li y Xie (2020) propone un sistema de diagnóstico de fallas en frenos de discos mineros utilizando algoritmos de aprendizaje automático, incorporando el coeficiente de concordancia de Kendall para mejorar la precisión diagnóstica. Este trabajo es relevante porque aborda, al igual que esta tesis, el desafío de predecir fallas en sistemas complejos de maquinaria minera a partir de datos de sensores (temperatura, presión, desplazamiento, velocidad de izaje, entre otros). Además, ambos estudios emplean un enfoque de minería de datos para extraer reglas de

diagnóstico, realizar el procesamiento de grandes volúmenes de datos y evaluar modelos predictivos con fines operativos y de mantenimiento. Entre las contribuciones más destacadas del estudio se encuentra la creación de un modelo dinámico de toma de decisiones que mejora significativamente la precisión diagnóstica frente a métodos tradicionales, pasando de un 85.4% con el algoritmo original a un 95.85% con la versión optimizada, demostrando su utilidad práctica en entornos industriales reales. Esto respalda la propuesta de esta investigación sobre el desarrollo de un sistema predictivo para motores de camiones mineros.

Figura 5

Estructura de un sistema de freno en minería (Juanli, Shuo, Menghui, & Jiacheng, 2020)



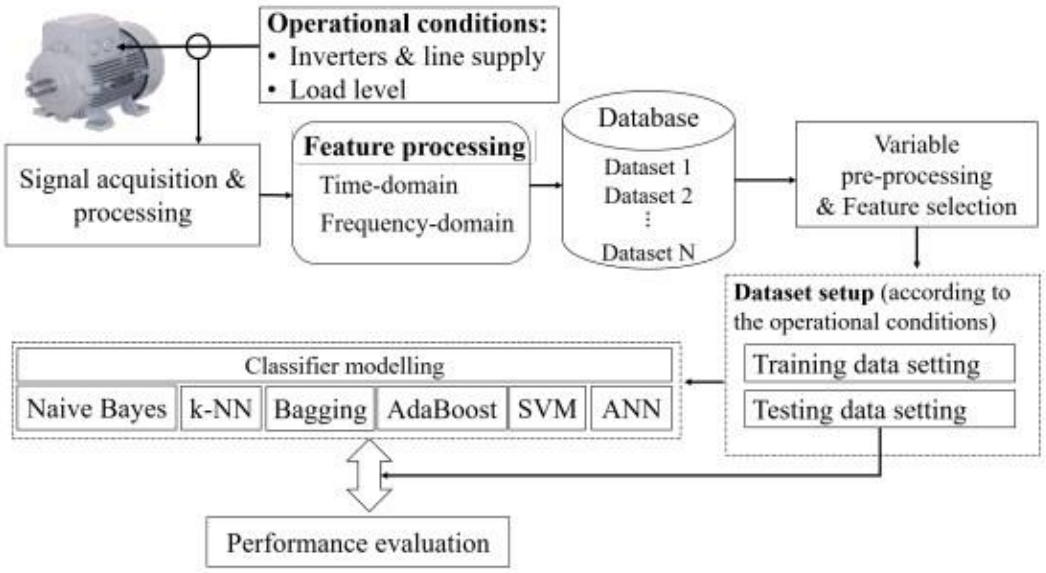
(a) Mine hoist system (b) Brake (c) Hydraulic station

El estudio de Martín-Díaz, Morinigo-Sotelo, Duque-Pérez y Romero-Troncoso (2020) constituye un antecedente relevante para el presente trabajo, ya que presenta una evaluación experimental comparativa de diversas técnicas de aprendizaje automático aplicadas al diagnóstico de fallas en motores de inducción, bajo diferentes condiciones operativas. El objetivo principal fue determinar la robustez y precisión de distintos clasificadores, incluyendo Naive Bayes, k-NN, SVM, MLP, Bagging y AdaBoost, al detectar fallas incipientes y severas en rotores con alimentación por distintos tipos de inversores. Este enfoque resulta estrechamente vinculado con la propuesta de esta tesis, dado que ambos trabajos se centran en la aplicación de inteligencia artificial para predecir fallas en maquinaria rotativa, considerando condiciones reales de operación como variaciones en la

carga y el tipo de alimentación. Este estudio demuestra que métodos como Bagging y Naive Bayes ofrecen mejores resultados en entornos ruidosos y con datos no vistos durante el entrenamiento, lo cual refuerza la necesidad de seleccionar modelos robustos y bien generalizados, como se plantea también en esta investigación. Además, la metodología incluye fases de adquisición de señales, extracción de características en el dominio del tiempo y frecuencia, selección de variables y validación mediante métricas como la curva ROC y el AUC, herramientas que también se emplean en este proyecto para garantizar la precisión del sistema predictivo desarrollado.

Figura 6

Metodología de diagnóstico de falla en motores de inducción (Martin-Diaz, 2020)



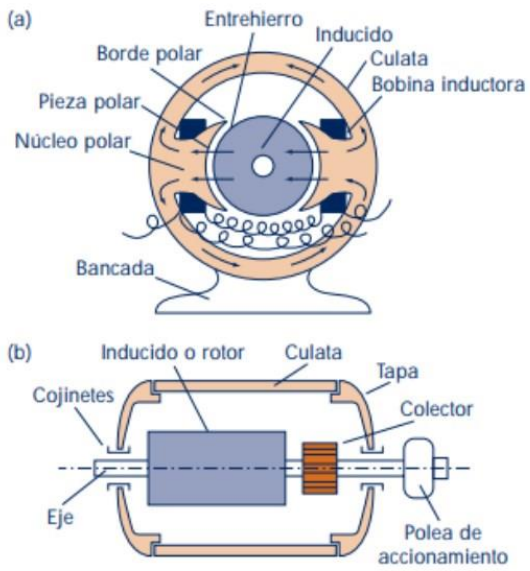
La tesis de Adauto Arana (2021), titulada Aplicación de la inteligencia artificial en la detección de fallas en los motores eléctricos de corriente continua de imán permanente, representa un valioso antecedente nacional en el campo del mantenimiento predictivo mediante inteligencia artificial. El autor plantea un análisis comparativo entre métodos tradicionales y algoritmos basados en machine learning para diagnosticar fallas en máquinas eléctricas rotativas, evaluando mejoras en la fiabilidad y eficiencia del procedimiento de ubicación y reconocimiento de averías.

Este trabajo guarda una relación directa con el presente proyecto, dado que ambos comparten la finalidad de optimizar la vida útil de componentes electromecánicos críticos, como los motores, mediante la aplicación de algoritmos inteligentes. Asimismo, la tesis emplea indicadores técnicos como el sistema de aislamiento, el análisis de vibraciones mecánicas y el comportamiento eléctrico del motor para desarrollar modelos predictivos basados en técnicas como regresión lineal, redes neuronales y árboles de decisión, lo cual es coherente con la metodología utilizada en el presente estudio.

Además, al tratarse de una investigación aplicada con diseño preexperimental, su enfoque metodológico resulta compatible con el desarrollo de prototipos predictivos en entornos industriales reales, como el de los camiones mineros, objetivo de esta tesis. La relevancia de este trabajo radica en su contribución práctica al ámbito del mantenimiento inteligente de maquinaria industrial en el Perú.

Figura 7

Clasificación de máquinas eléctricas rotativas (ADAUTO ARANA, 2021)



La tesis de Herrera Zeballos (Herrera Zeballos, 2021) titulada Método de gestión de mantenimiento centrado en la confiabilidad para mejorar la disponibilidad de los motores C175-16 en la flota 793F del proyecto minero Constancia – Cusco, constituye un aporte

fundamental al ámbito del mantenimiento predictivo aplicado a flotas mineras. En esta investigación se desarrolla e implementa una metodología basada en el enfoque Reliability-Centered Maintenance (RCM), con el objetivo de mejorar la disponibilidad operativa de los motores diésel Caterpillar C175-16, mediante un análisis sistemático de criticidad, modos de falla y efectos asociados. El estudio guarda una relación directa con el presente proyecto, ya que ambos buscan optimizar el mantenimiento de componentes críticos de camiones mineros utilizando herramientas de análisis estructurado y datos operacionales. Herrera plantea una evaluación completa del contexto operativo, aplicando herramientas como el Análisis de Modos y Efectos de Falla (AMEF), la jerarquización de sistemas según Pareto y el uso de hojas de decisión RCM, todo con el propósito de establecer un plan de mantenimiento preventivo y predictivo eficiente. Este antecedente es particularmente relevante porque valida el impacto positivo de aplicar metodologías de mantenimiento basadas en la confiabilidad, mostrando resultados medibles en indicadores clave como disponibilidad, tiempo medio entre fallas (MTBF) y eficiencia de mantenimiento.

Figura 8

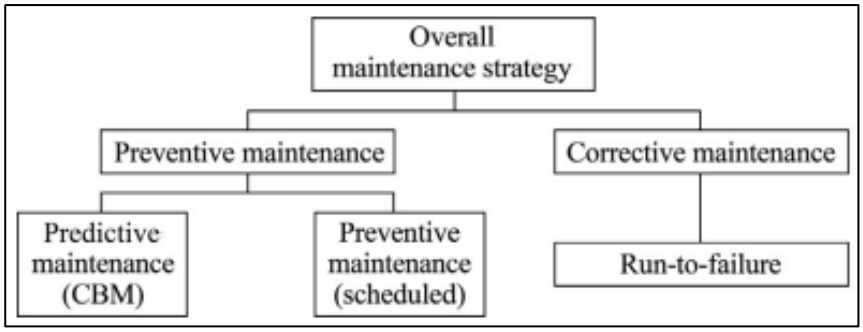
Camión Caterpillar 793D (Herrera Zeballos, 2021)

1.2 Identificación y Descripción del problema de investigación

La productividad de una mina depende significativamente de la planificación de la producción, la cual consiste en perforación, voladura, carga y transporte (Choi, Hoang, Xuan-Nam, & Trung Nguyen, 2020). Particularmente, es indispensable poder mantener operativo los equipos de transporte, principalmente camiones, ya que representan más de 30% del total de costo de producción. (Alarie & Gamache, 2002). Por lo que asegurar el correcto desempeño de la flota de acarreo(transporte) podría representar un significante ahorro para la compañía minera. Por esta razón, las compañías mineras realizan grandes esfuerzos para mantener un alto nivel de disponibilidad en sus equipos. Para ello, se aplica una estrategia de mantenimiento preventivo, la cual se basa en un monitoreo de condiciones donde se recopila y evalúa el estado del equipo en tiempo real y con esta información tomar decisiones de mantenimiento y programa actividades de mantenimiento. (Alaswad & Xlang, 2017).

Figura 9

Estrategias de Mantenimiento básicas (Alaswad & Xlang,2017)

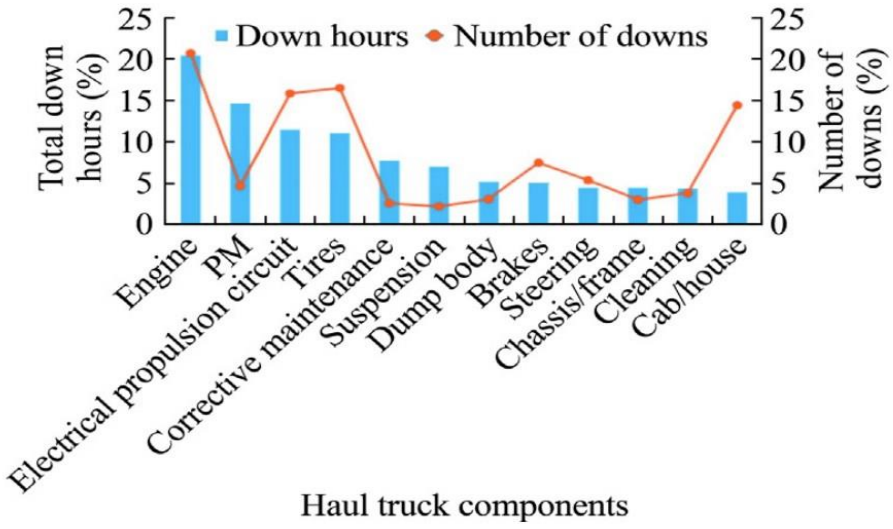


Además, debido a investigaciones realizada se sabe que la causa más frecuente de parada no programada de un camión minero es debido a una falla en el funcionamiento del motor, esto genera que él equipo se encuentre inactivo, retrasen la producción de la mina y genera grandes pérdidas económicas (Reddy Alla, Hall, & Apel, 2019). Por ende, el área de monitoreo de condiciones dentro de las empresas mineras trabaja tomando

algunos parámetros críticos como temperatura del aceite de motor, presión, velocidad de rotación (RPM) y temperatura de salida de combustión para poder monitorear el estado del motor.

Figura 10

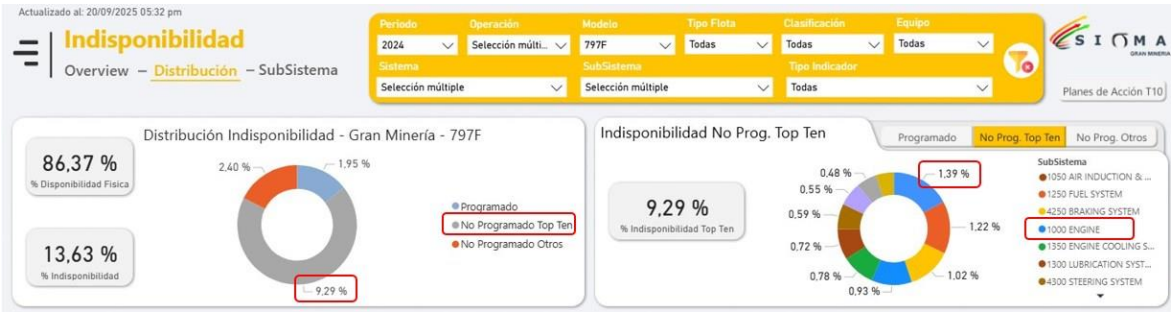
Principales fuentes de paradas en camiones (Reddy Alla, Hall, & Apel, 2019)



En el caso de la actualidad de camiones CAT trabajando, se puede observar que, durante el 2024 la principal causa de paradas no programadas en los camiones mineros CAT 797F High Altitud presentes en la mina Las Bambas y Toromocho fue el sistema del motor.

Figura 11

Indisponibilidad de la flota 797F

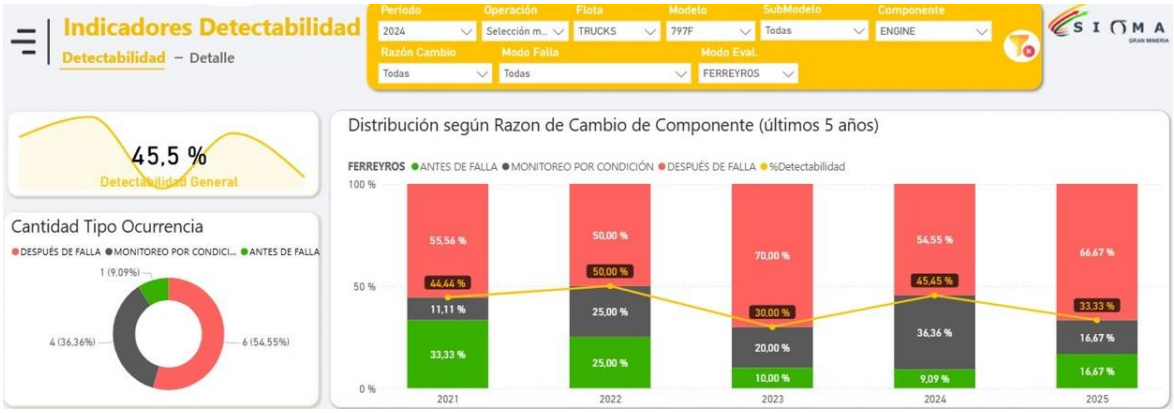


Nota:: Ferreyros S.A.

Además, los actuales sistemas de predictibilidad basados en monitoreo de condiciones y análisis estadísticos han arrojado los sistemas estadísticos muestran que durante el 2024 la predictibilidad de falla de motor fue de 45.4% y para el 2025 se encuentra en 33.3%.

Figura 12

Actual sistema de detectabilidad de falla de motores



Nota: Ferreyros S.A.

1.3 Formulación del Problema

1.3.1 Problema Principal

¿Cómo mejorar la disponibilidad de un equipo de acarreo de gran minería, a través de la predictibilidad de falla del motor, con un adecuado diseño metodológico y con mejor performance?

1.3.2 Problema Específicos:

- ¿Cómo realizar un procesamiento adecuado de los datos a trabajar?
- ¿Qué algoritmos trabajaran con una mejor performance en la predictibilidad de fallas de motores de camiones mineros?

- ¿La predicción de fallas de motores a través de machine learning es una alternativa viable para ser usada en el mantenimiento predictivo de motores de camiones mineros?

1.3.3 Justificación e Importancia:

Esta investigación se justifica al abordar la necesidad imperiosa de elevar la disponibilidad operativa de los equipos de acarreo en la gran minería, específicamente los camiones. La interrupción imprevista del motor de estos activos se traduce en graves perjuicios financieros, derivados de la paralización productiva, los elevados costos de reparaciones de emergencia y los riesgos de seguridad operacional. Los esquemas de gestión de mantenimiento tradicionales basados en correctivos o preventivo por intervalos demuestran ser insuficientes. La solución propuesta, basada en la predicción de fallas del motor mediante Aprendizaje Automático, ofrece una vía avanzada para migrar hacia un mantenimiento predictivo más eficaz y con un rendimiento superior.

1.4 Objetivos:

1.4.1 Objetivo Principal:

Diseñar de un sistema de predicción de falla de motor de un equipo de acarreo de gran minería basado en algoritmos de aprendizaje supervisado.

1.4.2 Objetivos Secundarios:

- Realizar un preprocesamiento de los datos obtenidos en un adecuado diseño metodológico.
- Validar que tipos de algoritmos de aprendizaje supervisado trabajan con una mejor performance en la base de datos disponible y validar el tiempo de ejecución de estos.

- Mostrar que la predicción de falla de motores a través de machine learning es una adecuada forma mejorar el mantenimiento predictivo.

1.5 Hipótesis y operacionalización de variables

1.5.1 Hipótesis General:

El uso de algoritmos de aprendizaje para la predicción de fallas de motor ayudará a mejorar la predictibilidad de falla de un equipo de acarreo de gran minería.

1.5.2 Hipótesis Específicas:

- Un adecuado preprocesamiento de los datos, que incluya limpieza, normalización y selección de variables relevantes, mejora significativamente la precisión de los modelos de predicción de fallas de motores en equipo de acarreo de gran minería.
- Entre los algoritmos de aprendizaje supervisado aplicados a la base de datos disponible, aquellos de tipo ensamblado presentan una mejor performance predictiva y una relación más eficiente entre precisión y tiempo de ejecución, en comparación con algoritmos lineales y no lineales.
- El uso de modelos de aprendizaje automático supervisado permite estimar con mayor precisión el porcentaje de fallas de motores de camiones mineros en un horizonte de tiempo determinado, en comparación con los métodos estadísticos tradicionales

1.6 Operacionalización de variables

La tabla 1 contiene la operacionalización de la variable independiente, la arquitectura de recolección de datos, mientras que la tabla 2 presenta la operacionalización de la variable dependiente, el estado de salud del motor.

Tabla 1

Operacionalización de variable independiente

VARIABLE INDEPENDIENTE	DEFINICIÓN CONCEPTUAL	DEFINICIÓN OPERACIONAL	INDICADORES
Rendimiento del modelo de pronóstico de falla basado en Machine Learning	El rendimiento de un algoritmo es la cantidad de líneas, tiempo de ejecución y uso de memoria que necesita para llegar una respuesta correcta. Mientras menor sea la cantidad, tendrá más rendimiento	El Tiempo de entrenamiento es la cantidad de tiempo de ejecución para analizar la data El tiempo de predicción es la cantidad de tiempo que se necesita para que el algoritmo muestre una predicción con la data de validación	Tiempo de Entrenamiento Tiempo de Predicción

Nota: elaboración propia

Tabla 2

Operacionalización de variable dependiente

VARIABLE DEPENDIENTE	DEFINICIÓN CONCEPTUAL	DEFINICIÓN OPERACIONAL	INDICADORES
Precisión del algoritmo de predicción de fallas	Es el modelo de aprendizaje supervisado, el cual contiene algoritmos que, alimentados por información de monitoreo, podrá analizar el estado del motor y proyectar una futura falla una respuesta correcta. Mientras menor sea la cantidad, tendrá más rendimiento.	Sistema programado para predicción de falla	Desviación estándar del algoritmo Porcentaje de accuracy del algoritmo.
Tiempo de ejecución del algoritmo	Es el período de tiempo que tarda un algoritmo en completar su tarea, desde el inicio hasta el final de su ejecución	Tiempo de ejecución del sistema de predicción	Tiempo de ejecución

Nota: elaboración propia

1.7 Metodología de la investigación

1.7.1 Unidad de análisis

En esta investigación se utilizará como unidad de análisis, los camiones mineros 797F de la marca CAT con la configuración High Altitud.

Figura 13

Camión minero CAT 797F (Adaptado de CAT,2025)

1.7.2 Tipo de investigación

Según Jose Gonzales en su libro Diseño y Metodología de investigación (Gonzales, 2021) Desde el punto de vista del tipo de variables, el trabajo de investigación las siguientes características:

- Tiene un enfoque cuantitativo, debido a que tiene como características epistemológicas ser objetiva, excluyente, inductiva, con finalidad de exploración, orientada al resultado, centrada en diferencias, perspectiva desde afuera y con antecedente específico.
- El nivel de investigación es predictiva, debido a que se extrajeron datos reales de los estados de los camiones para poder predecir a futuro su funcionamiento.

- El tipo de investigación es cuasiexperimental, debido a que no se toman todos los motores en funcionamiento, sino solamente los motores que han fallado.
- De acuerdo con la temporalidad de las variables es de tipo transversal, debido a que las variables son utilizadas por única vez, algunas para el entrenamiento y otras para la validación de los algoritmos.
- La variable independiente es el rendimiento del modelo de pronóstico de falla basado en machine learning, la cual es de tipo continua, mientras que la variable dependiente es la precisión del algoritmo de predicción de fallas
- La hipótesis, es de tipo inductiva, y que se generalizará a partir de los datos tomados.
- La forma de procesamiento y adquisición de datos fue de observación directa, ya que la información de los datos fue descargados a través de fuentes de información de la empresa.
- El diseño de la investigación es explicativo, ya que se establecerá la relación entre el diseño de la metodología de análisis de datos y la predicción de falla del motor.

1.7.3 Diseño de investigación y muestra

El proyecto tiene como objetivo diseñar una metodología de análisis basada en algoritmos de aprendizaje para la predicción de fallas en motor de un camión minero. Para validar la metodología, se utilizará los siguientes conceptos:

- Población: Todos los camiones CAT 797F que están trabajando en Perú
- Muestra: Camiones CAT 797F de la configuración High Altitud que trabajan en las minas Las Bambas y Toromocho
- Unidad: Camión CAT 797F de la configuración High Altitud.

1.7.4 Técnicas e instrumentos de recolección de datos

Para determinar la clase de nuestro algoritmo de aprendizaje supervisado, se llevará a cabo un proceso de recolección exhaustiva de datos provenientes de múltiples fuentes y sistemas relevantes en el entorno operativo de los equipos de acarreo en minería. Dentro de estos, se encuentra el sistema VIMS (Vital Information Management System),el cual permite obtener parámetros operaciones en tiempo real del camión minero CAT; también se encuentra el análisis de aceite.SOS, que ayuda a obtener información de signos de desgaste, contaminación o algún deterioro de los componentes internos del motor; finalmente, un sistema de gestión de mantenimiento donde se interviene las principales acciones como reparación reemplazos de componentes y reporte de fallas.

Una vez recopilada la información, esta se trabaja para obtener una base de datos estructurada. Este procedimiento es esencial para poder mostrar la calidad y principalmente la coherencia de los datos antes del entrenamiento del modelo con algoritmos. Una vez culminado, se identificarán y organizarán las variables claves asociados a cada evento de falla del motor, por ejemplo, las horas de operación del motor, temperatura de escape, análisis de partículas metálicas, entre otros.

El alcance final de esta fase es poder determinar de una manera precisa la clase de cada instancia de datos, es decir, si corresponde a una condición de trabajo normal o se origino por una falla en especifica. Su importancia es que una vez realizado, se podrá entrenar modelos supervisados que puedan aprender a reconocer patrones asociados a las ocurrencias de falla, realizando predicciones más precisas sobre el futuro del motor instalado.

1.7.5 Análisis y procesamiento de datos

Para cumplir con los objetivos establecidos en la presente investigación, se llevó a cabo un proceso de análisis y procesamiento de los datos adquiridos a través del software Anaconda, específicamente la distribución que incorpora Python 3.8. Esta elección

metodológica permitió aprovechar la potencia del lenguaje Python y su vasto ecosistema de bibliotecas (NumPy, Pandas, Scikit-learn, etc.) para tareas cruciales como la limpieza, transformación, modelado y visualización de los datos, asegurando la trazabilidad y reproducibilidad de todos los procesos analíticos y los modelos estadísticos desarrollados para sustentar los hallazgos de la investigación.

Recolección y organización de la información:

La recolección de datos se realizó a partir de tres fuentes principales:

- Datos descriptivos de cada motor, que incluyen variables como horas acumuladas, tipo de operación, configuración del motor y código del equipo.
- Análisis de muestras de aceite, los cuales aportan información sobre el desgaste interno de los componentes del motor mediante la detección de partículas metálicas y contaminantes.
- Registros del sistema VIMS, que contiene eventos relevantes sobre condiciones críticas del motor, tales como presión de aceite, temperatura, velocidad, entre otros.

Toda esta información fue organizada y estructurada, permitiendo vincular los eventos de falla con los indicadores operacionales precedentes. Se construyó una variable objetivo-binaria, diferenciando entre instancias que precedieron a una falla (etiquetadas como “1”) y aquellas que no (etiquetadas como “0”).

Filtración y depuración de datos:

Con el fin de asegurar la relevancia y consistencia de la base de datos, se aplicaron diversos criterios de filtrado, entre los que destacan:

- Considerar únicamente motores CAT 797F con configuración High Altitude.
- Incluir únicamente motores desmontados por fallas correctivas, excluyendo aquellos sometidos a mantenimiento preventivo.

- Excluir registros de motores con menos de 500 horas de operación.
- Eliminar eventos relacionados con accidentes u otras causas fortuitas no asociadas al deterioro del motor.

Estos criterios permitieron asegurar que los datos incluidos en el análisis correspondieran a eventos reales de falla.

Preprocesamiento de datos

El preprocesamiento se ejecutó en el lenguaje de programación Python, utilizando librerías especializadas como pandas, numpy, matplotlib y scikit-learn. Las acciones realizadas incluyeron:

- Conversión y verificación de tipos de datos
- Revisión del balance de clases, confirmando una distribución equilibrada cercana al 50% entre clases de falla y no falla.
- Estandarización y normalización de las variables, dependiendo del algoritmo a aplicar, para garantizar una escala uniforme entre atributos y mejorar el desempeño de los modelos

Análisis exploratorio de datos:

Se realizó un análisis tanto estadístico como gráfico para comprender la distribución y correlación entre las variables:

- Se calcularon medidas de tendencia central, dispersión y rangos intercuartílicos.
- Se generaron histogramas y diagramas de caja para identificar valores atípicos.
- Se construyó una matriz de correlación para identificar atributos redundantes o fuertemente relacionados, lo que ayudó en la selección de características más relevantes para el modelo.

Capítulo II. Marco Teórico y Marco Conceptual

2.1 Bases Teóricas

2.1.1 Machine Learning:

Machine Learning es una rama de la ciencia de los algoritmos computacionales que está diseñada para emular la inteligencia humana aprendiendo del entorno, es la punta de la lanza de la nueva era del Big-Data. Las técnicas basadas en el aprendizaje automático se han aplicado con éxito en diversos campos que van desde el reconocimiento de patrones, la visión por computadora, la ingeniería de naves espaciales, las finanzas, el entretenimiento y la biología computacional hasta las aplicaciones biomédicas y médicas. (El Naqa & Murphy, 2015)

Un algoritmo de aprendizaje automático es un proceso computacional que utiliza datos de entrada para lograr una tarea deseada sin estar literalmente programado para producir un resultado particular. Estos algoritmos son codificados de forma flexible en el sentido de que alteran o adaptan automáticamente su arquitectura a través de la repetición para que sean cada vez mejores en el logro de la tarea deseada. El proceso de adaptación se llama entrenamiento, y sirve para que el algoritmo pueda ser entrenado para poder encontrar los resultados deseados. Una vez entrenado el algoritmo, se configura de manera óptima para que produzca el resultado deseado con los datos de validación, que son datos nunca ha visto el algoritmo. Esta capacitación es el aprendizaje de los algoritmos, lo cual no se limita a solo una adaptación, sino que, al igual que los humanos, un algoritmo debe practicar el aprendizaje de forma permanente a medida que procesa nuevos datos y aprende de sus errores. (El Naqa & Murphy, 2015)

2.1.2 Tipos de algoritmos de machine learning

Aprendizaje Supervisado

El aprendizaje supervisado es una técnica del aprendizaje automático que consiste en que un algoritmo se relacione con un conjunto de variables de entrada (X) y con una salida esperada (Y), utilizando como base, un conjunto de datos que halla revisado previamente. Este proceso implica que existe un supervisor el cual le muestra las respuestas correctas durante su fase de entrenamiento, para que el modelo pueda predecir la salida para nuevos datos que no haya interactuado.

El artículo de Cunningham (s.f.) se describe que el aprendizaje supervisado es muy utilizado, especialmente en casos de clasificación, donde la tarea es asignar etiquetas a nuevos ejemplos con base en el conocimiento adquirido. El aprendizaje supervisado tiene como base teórica la minimización del riesgo, el cual busca reducir el error de la predicción, esto lo logra utilizando una función dentro de un conjunto posible (H), esta función debe representar de una mejor forma la relación entre X e Y según la función de perdida definida. También, el capítulo menciona que hay algoritmos comunes de aprendizaje supervisado, por ejemplo, las máquinas de vectores de soporte (SVM) y los clasificadores de vecinos cercanos (K_-NN), así como algoritmos ensamblados los cuales tiene múltiples modelos comprimidos para mejorar la precisión y estabilidad.

Aprendizaje no supervisado

El aprendizaje no supervisado es una rama del aprendizaje automático en la que un sistema recibe únicamente datos de entrada sin ninguna etiqueta, salida deseada o retroalimentación del entorno. A diferencia del aprendizaje supervisado, donde se aprende una relación entre entradas y salidas, en el aprendizaje no supervisado el objetivo es descubrir estructuras o patrones ocultos en los datos.

Este tipo de aprendizaje se basa en la idea de que incluso sin retroalimentación externa, es posible extraer representaciones útiles de los datos que sirvan para tareas

como tomar decisiones, hacer predicciones, o comprimir información de manera eficiente. Dos ejemplos clásicos de aprendizaje no supervisado son el agrupamiento (clustering) y la reducción de dimensionalidad, como en el análisis de componentes principales (PCA). El enfoque puede estar motivado tanto desde principios bayesianos como desde la teoría de la información, y abarca modelos como mezclas de gaussianas, análisis de factores, modelos ocultos de Markov (HMM), y modelos gráficos probabilísticos. Estas herramientas permiten modelar y entender la estructura subyacente en grandes volúmenes de datos no etiquetados. (Ghahramani, 2004)

2.1.3 Tipos de Algoritmos

Machine learning generalmente desarrolla la solución de tres tipos de problemas: Clasificación, Regresión y Clustering. Dependiendo de la disponibilidad y categoría de los datos de entrenamiento, uno puede usar técnicas de aprendizaje supervisado, no supervisado o un aprendizaje mixto (Ray, 2019). Para ello, los principales algoritmos a utilizar son los siguientes:

Algoritmo de gradiente descendiente(LDA)

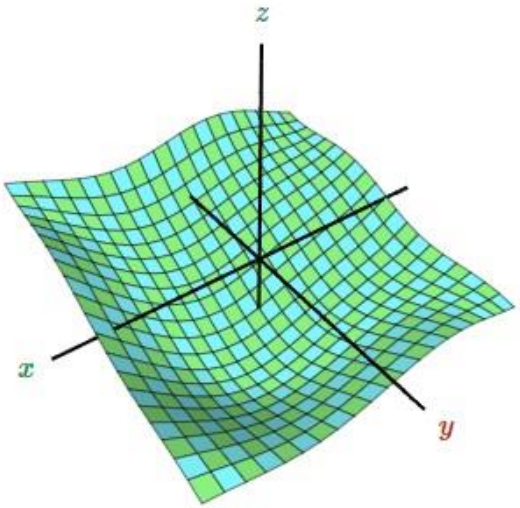
El algoritmo de gradiente descendiente es un algoritmo de optimización que se utiliza al entrenar un modelo de aprendizaje automático. Se basa en una función convexa y ajusta sus parámetros iterativamente para minimizar una función dada a su mínimo local. Se empieza definiendo los valores del parámetro inicial y, a partir de ahí, el algoritmo de descenso de gradiente utiliza el cálculo para ajustar iterativamente los valores de forma que minimicen la función de coste dada (IBM, s.f.).

La manera en que el descenso de gradiente encuentra los mínimos de una función se entiende más fácilmente si se visualiza en tres dimensiones. Se tiene la función $f(x, y)$, al graficarla, representa un paisaje montañoso en forma de un mapa de elevación. Se sabe que la pendiente en un punto indica la dirección en la que la elevación aumenta más rápidamente. Esta idea puede ayudar a visualizar cómo maximizar la función: empezamos

desde un punto aleatorio y damos pequeños pasos en la dirección de la pendiente para ir cuesta arriba. (Academy, s.f.).

Figura 14

Gráfica de una función f(x,y)



Nota: Khan Academy

Si, en cambio, queremos minimizar la función, tomamos la dirección opuesta a la del gradiente, es decir, seguimos la ruta del descenso más empinado. A este método se le conoce como descenso de gradiente. De manera formal, si se inicia en un punto x_0 y moverse una distancia positiva α en la dirección opuesta al gradiente, se obtiene un nuevo punto x_1 según la fórmula de la ecuación 1:

$$x_1 = x_0 - \propto \nabla f(x_0) \quad (1)$$

De forma general, la actualización del punto en cada iteración se expresa como la ecuación 2:

$$x_{n+1} = x_n - \propto \nabla f(x_n) \quad (2)$$

A partir de una estimación inicial x_0 el algoritmo mejora progresivamente hasta aproximarse a un mínimo local. (Academy, s.f.).

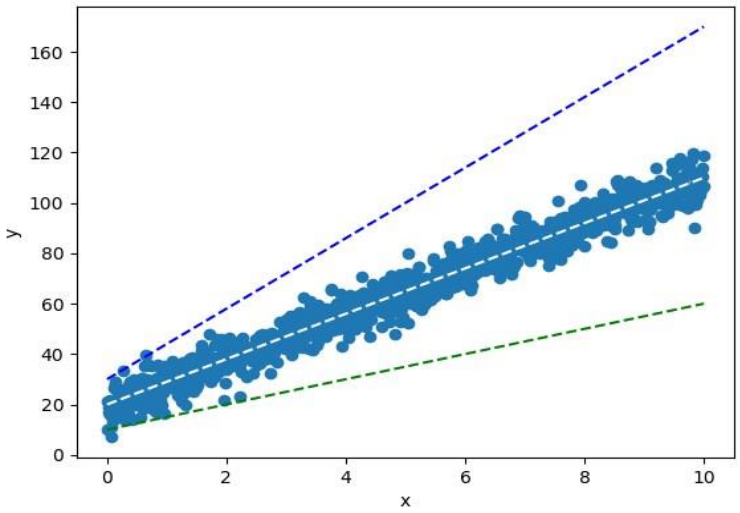
La principal ventaja del algoritmo es la eficiencia computacional, ya que produce una gradiente de error estable, sin embargo, algunas veces esta estabilidad del error no permite validar mejor el modelo que se puede lograr, además si la tasa de aprendizaje para el descenso de la gradiente es demasiado rápida, omitirá el verdadero mínimo para optimizar el tiempo. (Ray, 2019). Algunas aplicaciones son el entrenamiento de redes neuronales profundas (Deep Learning), optimización de modelos en visión por computadora y ajuste de modelos económicos y financieros.

Algoritmo de regresión linear(LoR)

La regresión lineal es un método utilizado para determinar la recta que mejor representa la relación existente entre dos variables. Considerando un conjunto de datos como el mostrado en una figura determinada, donde la variable independiente es x la variable dependiente es y, puede observarse que existe una relación lineal entre ambas. Esto implica que un aumento en x produce un incremento proporcional en y. Para ilustrar este concepto, se puede examinar una gráfica en la que se presentan tres líneas diferentes superpuestas sobre los datos originales. A partir de esta representación, resulta evidente que la línea blanca es la que describe con mayor precisión la relación entre las variables x y y. (codificandobits, 2021)

Figura 15

Tres posibles rectas resultado de la regresión lineal (codificandobits, 2021)



Para abordar el problema de la regresión lineal, es fundamental expresar de manera matemática los conceptos de “La ecuación de la recta” y “mejor ajuste”, los cuales se explican a continuación:

- La ecuación de la recta

  La relación entre una variable independiente x y una variable dependiente y puede modelarse mediante la ecuación 3:

  $$y = mx + b \quad (3)$$

  donde m representa la pendiente de la recta, que indica la inclinación, y b es el valor de intersección con el eje y (ordenada al origen).

  El objetivo principal de la regresión lineal consiste en determinar los valores óptimos de m y b que describan de manera precisa la relación entre x e y, ajustando lo mejor posible la recta a los datos observados.

  Para lograr esta representación óptima, es necesario establecer una métrica que permita evaluar la precisión del modelo. Esta métrica es conocida como función de costo, o función de pérdida dentro del contexto del aprendizaje automático. (codificandobits, 2021)

- Función de costo (o pérdida):

  La función de pérdida cuantifica la discrepancia entre los valores reales (y) y los valores predichos por el modelo (ŷ). Existen diversas formas de definir esta función, aunque la más comúnmente utilizada es el Error Cuadrático Medio (ECM), el cual se expresa como la ecuación 4:

  $$ECM = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \quad (4)$$

donde:

- *N* es el número total de observaciones,
- y_i *e* $\hat{y}_i$ representa el valor real de la *i*-ésima observación,
- $(y_i - \hat{y}_i)$ es el valor predicho por el modelo para esa misma observación.

La resta al cuadrado asegura que todas las diferencias sean positivas, evitando que errores positivos y negativos se cancelen entre sí. En este sentido, el ECM proporciona una medida del error promedio cometido al ajustar el modelo a los datos.

Una vez definida la función de pérdida, se puede entender el "mejor ajuste" como aquel que minimiza el ECM. Es decir, la recta óptima es aquella cuya predicción de los valores *y* a partir de x genera el menor error promedio.

Dado que cada predicción se calcula como la ecuación 5:

$$y_i = wx_i + b \quad (5)$$

al sustituir en la ecuación del ECM, el objetivo se convierte en encontrar los valores de *w* y *b* que reduzcan al mínimo dicha función de pérdida. (codificandobits, 2021)

La regresión lineal tiene la ventaja de que es fácil evitar el overfitting (entrenamiento con demasiados datos) por regularización, es una buena opción si se conoce que la relación entre las covariables y la variable respuesta es lineal. Sin embargo, no brinda un buen ajuste cuando se trata de relaciones no lineales por lo que los problemas complejos del mundo real son simplificados en gran medida dando un alto RSS. (Ray, 2019)

Regresión Multivariable (MLR)

La regresión multivariable es una extensión de la regresión lineal tradicional que permite modelar simultáneamente la relación entre múltiples variables dependientes y un

conjunto de variables independientes. Este enfoque es ampliamente utilizado en análisis estadístico, aprendizaje automático, procesamiento de señales e inteligencia artificial, debido a su capacidad para captar patrones complejos y relaciones interdependientes entre variables múltiples

Una de las principales limitaciones de la regresión multivariable clásica (MLR, por sus siglas en inglés) es su dependencia de un tamaño de muestra suficientemente grande. Cuando la cantidad de muestras en mucho menor a la dimensión de las características, se produce el famoso problema del submuestreo, que impide que se pueda estimar de manera correcta los coeficientes del modelo. Una opción para solucionar este problema, es la regresión por componentes principales (PCR), que reduce la dimensionalidad de los datos. No obstante, si no se selecciona correctamente los componentes principales, puede conducir a errores de sobreajuste o perdida de información relevante para el modelo.

Con la finalidad de resolver este dilema, Su et al. (2012) propone la técnica de regresión multilíneal multivariable (MMR), un modelo que permita mantener la estructura multidimensional de los datos si necesidad de vectorizarlos. Es decir, que en lugar de mostrar las variables independientes como vectores, las mantiene en su forma de matriz o tensor, esto permite que:

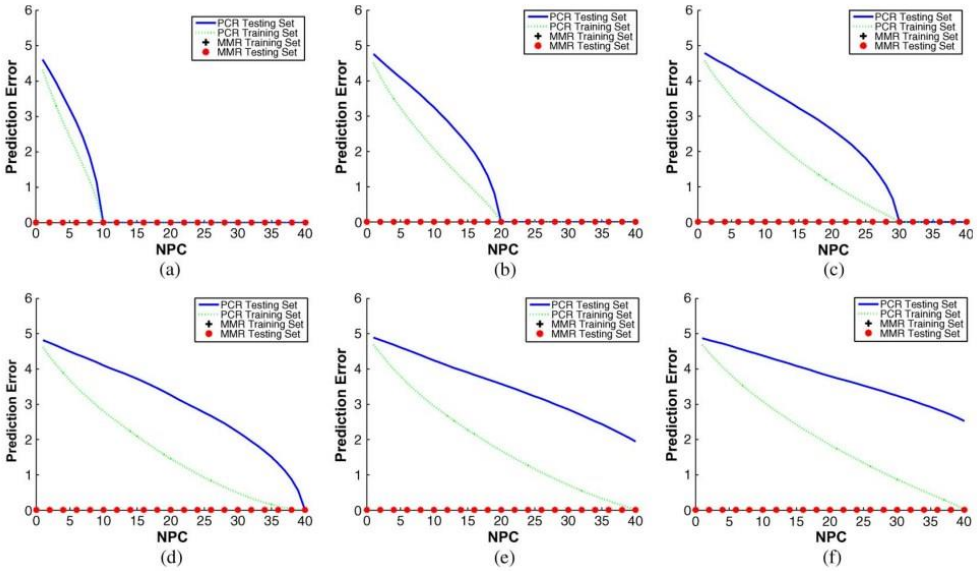
- Preserva la información estructural de los datos (por ejemplo, en imágenes o señales multicanal),
- Reduce la dimensionalidad efectiva del problema al operar con matrices de baja dimensión,
- Y alivia el problema de submuestreo, mejorando la eficiencia computacional y la precisión del modelo.

El modelo MMR es capaz de transformar el problema de regresión en uno de optimización bilineal, en el que se estiman dos matrices de proyección (nodo-1, nodo-2) en lugar de una única matriz de coeficientes. El algoritmo utiliza una técnica iterativa de

protecciones alternes con la finalidad de alcanzar la convergencia hacia una solución que minimiza el error de la predicción. Con esto se demuestra que es eficaz para tareas de visión por computadora, por ejemplo, ajuste de modelos de apariencia activa, donde se necesita modelar relaciones entre imágenes de alta definición con parámetros como forma y textura.. (Ya, Xinbo, & Xuelong, 2012)

Figura 16

Graficas de regresión lineal con diferentes coeficientes. (Ya, Xinbo, & Xuelong, 2012)



Uno de los principales méritos para usar la regresión multivariable es su capacidad para brindar una visión profunda de la relación entre las variables independientes y dependientes, también, es capaz de brindar un modelo complejo y realista. Sin embargo, tiene deficiencias como que se necesita un alto nivel de conocimiento en estadística y modelado, por lo que se vuelve complicado realizar un análisis correcto de los resultados del modelo estadístico. Además, la muestra con la que se trabaja debe ser grande para poder obtener un nivel de confianza alto en el resultado. (Ray, 2019)

Regresión Logística (NB)

Sperandei (2014) propone en su trabajo la utilidad y los fundamentos detrás del modelo de análisis por regresión lineal y su potencial como herramienta estadisica para

poder predecir la probaibilidad de ocurrencia de un evento dicotómico con gran cantidad de variables explicativas. A diferencia de la regresión lineal multiple, la regresión logística solo se aplica cuando la variable dependiente es binomial, permitiendo calcular de monios ajustadas, con el punto de que se muestren la asociación de variables independientes y la variable dependiente.

El autor describe que mediante un estudio sobre los tratamientos para endocarditis por Staphylococcus Aureus, se mostró que el análisis variado también puede inducir a errores por confusión, principalmente en variables como la edad. Teniendo en cuenta eso, la regresión logística puede controlar variables simultaneas y mejorar la validez interna del análisis, también, debido a que el modelo logístico se construye sobre la transformación logística de la probabilidad del evento. Sin embargo, el procedimiento del modelado no debe automatizarse, sino trabajarse de acuerdo al conocimiento teórico del fenómeno que se desea estudiar, así evitando la inclusión de variables ilegitimas o incluso la omisión de efectos relevantes por falta de conocimiento estadístico. . (Sperande, 2014)

En conclusión, la regresión logística constituye una herramienta robusta para el análisis multivariable en investigaciones epidemiológicas, siempre que se aplique con criterio técnico y sustentación teórica correcta. La regresión logística tiene las siguientes ventajas: Mantiene una simplicidad en la implementación del algoritmo, eficiencia computacional y eficacia dentro de los parámetros para entrenar. Y por último no se ve afectada por pequeños ruidos en la data. Entre sus principales aplicaciones esta la utilización de sus algoritmos para la detección de enfermedades en la medicina o predecir la probabilidad de falla de un determinado proceso o sistema. Sin embargo, las desventajas que se encuentra son la incapacidad de poder resolver problemas no lineales. (Ray, 2019)

Árbol de Decisión (CART)

Los árboles de decisión constituyen una de las metodologías más utilizadas en la clasificación supervisada dentro del aprendizaje automático. Tijo y Abdulazeez (2021)

presentan un análisis exhaustivo sobre el uso de algoritmos de árboles de decisión en diferentes escenarios, destacando su aplicabilidad y precisión frente a otras técnicas de clasificación.

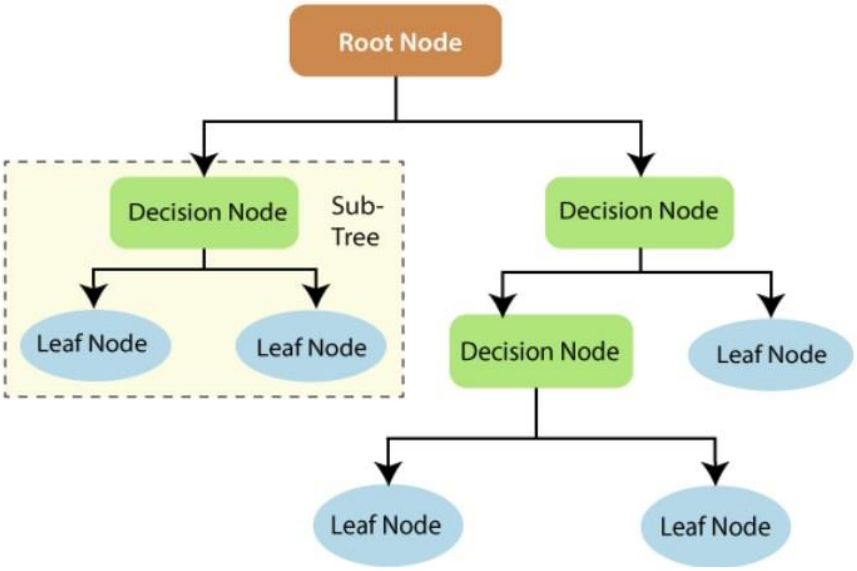
En el trabajo de investigación también explica que los árboles de decisión son modelos jerárquicos que dividen un conjunto de datos a través de secuencias lógicas hasta alcanzar decisiones binarias en los nodos. Su estructura permite una interpretación sencilla, haciendo que sean útiles, no solo en informática, sino también en áreas como medicina, reconocimiento de texto e imágenes, detección de intrusiones y diagnóstico clínico. Se revisan distintos algoritmos representativos como ID3, C4.5, CART, CHAID, QUEST y otros, criterios de partición (como entropía, ganancia de información y el índice Gini), y mecanismos de poda (pre-pruning y post-pruning). También se discuten métricas fundamentales como entropía e información ganada, empleadas para evaluar la calidad de las particiones en cada nodo del árbol. (Jijo & Adnan Mohsin, 2021)

El estudio incluye una revisión de informes de investigaciones recientes que han empleado árboles de decisión en problemas de clasificación como reconocimiento de escritura a mano, diagnóstico de cáncer de mama, predicción de diabetes, clasificación de genes y segmentación de imágenes médicas. Los resultados mostraron que los árboles de decisión pueden alcanzar niveles de precisión muy altos (hasta un 99.93%) dependiendo del conjunto de datos y la técnica de optimización utilizada. Además, se identifican las principales ventajas de los árboles de decisión como la facilidad de interpretación, capacidad de manejar datos tanto categóricos como numéricos, y una eficiencia computacional alta. Sin embargo, también se reconocen sus limitaciones, como una tendencia al sobreajuste, sensibilidad al ruido, y complejidad creciente con grandes volúmenes de datos. En conclusión, el uso de árboles de decisión en tareas de clasificación demuestra ser una opción altamente efectiva y robusta para el trabajo actual, particularmente cuando se busca combinar precisión, interpretabilidad y capacidad de generalización. Su integración en modelos híbridos, ensamblados y su optimización con

técnicas de reducción de dimensionalidad y cambio de parámetros pueden dar una luz en el trabajo actual. (Jijo & Adnan Mohsin, 2021)

Figura 17

Esquema de Árbol de decisión. (Jijo & Adnan Mohsin, 2021)



Support Vector Machine (SVM)

Las Máquinas de Vectores de Soporte (SVM, por sus siglas en inglés) representan una forma innovadora dentro del aprendizaje automático, el cual está plasmado en la teoría estadística del aprendizaje. Este método se caracteriza por su eficacia en ejercicios con muestras pequeñas, datos de alta dimensionalidad y relaciones no lineales (Zhang, 2023).

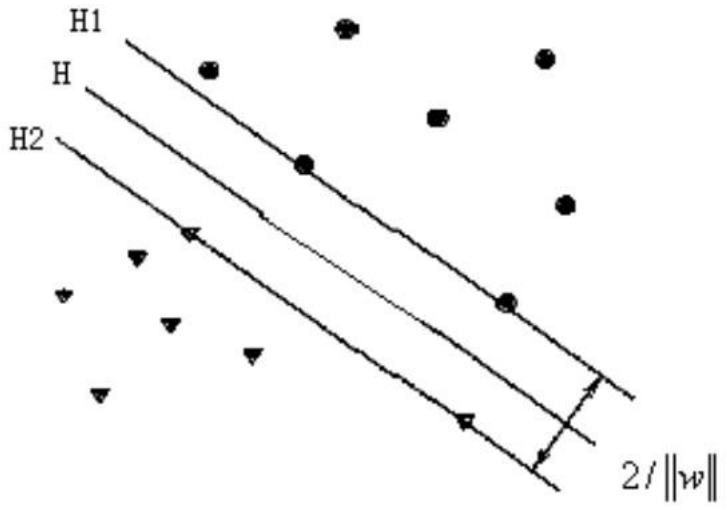
A diferencia de técnicas tradicionales que se basan en la minimización del riesgo empírico, las Máquinas de Vectores de Soporte aplican el principio de minimización del riesgo estructural, lo que les otorga una mayor capacidad de generalización durante su aplicación. La finalidad de un SVM es encontrar un hiperplano optimo que pueda separar 2 clases distintas, maximizando el margen entre los datos más cercanos de cada clase, este margen también e s llamado como vector de soporte. El artículo de Zhang (2023) destaca la formulación binaria y las extensiones multicategórica del algoritmo.

En los problemas de clasificación multiclase, existen dos estrategias comunes: la primera es descomponer el problema en múltiples subproblemas binarios ("one-vs-one" o "one-vs-all"), y la segunda es construir directamente un modelo multicategoríco, lo cual es donde actualmente se da esfuerzos para investigar.

En la vida real, el algoritmo SVM ha mostrado ser eficaz en contextos como reconocimientos de patrones, diagnostico médicos, minería de datos, clasificación de textos, entre otras cosas, principalmente gracias a su robustez teórico y un rendimiento empírico eficiente.

Figura 18

Ejemplo hiperplano óptimo separando dos clases distintas. (Zhang, 2023).



K Nearest neighbour (KNN)

El algoritmo k-Nearest Neighbor (k-NN) es modelo de aprendizaje supervisado que se usa mucho en problemas de clasificación y estimación. El concepto principal es que funciona gracias al principio de aprendizaje basado en instancias, es decir, utiliza los datos históricos (conjunto de entrenamiento) para aprender y poder clasificar nuevos registros en función de su similitud con los ejemplos previamente clasificados dentro de la base de datos. (T. Larose & D. Larose., 2014)

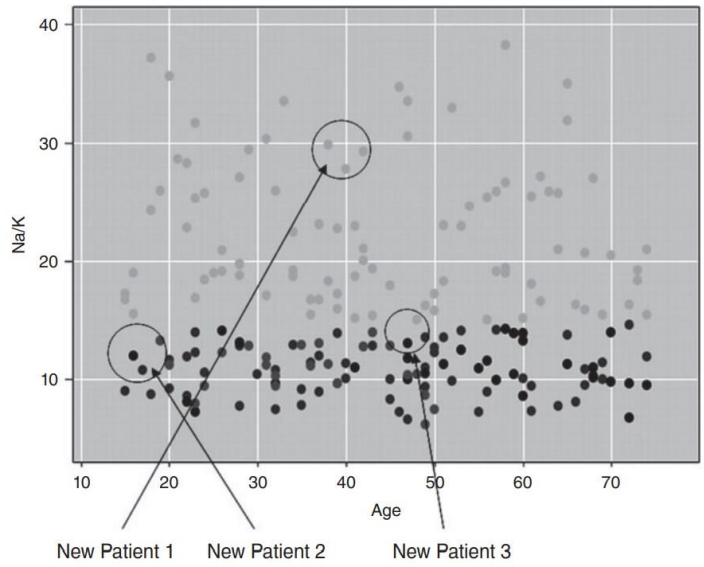
El proceso de clasificación mediante el modelo K-NN se basa en poder hallar los k registros mas cercanos, o también llamados vecinos, a un nuevo caso, y esto se logra midiendo la distancia euclidiana entre ellos. La clase más común entre estos vecinos se asignaal nuevo registro. El método de por si es muy intuitivo y no necesita de una fase de entramiento explicida,sin embargo, si necesita una base de datos que se encuentre representativa y balanceada, para poder detectar las clases minoritarias.

Dentro del algoritmo, el valor de k tiene un papel importante y tiene que tomarse muy en cuenta ya que si es un valor pequeño puede causar sobreajuste y sensibilidad al ruido, por otro lado, si es muy grande pueden diluir patrones relevantes. Para optimizar este parámetro, se recomienda el uso de validación cruzada. Los autores también abordan aspectos esenciales como la selección de funciones de distancia, la normalización de atributos mediante técnicas como min-max o Z-score, y el uso de funciones de combinación para la toma de decisiones. También, se analizan enfoques de votación simple de tipo no ponderada y ponderada (basada en la inversa de la distancia), esta última tiene la ventaja de reducir empates y mejorar la precisión de la clasificación. Además de la clasificación, el algoritmo k-NN puede emplearse en proyectos de predicción de variables continuas mediante promedios localmente ponderados. En estos casos, se estima el valor de la variable objetivo como una media ponderada de los valores correspondientes de los vecinos más cercanos. (T. Larose & D. Larose., 2014)

Finalmente, el trabajo de investigación enfatiza la importancia de seleccionar atributos relevantes y cuantificar su importancia mediante técnicas como el estiramiento de ejes, que consiste en asignar pesos diferenciales a los atributos con mayor relevancia para la clasificación.

Figura 19

Ejemplo de reconocimiento de patrones con k-NEAREST NEIGHBOR. (T. Larose & D. Larose., 2014)



2.1.4 Transformación de Datos

Estandarización

La estandarización de datos consiste en transformar la información para que adopte un formato y una estructura uniformes, lo que permite su análisis e integración sin inconvenientes entre distintos sistemas. Este proceso convierte datos provenientes de diversas fuentes en una forma homogénea, asegurando una representación consistente de los mismos elementos, sin importar su procedencia. Su propósito es eliminar inconsistencias y discrepancias, y facilitar así la comparación, combinación y evaluación de los datos. Un ejemplo es una empresa que recopila información de clientes a través de diferentes medios como formularios web, llamadas telefónicas o visitas físicas, esta empresa podría aplicar la estandarización para unificar la manera en que se registran los nombres, asegurándose de que todos sigan el mismo formato, como "John Doe" en lugar de "Doe, John". (Brownlee, Wright, He, & Timothy, 2020)

Matemáticamente, Existen cuatro enfoques principales para la estandarización de datos: puntuación Z, normalización mínimo-máximo, escalado por valor absoluto máximo y estandarización robusta.

a. Puntuación Z: Consiste en calcular cuántas desviaciones estándar se encuentra un valor con respecto a la media del conjunto de datos; también se le conoce como puntuación estándar. Este método resulta útil para valorar la relevancia de un dato dentro de la distribución general y espera una distribución normal. Por lo tanto, no se recomienda este método si la distribución de los datos está muy sesgada.

   La ecuación es $x' = \frac{x-\bar{x}}{\sigma_x}$ (6) donde:

   - x' = valor estandarizado
   - x = valor original
   - $\bar{x}$ = valor medio (promedio)
   - σ_x = desviación estándar.

b. Mínimo-máximo: El método de estandarización de mínimo-máximo transforma los datos manteniendo las proporciones originales entre los valores, ajustándolos a una escala definida por los valores mínimo y máximo que defina el usuario. Un ejemploo sería un agente inmobiliario que necesite llevar a una misma escala características como el número de habitaciones o la antigüedad de las propiedades en años antes de incorporarlas en modelos predictivos, como los algoritmos de clasificación y regresión basados en bosques aleatorios. Sin embargo, esta técnica puede verse afectada por la presencia de valores extremos o atípicos, lo que podría distorsionar los resultados.

   Su ecuación es $x' = a + \frac{(x-\min(x))(b-a)}{\max(x)-\min(x)}$ (7) donde:

   - x' = Valor estandarizado
   - x = Valor original

- min(x) = mínimo de los datos
- max(x) =máximo de los datos
- a = mínimo especificado por el usuario
- b = mínimo especificado por el usuario

c. Máximo absoluto: Este método es especialmente útil cuando se trabaja con datos que tienen un límite superior claro y constante, y se desea evaluar los valores en función de ese límite. Por ejemplo, si se analiza la cantidad de votos en una región, se puede considerar como tope el número total de ciudadanos con derecho a voto. El condado con la mayor participación se toma como referencia (valor máximo), y los demás se comparan con respecto a este. Sin embargo, hay que tener en cuenta que los valores resultantes estarán en una escala de -1 a 1. Los valores positivos más altos indican cercanía al máximo absoluto, mientras que los negativos más altos reflejan cercanía al mínimo dentro del rango negativo.

Su ecuación es: $x' = \frac{x}{\max(|x)|}$ (8)

- x' = Valor estandarizado
- x = Valor normal
- $\max(|x)|$ = máximo de los valores absolutos de los datos.

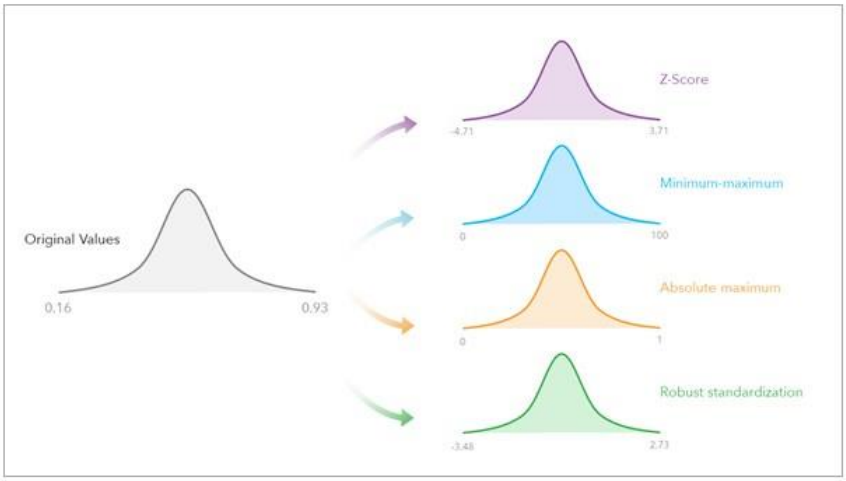
d. Estandarización sólida: Este método normaliza los valores de los campos seleccionados mediante una versión robusta de la puntuación z, que emplea la mediana y el rango intercuartílico (IQR) en lugar de la media y la desviación estándar. Al basarse en estas medidas más resistentes, resulta particularmente útil para minimizar el impacto de los valores atípicos dentro de la distribución.

Su ecuación es: $x = \frac{x - median(x)}{IQR(x)}$ (9)

- x' = Valor estandarizado
- x = Valor normal
- IQR(x) = rango entre cuartiles de los datos.

Figura 20

Tipos de estandarización (ESRI, 2021)



Los componentes clave de la estandarización de los datos incluyen la transformación de los datos, la limpieza de los datos y la asignación de datos. La transformación de datos convierte los datos en un formato estándar, como normalizar los campos de texto o estandarizar los formatos de fecha. La limpieza de datos identifica y corrige errores, incoherencias y duplicados. El mapeo de datos establece relaciones entre diferentes fuentes de datos para permitir una integración perfecta. Estas prácticas secuenciales ayudan a las organizaciones a mejorar la calidad de los datos, facilitar la integración y mejorar sus procesos de toma de decisiones. Ayudan a crear una visión unificada de los datos que permite tomar decisiones fundamentadas y basadas en los datos y colaborar entre departamentos y sistemas. (ESRI, 2021)

2.1.4.1 Normalización

La normalización de datos es un proceso de escalado numérico dentro de una matriz de datos, se utiliza mayormente cuando la heterogeneidad de los números dificulta el análisis estadístico estándar. Se considera una etapa de preprocesamiento de datos, se aplica antes de iniciar cualquier proceso de relacionado al análisis estadístico. Generalmente, las ventajas de la normalización de datos son: (Balemir & Ramesh , 2011).

- Proporcionar un rango más significativo de números escalados.
- Reorganizar la matriz de datos en una distribución más regular.
- Mejorar la exactitud de los cálculos posteriores.
- Aumentar la importancia de los números más descriptivos en un conjunto de datos con distribución no normal.

Existen diversas técnicas de normalización de datos, su elección depende del tipo y tamaño de los datos. Algunas de las técnicas estudiadas son:

- Corrección de la media: Elimina los valores medios de las variables de cada vector, resultando en datos con media cero. Sin embargo, puede distorsionar los valores originales y exagerar las variaciones de ciertos patrones.
- Valor máximo único: Escala todos los datos no normalizados entre 0 y un valor máximo único, siendo el máximo general el que se usa para dividir cada componente de la matriz.
- Valor máximo de cada vector: Divide los puntos de datos de cada conjunto por el valor máximo de ese conjunto, escalando todas las funciones a un máximo de 1.
- Desviación estándar de cada vector: Normaliza cada punto de datos por su desviación estándar, lo que puede mejorar la dispersión de los datos, especialmente para funciones asimétricas.
- Normalización logarítmica: Una técnica no lineal que reduce la varianza y es adecuada para datos no lineales, donde cada punto de datos se divide por la media de esa función y luego se aplica el logaritmo en base 2.
- Normalización de masa de probabilidad total unitaria (UTPM): Divide cada elemento de un vector por la suma de las variables de ese vector y lo multiplica por la media. Esta técnica fue considerada la más adecuada en un estudio específico para

funciones de respuesta ILD (Diferencias de Nivel Interaural) porque preserva la forma de las funciones, mantiene las posiciones de los puntos de corte y las pendientes, y las transiciones entre tipos de funciones no se distorsionan.

- Estandarización de datos: Se logra dividiendo los datos que fuero corregidos previamente por la media de su desviación estándar. Un plus es que los datos se pueden expresar en unidades comparables y con varianzas estandarizadas de 1 y covarianzas entre -1 y +1.

En resumen, la normalización es un proceso fundamental para reducir algunos errores sistemáticos y maximizar la varianza entre los datos, con la finalidad de poder realizar un análisis estadístico optimo. La selección del método correcto es importante y debe ser evaluada cuidadosamente, mediante comparaciones visuales con los datos en bruto y también calculando la suma de cuadrados de diferencias y coeficientes de correlación . (Al-Shalabi & Shaaban, 2006).

2.1.5 Selección de características

La selección de características es un conjunto de técnicas que buscan identificar un subconjunto de características de entrada que sean más importantes para la variable de salida que se quiere predecir. Los objetivos de la selección de características son múltiples, siendo los más importantes:

- Evitar el sobreajuste y mejorar el rendimiento del modelo, es decir, el rendimiento de predicción en el caso de clasificación supervisada y una mejor detección de conglomerados en el caso de conglomerados
- Proporcionar modelos más rentables para el proyecto.
- Obtener una visión más profunda de los procesos internos que generaron los datos.

Sin embargo, las ventajas de las técnicas de selección de características tienen un alcance máximo determinado, ya que la búsqueda de un subconjunto relevante de

características introduce una capa adicional de complejidad en la tarea de modelado. (Saeys, Iñaki, & Larrañaga, 2007)

2.1.6 Funcionamiento de un motor Diésel de camión minero

El funcionamiento de un motor diésel se basa en el ciclo de cuatro tiempos (admisión, compresión, combustión/expansión y escape), aunque existen motores de dos tiempos también utilizados en aplicaciones de gran potencia.

1. Admisión (Primer Tiempo):

    - Válvulas de Admisión Abiertas: El pistón desciende desde el Punto Muerto Superior (PMS) hasta el Punto Muerto Inferior (PMI).
    - Entrada de Aire Puro: A medida que el pistón baja, este crea un vacío en el cilindro, succionando aire fresco y sin mezclar (a diferencia de un motor de gasolina que aspira una mezcla aire y combustible) a través de las válvulas de admisión. En camiones mineros, este aire es a menudo sobrealimentado (mediante turbocompresores) para incrementar la densidad del aire y, por ende, la potencia.

2. Compresión (Segundo Tiempo):

    - Válvulas Cerradas: Las válvulas de admisión y escape se cierran.
    - Compresión Extrema del Aire: El pistón sube desde el PMI hasta el PMS, comprime el aire atrapado en el cilindro a presiones extremadamente altas (típicamente entre 30 y 55 bares).
    - Aumento de Temperatura: Esta compresión adiabática eleva drásticamente la temperatura del aire a valores entre 700°C y 900°C, superando el punto de autoignición del diésel. Este es el principio clave de la ignición por compresión.

3. Combustión y Expansión (Tercer Tiempo - Tiempo de Potencia):

   - Inyección de Combustible: Justo antes de que el pistón alcance el PMS, un inyector de alta precisión pulveriza diésel finamente atomizado directamente en la cámara de combustión.

   - Autoignición y Explosión: El combustible, cuando entra en contacto con el aire caliente y comprimido, esto genera que se auto encienda de inmediato, generando una combustión rápida y potente. Esta explosión eleva la presión y temperatura dentro del cilindro.

   - Empuje del Pistón: La alta presión de los gasos que se encuentra en expansión empuja violentamente el pistón hacia el PMI, transmitiendo la energía a la biela y esta asu vez al cigüeñal, transformando el movimiento lineal en rotatorio.

4. Escape (Cuarto Tiempo):

   - Válvulas de Escape Abiertas: Cuando el pistón ha alcanzado el PMI y completado su carrera de potencia, las válvulas de escapen logran abrirse.

   - Expulsión de Gases Quemados: El pistón sube nuevamente desde el PMI hasta el PMS, moviendo los gases de escape fuera del cilindro a través del múltiple de escape. Estos gases a menudo son aprovechados por el turbocompresor para impulsar su turbina.

<u>Características de motores diésel de camiones mineros:</u>

- Turbocompresión y Post-enfriamiento (Intercooler): Esto es esencial para maximizar la densidad del aire de admisión. Los turbocompresores aprovechan la energía de los gases de escape para forzar más aire en los cilindros, aumentando la potencia y la eficiencia. El intercooler enfría este aire comprimido para aumentar aún más su densidad.

- Sistemas de Inyección de Combustible de Alta Presión: Los camiones mineros utilizan sistemas avanzados como Common Rail o inyectores unitarios electrónicos (EUI/HEUI) que operan a presiones extremadamente altas (hasta 2500 bar o más). Esto asegura una atomización ultrafina del combustible para una combustión más completa, eficiente y con menores emisiones.

- Robustez y Durabilidad: Construidos con materiales resistentes y tolerancias precisas para soportar las vibraciones, el polvo, las temperaturas extremas y las cargas de trabajo continuas y pesadas inherentes a la minería.

- Alto Par Motor a Bajas Revoluciones: Crucial para mover cargas masivas cuesta arriba y en terrenos irregulares. Los motores diésel naturalmente producen un par elevado a bajas RPM, lo que permite a los camiones mineros operar de manera efectiva sin necesidad de revoluciones excesivas.

- Sistemas de Enfriamiento Avanzados: Debido a las altas cargas térmicas, los motores cuentan con unos sistemas de enfriamiento sobredimensionados para mantener temperaturas correctas de operación.

- Sistemas de Filtración de Aire y Combustible de Alto Rendimiento: debido a la naturaleza de exceso de polvo y exigente del entorno minero, la filtración es crítica para proteger el motor de contaminantes abrasivos.

- Sistemas de Control Electrónico (ECM): Son unidades de control electrónico que gestionan y optimizan continuamente todos los parámetros del motor (inyección, turbo, EGR, etc.) para maximizar el rendimiento, la eficiencia del combustible y reducir las emisiones.

En resumen, el motor diésel de un camión minero es un sistema altamente ingenioso que aprovecha la autoignición por compresión para convertir eficientemente la energía del combustible en la fuerza bruta necesaria para las operaciones mineras más exigentes. Su

diseño robusto y sus tecnologías avanzadas lo hacen indispensable en esta industria. (Finning, s.f).

2.1.7 Estrategia de mantenimiento del Motor

El mantenimiento de motores diésel es una actividad crítica para garantizar la disponibilidad, fiabilidad y vida útil de los equipos en diversas industrias, especialmente en sectores como la minería. Cuando se tiene una estrategia de mantenimiento estructurada se puede minimizar los tiempos muertos no planificados, reducir costos operativos y evitar fallas catastróficas. Para lograr esta reducción de fallas, se implementa una combinación de estrategias de mantenimiento. (Larico, 2021)

- Mantenimiento Preventivo (Basado en Tiempo o Uso): En el contexto minero, el mantenimiento preventivo se ordena en función de las horas de operación del motor (horómetro) y no del kilometraje debido a la variación en las condiciones de carga y terreno. Su objetivo es anticiparse al desgaste normal y reemplazar componentes antes de que alcancen su punto de falla. Algunos aspectos claves en minería son:
    - Intervalos de Servicio Estrictos: Los fabricantes de motores para minería (ej., Cummins, Caterpillar, Komatsu) especifican intervalos de mantenimiento muy detallados para cambios de aceite, filtros (aceite, combustible, aire, hidráulicos) y refrigerante, adaptados a las condiciones de operación severas. Estos intervalos son a menudo más cortos que en aplicaciones de transporte por carretera
    - Inspecciones Pre-operacionales y Periódicas: Los operadores y técnicos realizan inspecciones diarias, semanales y mensuales para detectar fugas, ruidos anómalos, vibraciones, estado de mangueras, correas, sistemas de enfriamiento y lubricación. La detección temprana de anomalías visuales es crítica.

- Limpieza y Control de Contaminantes: Dada la presencia constante de polvo y partículas en el ambiente minero, la limpieza regular de los filtros de aire, radiadores y el motor en general es vital para prevenir el sobrecalentamiento y la entrada de abrasivos al sistema. Los sistemas de filtración de aire son robustos y a menudo de varias etapas.

- Ajustes y Calibraciones Programadas: La calibración de inyectores, el ajuste de válvulas, la verificación de los sistemas de inyección y turbocompresión se realizan en intervalos definidos para mantener la eficiencia de la combustión y el rendimiento óptimo del motor.

Algunas ventajas del mantenimiento preventivo son: alta fiabilidad operacional, preservación de activos y control de costo. Sin embargo, también cuenta con desventajas como el potencial sobremantenimiento o el no cubrir fallas imprevistas. (Naranjo, s.f.)

Figura 21

Mantenimiento preventivo (Naranjo,s.f)

- Mantenimiento Predictivo (Basado en la Condición): El mantenimiento predictivo es la estrategia más importante para la reducción de fallas en motores diésel mineros.

Utiliza tecnologías avanzadas y análisis de datos para monitorear la salud del motor en tiempo real o casi real, permitiendo prever fallas y programar el mantenimiento de manera óptima, minimizando interrupciones. Algunos aspectos claves en minería son:

- Análisis de Aceite (Análisis de Lubricantes): Es una de las herramientas más valiosas. Se toman muestras periódicas del aceite del motor para analizar:
    - Metales de Desgaste: Presencia de hierro, cromo, cobre, plomo, aluminio, etc., que indican el desgaste de componentes específicos por ejemplo cilindros, rodamientos, cojinetes en mal estado.
    - Contaminantes: mide niveles de sílice (polvo), agua, combustible, hollín y anticongelante, que pueden señalar problemas en la filtración, sellos o combustión.
    - Propiedades del Aceite: Mide la viscosidad, número total de base (TBN), número total de ácido (TAN) para evaluar la degradación del lubricante y su capacidad para proteger el motor. (Zambrano & Perez, 2021).
- Monitoreo de Parámetros Operativos (Telemetría / IoT): Los camiones mineros modernos están equipados con una gran cantidad de sensores que transmiten datos en tiempo real a sistemas de gestión de flotas. (Naranjo, s.f.). Se monitorean parámetros como:
    - Temperaturas (aceite, refrigerante, gases de escape).
    - Presiones (aceite, turbo, combustible).
    - Consumo de combustible.
    - RPM del motor y carga.
    - Códigos de falla del ECM (Engine Control Module

- Análisis de Vibraciones: Aunque es más común en equipos rotativos como bombas o ventiladores, es posible aplicar a componentes específicos del motor como turbocompresores o alternadores para detectar desequilibrios o fallas en rodamientos.

Algunas ventajas del mantenimiento predictivo son la reducción drástica de fallas catastrófica que optimizan la vida útil de componentes, minimización del tiempo de inactividad no programado y incremento. Las desventajas son la inversión inicial, la necesidad de personal altamente especializado y la gestión de grandes volúmenes de datos (Big Data).

- Mantenimiento Correctivo (Reactivo): Aunque el objetivo es minimizarlo, el mantenimiento correctivo es una realidad inevitable. En minería, una falla inesperada de un camión puede paralizar una sección de la operación, generando pérdidas significativas. (Valencia, s.f.). Algunos impactos pueden ser:
    - Alto Costos de Reparación: Las fallas imprevistas a menudo resultan en daños secundarios severos, requiriendo reparaciones más extensas y costosas.
    - Pérdida de Producción: El tiempo que el equipo está inactivo impacta directamente en la capacidad de extracción y transporte de mineral.
    - Riesgos de Seguridad: Algunas fallas pueden comprometer la seguridad del personal y del equipo.

2.1.8 Principales Kpis de Mantenimiento.

- Disponibilidad

La disponibilidad es la medida de tiempo que está disponible una máquina para que la use el departamento de producción quiere decir, que no esté desactivada para su mantenimiento. En otras palabras, puede describirse como el período de tiempo en que el activo debe estar en funcionamiento. Para calcular la

disponibilidad, hay que comparar las horas en las que el activo estuvo disponible con las horas de trabajo planificadas. Esta es la fórmula para calcular la disponibilidad: (Caterpillar, Caterpillar Administración de equipos de minería, 2019)

$$\boldsymbol{Disponibilidad}\ (\%) = \frac{\boldsymbol{Total\ de\ Horas - Horas\ de\ inactividad}}{\boldsymbol{Total\ de\ Horas}} \qquad (1)$$

- Relación de mantenimiento

La relación de mantenimiento es una relación sin dimensión de las horas de trabajo de mantenimiento y reparaciones divididas por las horas operativas de la máquina. La relación de mantenimiento directa solo tiene en cuenta las horas de mano de obra de las órdenes de trabajo , es decir, el trabajo directo. Algunos ejemplos de las horas de mano de obra que no se incluyen en este cálculo son el taller de reparaciones, el Centro de reconstrucción de componentes, etc. La relación de mantenimiento "general" incluye todos los elementos de Relación de mantenimiento "cargada" más el personal, la supervisión y el tiempo de inactividad. (Caterpillar, Caterpillar Administración de equipos de minería, 2019)

$$\boldsymbol{Relación\ de\ mantenimiento} = \frac{\boldsymbol{Horas\ hombre\ de\ mantenimiento\ y\ reparación}}{\boldsymbol{Horas\ de\ funcionamiento}} \qquad (2)$$

- Tiempo promedio entre paradas (MBTS)

Es el tiempo de funcionamiento promedio entre todas las interrupciones de máquinas, sean planificadas o no. Esta es la frecuencia promedio de eventos de inactividad de los equipos, expresados en horas. El MTBS es una medida que combina los efectos de la fiabilidad inherente de la máquina con la eficacia de la organización de administración de mantenimiento para influir en los resultados a base de evitar que los equipos interrumpan su funcionamiento (Caterpillar, Caterpillar Administración de equipos de minería, 2019)

$$\boldsymbol{MTBS} = \frac{\boldsymbol{Horas\ de\ funcionamiento}}{\boldsymbol{Paradas\ totales}} \qquad (3)$$

- Mantenimiento Predictivo

Consiste en anticipar cuándo se puede presentar la falla en un equipo, y realizar acciones preventivas sin perjuicio a su funcionamiento normal. Estos controles se realizan de forma periódica, de acuerdo con el tipo de máquina, edad y condiciones de operación. El mantenimiento predictivo surge como respuesta a la necesidad de reducir los costos de los métodos tradicionales de mantenimiento, preventivo y correctivo, y parte del conocimiento del estado de los equipos. La dificultad de implantar este tipo de mantenimiento radica en la localización de la variable identificadora y en correlacionar niveles de aceptación o rechazo de dicha variable con estados reales de la máquina fácilmente medibles. Debe verse complementado por la utilización de técnicas estadísticas a través de la medición rigurosa de variables y tratamiento de dichas medidas. El mantenimiento predictivo basado en el análisis de aceite es un método que ayuda a determinar los períodos óptimos de sustitución del lubricante y las causas que estén originando su degradación y contaminación (Caterpillar, Caterpillar Administración de equipos de minería, 2019)

2.2 Marco Conceptual

2.2.1 Análisis de Aceite en Motores Diésel

Definición:

El análisis de aceite es una técnica de monitoreo de condición no invasiva y predictiva que se enfoca en la extracción periódica de una pequeña muestra de lubricante de un sistema en operación para su evaluación en un laboratorio especializado (Fitch, 2007). Su propósito principal es determinar la condición actual del aceite, identificar la presencia y tipo de contaminantes, y detectar partículas de desgaste metálicas que puedan indicar una degradación incipiente o avanzada de los componentes internos de la máquina (Bloch, & Geitner, 2006).

En el ámbito de los motores Diesel, los cuales operan bajo condiciones de alta carga, exposición a ambientes abrasivos y programas de trabajo continuo, el análisis de aceite se muestra como una herramienta crucial para el mantenimiento, ya que permite identificar con antelación anomalías en el funcionamiento y permite la prolongación de la vida útil de los componentes críticos, reduciendo los costos operativos y mejorando de manera sustancial la disponibilidad y confiabilidad de la flota (Mobley, 2012). La implementación sistemática de esta técnica contribuye directamente a la estrategia de mantenimiento predictivo, pasando de un enfoque reactivo o basado en el tiempo a uno basado en la condición real del activo.

Propiedades del aceite lubricante

El aceite lubricante en un motor diésel cumple funciones multifacéticas esenciales para su operación y durabilidad. Estas incluyen la reducción de la fricción y el desgaste entre superficies en movimiento, la disipación del calor generado por la combustión y la fricción, la suspensión y transporte de contaminantes como hollín y partículas de suciedad hacia el filtro, la protección contra la corrosión y la herrumbre, y el sellado de las holguras entre los componentes del motor, como los anillos del pistón y las camisas del cilindro. (Westbrook, 2013).

La eficacia de un aceite lubricante depende de sus propiedades físico-químicas inherentes y de la acción de los aditivos formulados para mejorar su rendimiento. Entre las propiedades más relevantes para el monitoreo se encuentran:

- Viscosidad: Es la resistencia del aceite a fluir y es crítica para la formación de una película lubricante adecuada. Cambios en la viscosidad pueden indicar dilución por combustible, contaminación por agua o degradación por oxidación. La viscosidad se mide típicamente a 40°C y 100°C para evaluar su comportamiento a diferentes temperaturas operativas (Fitch, 2007).

- Índice de Viscosidad (IV): Refleja la estabilidad de la viscosidad del aceite ante cambios de temperatura. Un IV alto indica que la viscosidad varía menos con la temperatura (Schofield, 2010).
- Número Básico Total (TBN): Representa la capacidad del aceite para neutralizar los ácidos formados durante la combustión, especialmente relevante en motores diésel debido a la presencia de azufre en el combustible. Un TBN bajo indica el agotamiento de los aditivos alcalinos y la proximidad al final de la vida útil del aceite.
- Punto de Inflamación (Flash Point): Es la temperatura más baja a la que los vapores del aceite se encienden momentáneamente en presencia de una llama. Una disminución significativa del punto de inflamación es un fuerte indicador de dilución por combustible. (Fitch, 2007).
- Aditivos: Son componentes químicos que son añadidos al aceite base para mejorar sus propiedades. Incluyen detergentes (limpieza), dispersantes (suspensión de hollín), anti-desgaste, antioxidantes para prevenir oxidación, e inhibidores de corrosión, entre otros (Westbrook, 2013). También, el análisis de ciertos elementos como por ejemplo P, Zn, Ca, Mg, B permite monitorear el estado de estos aditivos.

<u>Parámetros clave en el monitoreo de condiciones:</u>

- Análisis de Metales de Desgaste: Esta técnica cuantifica la concentración de partículas metálicas submicrométricas suspendidas en el aceite, proporcionando una huella digital del desgaste de los componentes internos. Los elementos comunes detectados incluyen hierro (Fe) de camisas de cilindro y engranajes; cromo (Cr) de anillos de pistón; aluminio (Al) de pistones y carcasas; cobre (Cu), plomo (Pb) y estaño (Sn) de cojinetes y bujes. El análisis de tendencias de estos elementos es crucial para identificar patrones de desgaste y predecir posibles fallas de componentes específicos (Fitch, 2007)

- Análisis de Contaminantes:
    - Silicio (Si): es un indicador primario de la entrada de polvo o suciedad al motor, generalmente a través de un sistema de filtración de aire comprometido. Las partículas de sílice son altamente abrasivas y son una de las principales causas de desgaste en motores diésel (Corporation, s.f.).
    - Agua (H_2O): La presencia de agua en el aceite puede deberse a condensación, fugas de refrigerante o ingreso externo. El agua reduce la capacidad lubricante del aceite, promueve la oxidación y puede causar corrosión y cavitación. Normalmente se puede cuantificar mediante el método Karl Fischer (Bloch, & Geitner, 2006)
    - Hollín (Soot): Son partículas de carbono resultantes de la combustión incompleta del diésel. Niveles elevados de hollín pueden aumentar la viscosidad del aceite, acelerar el desgaste abrasivo y causar la obstrucción de filtros. Se detecta comúnmente mediante espectroscopia infrarroja por transformada de Fourier (FTIR) (Bloch, & Geitner, 2006)
    - Combustible (Dilución por Combustible): La presencia de diésel no quemado en el aceite reduce significativamente su viscosidad, comprometiendo la capacidad de la película lubricante y aumentando el desgaste. Se detecta por la disminución del punto de inflamación o mediante cromatografía de gases (Fitch, 2007)
    - Glicol (Anticongelante): La detección de glicol, a menudo acompañada de sodio (Na) y potasio (K), es un signo inequívoco de fuga del sistema de refrigeración al cárter. El glicol degrada severamente el aceite y puede formar lodos perjudiciales (Corporation, s.f.)

<u>Interpretación de Resultados:</u>

La verdadera potencia del análisis de aceite esta en la interpretación experta de los resultados, no solo en la obtención de datos (Fitch, 2007). Esta interpretación se basa

principalmente en el análisis de tendencias, donde se monitorean los cambios en los parámetros a lo largo del tiempo con la finalidad de identificar desviaciones significativas de los valores normales o de línea base. Los valores de alerta y críticos se establecen en función del tipo de motor, las recomendaciones del fabricante, el historial operativo del equipo y los estándares de la industria (Mobley, 2012)

La correlación entre diferentes parámetros es vital para un diagnóstico preciso. Por ejemplo, un aumento simultáneo de hierro (Fe) y silicio (Si) pudiera indicar un desgaste abrasivo severo debido a la entrada de polvo; mientras que una disminución de la viscosidad junto con una caída del punto de inflamación sugiere dilución por combustible. La interpretación debe considerar las horas de operación del aceite y del motor, las condiciones ambientales, la calidad del combustible y cualquier evento de mantenimiento reciente que pueda influir en los resultados. La detección temprana de estas desviaciones permite programar intervenciones correctivas antes de que se produzca una falla catastrófica, optimizando la gestión de la vida útil de los componentes y la operación general de los camiones mineros.

Figura 22

Toma de aceite de motor

Nota: CAT

2.2.2 Sistema de información vital del motor (VIMS):

Para poder lograr una predicción de fallas de motores, es necesario tener un sistema integrado de monitoreo a bordo. En ese escenario, el Sistema de Información Vital del Motor (VIMS, por sus siglas en inglés, Vital Information Management System) de Caterpillar es una herramienta fundamental. El VINS es un sistema electrónico de monitoreo y diagnóstico de equipos, recopila analiza y comunica datos operativos y de condición de tiempo real, tales como la condición del motor, transmisión, sistemas hidráulicos, entre otros. (Caterpillar, 2018)

El objetivo principal del sistema VIMS es proporcionar a los operadores y a todo le personal de mantenimiento, información que permita identificar con antelación problemas que puedan escalar a fallas mayores, también sirve para optimizar el rendimiento del equipo y aumentar la seguridad operativa. VIMS trabaja a través de una red de sensores que están distribuidos a lo largo del camión, estos miden continuamente parámetros críticos como las temperaturas de los fluidos, velocidades, flujo de combustible, vibraciones, etc. (Caterpillar, 2018)

Las funcionalidades clave del sistema VIMS que lo hacen relevante para la el mantenimientos predictivos en motores diésel incluyen:

- Monitoreo en Tiempo Real: El VIMS recolecta información a alta frecuencia, esto permite detectar instantáneamente valores fuera de rango lo cual es importante para poder identificar cambios en el rendimiento o condiciones que pueden indicar una falla inminente. (Caterpillar, 2018)
- Capacidad de Registro de Datos: El sistema almacena un historial detallado de indicadores operativos y eventos de la máquina. Esto permite el análisis de tendencias a largo plazo, también la reconstrucción de eventos que llevaron a una falla y el resultado de intervenciones de mantenimiento (CAT, 2022). Los datos registrados

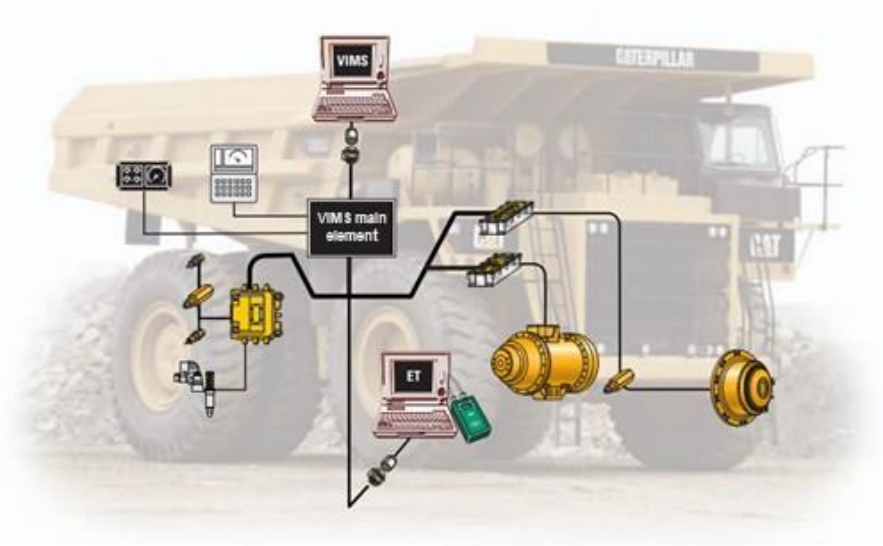
pueden descargarse y analizarse utilizando software específico, como VIMS PC, para obtener informes detallados y gráficos de rendimiento.

- Generación de Advertencias y Alarmas: VIMS está programado para trabajar con limites niños y máximos para cada indicador. Cuando un valor se aproxima o excede estos límites, el sistema genera advertencias para el operador, estas pueden ser visuales o auditivas. Incluso en cas (Caterpillar, 2018) Esta jerarquía de alertas facilita la priorización de las acciones de mantenimiento.
- Detección de Eventos: El VIMS es capaz de archivar eventos específicos, los cuales son ocurrencias que se originan cuando se exceden los límites de operación o que indican un funcionamiento anómalo. Estos eventos se registran con fecha y hora, además de un recopilado de los datos en el momento del evento, lo que servirá para su diagnóstico. (Caterpillar, 2020).
- Integración con Otros Sistemas: El VIMS es capaz de integrarse con la red de comunicaciones de la maquina y permitiéndole interactuar con otros módulos de control electrónicos del motor, la transmisión y los sistemas hidráulicos.

En el contexto del mantenimiento predictivo de motores Diesel, el VIMS brinda información esencial sobre el estado del motor, tales como, temperatura de escape, presiones de aceite, velocidades del motor y cargas, las cuales, al ser analizadas en conjunto con otras técnicas de monitoreo como el análisis de aceite, logran mostrar el estado de salud del motor. Con lo mencionado, se prueba lo esencial de contar con esta herramienta para el desarrollo de algoritmos de predicción de fallas.

Figura 23

Controlador VIMS



Nota: CAT

2.2.3 Camión minero CAT 797F

Para un mayor entendimiento del contexto operativo de los camiones mineros es necesario poder explicar la maquina sobre la cual se enfoca el trabajo. El camión Caterpillar 797F es un camión de acarreo minero Ultraclass, que está diseñado para trabajo de movimiento en minas de tajo abierto. Este modelo se impone en la vanguarda de los modelos CAT en cuestión de capacidad de carga, tecnología, eficiencia y durabilidad, por lo que es un equipo muy utilizado a nivel mundial. (Caterpillar, 2024). Su selección como plataforma de estudio permite analizar la predicción de fallas en un escenario de alta exigencia y con implicaciones significativas para la productividad de las operaciones mineras.

Las características principales del camión minero Cat 797F son:

- Capacidad de Carga Útil: El camión 797F tiene una carga útil nominal de 363 toneladas métricos o 400 toneladas cortas, por lo que es uno de los camiones más

grandes dentro de la industria. (Caterpillar, 2024). Esta capacidad exige una robustez estructural y una potencia de motor alto.

- Motorización: Cuenta con un motor Diesel Caterpillar C175-20, un motor de 20 cilindros en configuración tipo V, además de tener una potencia bruta de 2.983 kW (4.000 hp) (Caterpillar, 2024).Este motor es sometido a altas temperaturas y cargas mecánicas extremas durante su vida de operación, por lo que también es el foco principal de fallas. Su complejidad y potencia hace que sea necesario un monitoreo constante de su estado de salud.
- Tren de Potencia Integrado: El 797F cuenta con un tren de potencia mecánico, diseñado por Caterpillar, este tren cuenta con una transmisión de 7 velocidades con un control secundario. Esta configuración brinda una transferencia de potencia al suelo óptima, además de una excelente capacidad de subida.(Caterpillar, 2024)
- Tecnología a Bordo: El camión 797F incorpora una avanzada tecnología de monitoreo y diagnóstico, que incluye el Sistema de Información Vital del Motor(VIMS), la información que transmite el controlador es esencial para recolección de datos en tiempo real que se aplicaran en el presente trabajo (Caterpillar, 2024).

La comprensión del entorno de trabajo del camión 797F, es importante para poder contextualizar las técnicas de análisis de aceite y el monitoreo de condiciones a través de VIMS.

2.2.4 Vida Útil y Desgaste del motor C175

La vida útil de los componentes en un camión minero como el Caterpillar 797F es un factor crítico que influye directamente en la disponibilidad operativa, los costos de mantenimiento y la planificación de la producción en la industria minera. A diferencia de un equipo de menor escala, los componentes de estos gigantes están diseñados para operar bajo condiciones de estrés extremo y cargas continuas, lo que dicta ciclos de vida específicos y requiere un monitoreo constante para predecir su eventual desgaste y falla

(Caterpillar, 2024). Comprender estos ciclos y los factores que los afectan es fundamental para el desarrollo de modelos predictivos de fallas.

La vida útil esperada de los componentes principales se estima a partir de la experiencia del fabricante, datos históricos de operación de flotas y análisis de fallas. Sin embargo, factores como las condiciones ambientales (altas temperaturas, polvo, altitud), la severidad de la aplicación (distancia de acarreo, pendientes, material transportado), las prácticas operativas (ej., hábitos de conducción) y la calidad del mantenimiento (ej., lubricación, filtración) pueden acortar o prolongar significativamente esta vida útil proyectada (Mobley, 2012).

Para el motor C175-20, su vida útil hasta el primer reacondicionamiento mayor (major overhaul) o reemplazo puede variar, pero típicamente se mide en horas de operación del motor. Para motores de este calibre en aplicaciones mineras pesadas, un objetivo puede ser de 15,000 a 25,000 horas, o incluso más con programas de monitoreo y mantenimiento proactivo. La vida útil está directamente ligada a la eficiencia de la combustión, la calidad de la lubricación, la gestión térmica y el control de contaminantes (Caterpillar, 2024).

Figura 24

Motor C175-20

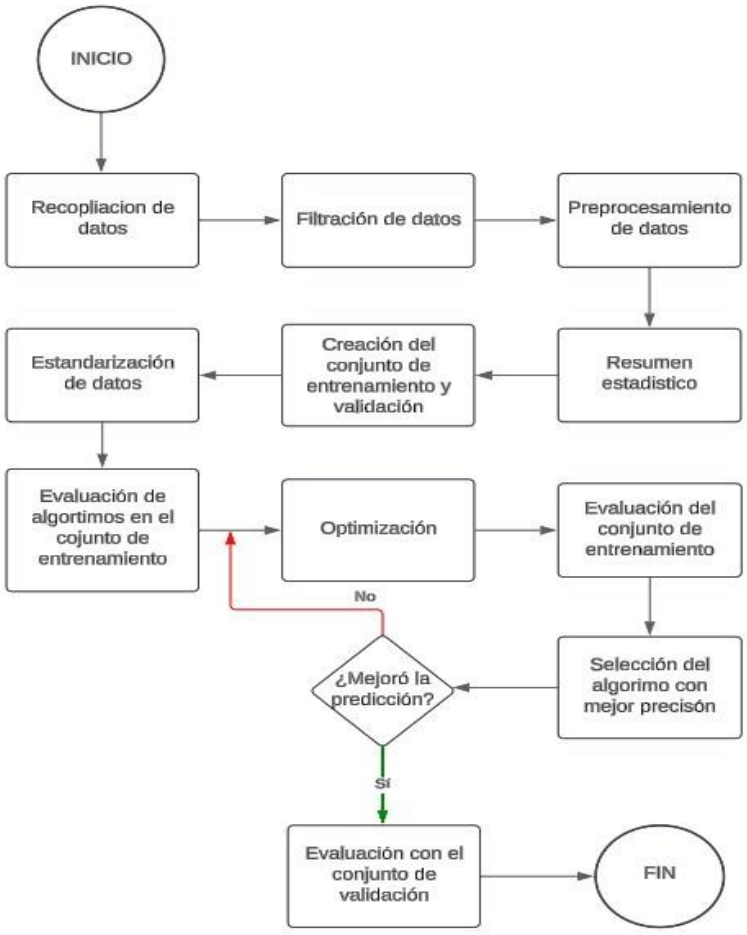
NOTA: Foto propia

Capítulo III. Desarrollo del trabajo de investigación.

A partir de la información previa del marco teórico y conceptual, se procede a la parte empírica de la investigación. Para ello se utilizó dos fuentes principales de información: la data recopilada de los análisis de aceite de motor enviados al laboratorio; y la data del controlador del equipo VIMS, que adquiere datos en tiempo real de eventos o condiciones físicas del equipo. Toda esta información esta gestionada en la plataforma Google Cloud Platform (GSP) y la información de ambas fuentes está ligada al número se serie del camión y la fecha. Estas son ordenadas según algunos datos descriptivos de relevancia como la operación minera, el tipo de motor y principalmente la codificación del motor estudiado.

Figura 25

Diagrama de flujo de desarrollo de trabajo de suficiencia.



Nota: elaboración propia

3.1 Recopilación de datos

3.1.1 Datos Descriptivos

Son los datos descriptivos que nos permite saber información relevante del motor que se está analizando, para ello estamos utilizando las siguientes variables:

Tabla 3:

Datos descriptivos del motor

Variable	Descripción
OPERACION	Operación donde se encuentra el equipo que el motor está montado
CONFIG_OPERACION	Tipo de Configuración del equipo
FLAG_ULTIMO_REGISTRO	Muestra el valor de “1” al último registro del motor, caso contrario muestra el valor de “0”.
TIPO_MOTOR	Muestra los 2 tipos de motor con configuración HA a analizar, “4x2” y “4x4”.
NOMBRE_EQUIPO	Nombre del equipo donde está montado el motor
MTR	Codificación del motor a analizar
HORAS_ACUMULADAS	Es las horas del motor durante el muestreo de los datos.

Nota: elaboración propia

3.1.2 Análisis de aceite de motor

Como lo mencionado en el marco conceptual, el análisis de aceite de motor es una herramienta de monitoreo del estado del motor, permite detectar problemas de desgaste a través del análisis de muestras de aceite en laboratorio donde se busca de partículas, contaminantes y otros indicadores de problemas potenciales. Para nuestro proyecto se toma muestras mensuales de cada camión y se envían a los laboratorios de Ferreyros en Lima para su análisis, el cual mide las variables mencionadas en la Tabla 4.

Tabla 4

Tabla de datos del análisis de aceite de motor

Variable	Descripción
SOS_FE_ACUM	Variable que muestra la cantidad acumulada de partículas por millón de Hierro(Fe)
SOS_PB_ACUM	Variable que muestra la cantidad acumulada de partículas por millón de Plomo(Pb)
SOS_NA_ACUM	Variable que muestra la cantidad acumulada de partículas por millón de Sodio(Na)
SOS_SI_ACUM	Variable que muestra la cantidad acumulada de partículas por millón de Silicio(SI)
SOS_CU_ACUM	Variable que muestra la cantidad acumulada de partículas por millón de Cobre(Cu)
SOS_CR_ACUM:	Variable que muestra la cantidad acumulada de partículas por millón de Cromo(Cr)
SOS_NI_ACUM:	Variable que muestra la cantidad acumulada de partículas por millón de Níquel(Ni)
SOS_OXID_ACUM	Variable que muestra la cantidad acumulada de partículas por millón de óxido
SOS_NIT_ACUM	Variable que muestra la cantidad acumulada de partículas por millón de nitrato
SOS_VISC100_ACUM	Variable que acumula la cantidad de eventos de Viscosidad alta.
SOS_SN_ACUM	Variable que muestra la cantidad acumulada de partículas por millón de Estaño(Sn)
SOS_TBN_ACUM	Variable que mide el numero total de bases de viscosidad.
SOS_AL_ACUM	Variable que muestra la cantidad acumulada de partículas por millón de Aluminio(Al)
SOS_B_ACUM	Variable que muestra la cantidad acumulada de partículas por millón de Boro(B)
SOS_CA_ACUM	Variable que muestra la cantidad acumulada de partículas por millón de Calcio(Ca)
SOS_MG_ACUM	Variable que muestra la cantidad acumulada de partículas por millón de Magnesio(Mg)
SOS_MO_ACUM	Variable que muestra la cantidad acumulada de partículas por millón de Molibdeno(Mo)
SOS_P_ACUM	Variable que muestra la cantidad acumulada de partículas por millón de Fosforo(P)
SOS_SUL_ACUM	Variable que muestra la cantidad acumulada de partículas por millón de Sulfato
SOS_ZN_ACUM	Variable que muestra la cantidad acumulada de partículas por millón de Zinc(Zn)
SOS_K_ACUM	Variable que muestra la cantidad acumulada de partículas por millón de Potasio(K)

Nota: elaboración propia

3.1.3 VIMS

Como lo explicado en el marco conceptual. Dentro de los camiones 797F se encuentra el controlador VIMS, este controlador recolecta diferentes datos de presión, temperatura y niveles de aceite y lubricante del motor. Estos datos son almacenados diariamente para cada camión de manera automática en la nube y utilizados para realizar análisis predictivos. Para el siguiente trabajo de suficiencia, se estará tomando las variables presentes en la Tabla 5

Tabla 5

Tabla de datos de VINS

Variable	Descripción
LOW_ENG_OIL_PRS_L1_CNT	Es un acumulador de baja presión de aceite de motor, cuando supera el nivel bajo
LOW_ENG_OIL_PRS_L2_CNT	Es un acumulador de baja presión de aceite de motor, cuando supera el nivel medio
LOW_ENG_OIL_PRS_L3_CNT	Es un acumulador de baja presión de aceite de motor, cuando supera el nivel alto
ENG_OIL_FILT_REST_W_CNT	Es un acumulador de evento de restricción de filtro de aceite de motor
LOW_ENG_PREL_PRS_L1_CNT	Es un acumulador de baja presión de prelubricación de motor, cuando supera el nivel bajo
LOW_ENG_PREL_PRS_L2_CNT	Es un acumulador de baja presión de prelubricación de motor, cuando supera el nivel medio
LOW_ENG_PREL_PRS_L3_CNT	Es un acumulador de baja presión de prelubricación de motor, cuando supera el nivel alto
ENG_PRELUBE_CNT	Son contadores acumulativos de Override Prelube
HI_ENG_OIL_TMP_L1_CNT	Es un acumulador de alta temperatura de aceite de motor, cuando supera el nivel bajo
HI_ENG_OIL_TMP_L2_CNT	Es un acumulador de alta temperatura de aceite de motor, cuando supera el nivel medio
HI_ENG_OIL_TMP_L3_CNT	Es un acumulador de alta temperatura de aceite de motor, cuando supera el nivel alto
HI_ENG_COOL_TMP_L1_CNT	Es un acumulador de alta temperatura de refrigerante de motor, cuando supera el nivel bajo
HI_ENG_COOL_TMP_L2_CNT	Es un acumulador de alta temperatura de refrigerante de motor, cuando supera el nivel medio
HI_ENG_COOL_TMP_L3_CNT	Es un acumulador de alta temperatura de refrigerante de motor, cuando supera el nivel alto

ENG_OVRSPD_CNT	Es un acumulador de eventos de sobrevelocidad de motor
OIL_LEVEL_LOW_MARK_N3_CNT	Es un acumulador de eventos de bajo nivel de aceite de motor
OIL_LEVEL_ADD_MARK_CNT	Es un acumulador de eventos de añadir aceite de motor
LOW_COOL_TMP_CNT	Es un acumulador de eventos baja temperatura de refrigerante
HI_TURBIN_TEMP_1_CNT	Es un acumulador de alta temperatura de turbo N1 del motor
HI_TURBIN_TEMP_2_CNT	Es un acumulador de alta temperatura de turbo N2 del motor
HI_TURBIN_TEMP_3_CNT	Es un acumulador de alta temperatura de turbo N3 del motor
HI_TURBIN_TEMP_4_CNT	Es un acumulador de alta temperatura de turbo N4 del motor

Nota: elaboración propia

3.2 Filtración de datos

Con la finalidad de poder enfocar el trabajo de suficiencia a una muestra en particular, se ha realizado la siguiente condición para los datos a analizar:

- Se toma solo motores CAT 797F con la configuración High Altitud
- Considera motores que fueron desmontados de manera correctiva y no de manera preventiva.
- Se consideran motores que estuvieron más de 500 horas instalados en el camión
- Se considera motores que tuvieron algún evento antes de falla y no por situaciones imprevistas(accidentes)
- La clase FALLA se considera como 0(No falla) cuanto se encuentra desde de las 500 horas de inicio hasta 1000 horas antes de su último registro y "1"(Falla) cuando se encuentra a desde 1000 horas antes del último registro hasta el último registro.
- Se trabajo manualmente la cantidad de datos con la finalidad de poder balancear la clase FALLA.

Con estas restricciones se asegura que el algoritmo pueda tener en cuenta solamente los eventos donde el motor ha sufrido una falla y no por otros aspectos fortuitos.

3.3 Preprocesamiento de datos

Este paso se centró en la adquisición de las librerías de Python a utilizar y la limpieza de los datos. En primer lugar, se realiza el llamado a las librerías de Python enfocadas en el aprendizaje supervisado, que se utilizará en el proyecto.

Figura 26

Librerías de Python a utilizar

```
#1.1 Load Libraries
import math
import seaborn as sns
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
# Load sklearn
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import GridSearchCV, GroupKFold
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.pipeline import Pipeline
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.metrics import roc_curve
from sklearn.metrics import precision_score, recall_score, f1_score
from sklearn.metrics import precision_recall_curve, auc

import warnings
warnings.filterwarnings('ignore')
```

En segundo lugar, Una vez ingresado las librerías se procede a cargar la base de datos con la información de los motores descargada de las fuentes mencionadas anteriormente: Datos descriptivos, Análisis de aceite y data VIMS con la función read_csv.

Figura 27

Carga de base de datos

```
# Load dataset
filename="datos_tesis_final_3.csv"
dataset = pd.read_csv(filename, sep=';')
```

En tercer lugar, es relevante revisar la distribución de la clase sea balanceada o no, con la finalidad de aplicar los algoritmos necesarios, para ello utilizamos el siguiente código

Figura 28

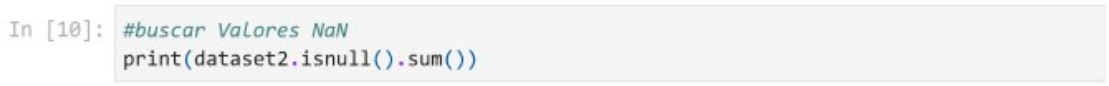
Análisis de distribución de clases

Con el código anterior se observa que, de los 12826 datos, 6637 son considerados como falla y 6189 como no falla. Por lo que la clase llamada “FALLA” se encuentra balanceada. Es prioritario revisar esto, ya que una data desbalanceada genera modelos sesgados que clasifican erróneamente la clase minoritaria y dan como resultado un desempeño de clasificación deficiente

En último lugar se revisará si dentro de la data existen valores vacíos o NaN, para ello se utiliza el siguiente código.

Figura 29

Algoritmo para busca de valores NaN



Con este código pudimos revisar que la base de datos no cuenta con valores NaN. Esta revisión es importante debido a que si existiera algún valor NaN nos puede impedir la aplicación de la mayoría de los algoritmos.

3.4 Resumen Estadístico

En esta etapa se revisará los datos de manera analítica y grafica.

3.4.1 Fase Analítica:

En primer lugar, para la parte analítica se puede echar un vistazo al resumen estadístico de cada atributo, esto incluye la media, valores mínimos y máximo, así como algunos percentiles. Para ello se utiliza el siguiente código.

Figura 30:

Resumen analíticos de los datos

```
In [7]: #Descripcion estadistico
        pd.set_option('display.precision', 3)
        dataset2.describe()
```

Out[7]:

	HORAS_ACUMULADAS	SOS_FE_ACUM	SOS_PB_ACUM	SOS_NA_ACUM	SOS_SI_ACUM
count	12826.000	12826.000	12826.000	12826.000	12826.00
mean	9370.887	297.858	8.862	113.304	156.91
std	4226.199	320.532	10.469	195.606	152.73
min	0.000	0.000	0.000	0.000	0.00
25%	6148.873	0.000	0.890	0.000	33.00
50%	9399.216	185.500	6.000	35.000	124.00
75%	12534.425	511.000	12.000	125.000	233.00
max	20097.437	1542.000	66.030	1395.000	974.45

8 rows × 47 columns

La tabla resultante presenta las estadísticas descriptivas de las horas acumuladas del motor, además de la concentración de elementos en el aceite, obtenidas del análisis de la data global. Tomando como ejemplo las columnas mostradas, estas representan: HORAS_ACUMULADAS (horas de operación del motor), SOS_FE_ACUM (hierro acumulado en el aceite, en ppm), SOS_PB_ACUM (plomo acumulado en el aceite, en ppm), SOS_NA_ACUM (sodio acumulado en el aceite, en ppm), y SOS_SI_ACUM (silicio acumulado en el aceite, en ppm). Y se puede sacar las siguientes conclusiones:

Horas acumuladas:

- Conteo (count): Se observa un total de 12,826 registros, lo que indica un tamaño de muestra considerable para el estudio.
- Media (mean): La media de horas acumuladas es de aproximadamente 9,370 horas, sugiriendo que la mayoría de los motores en la base de datos han operado durante un tiempo considerable, abarcando una parte significativa de su vida útil esperada antes de un reacondicionamiento mayor.
- Desviación Estándar (std): Con una desviación estándar de 4,226 horas, existe una variabilidad considerable en las horas de operación entre las muestras, lo cual es beneficioso para capturar diferentes etapas de desgaste de los componentes.
- Mínimo (min): El valor mínimo es 0 horas, lo que puede corresponder a registros iniciales o de motores recién instalados/reacondicionados.
- Máximo (max): El valor máximo alcanza 20,097 horas, lo cual es un indicativo de motores que han estado en servicio por un período muy extendido, probablemente acercándose o superando su vida útil esperada entre reacondicionamientos.
- Percentiles (25%, 50%, 75%): Los percentiles (6,148, 9,399 y 12,534 horas, respectivamente) muestran una distribución relativamente simétrica alrededor de la media, aunque el 50% (mediana) está muy cerca de la media, sugiriendo una distribución de horas que puede ser cercana a la normal o ligeramente sesgada.

Concentraciones de Elementos de Desgaste:

- Hierro Acumulado (SOS_FE_ACUM): La concentración media de hierro es de 297ppm, con un máximo de 1542 ppm. La alta desviación estándar (320 ppm) y la diferencia entre la mediana (185. ppm) y la media indican una distribución sesgada a la derecha, con la presencia de valores atípicos (outliers) altos que sugieren eventos

de desgaste significativo en algunos casos. El hierro es un indicador primario del desgaste de componentes como camisas de cilindro, anillos de pistón y engranajes.

- Plomo Acumulado (SOS_PB_ACUM): La media de plomo es 8.86 ppm, con un máximo de 66.03 ppm. Similar al hierro, la desviación estándar (10.45 ppm) es alta en relación con la media, y la mediana (6 ppm) es inferior a la media, lo que también sugiere una distribución sesgada y la presencia de eventos de desgaste de cojinetes (que a menudo contienen plomo) en algunos registros.

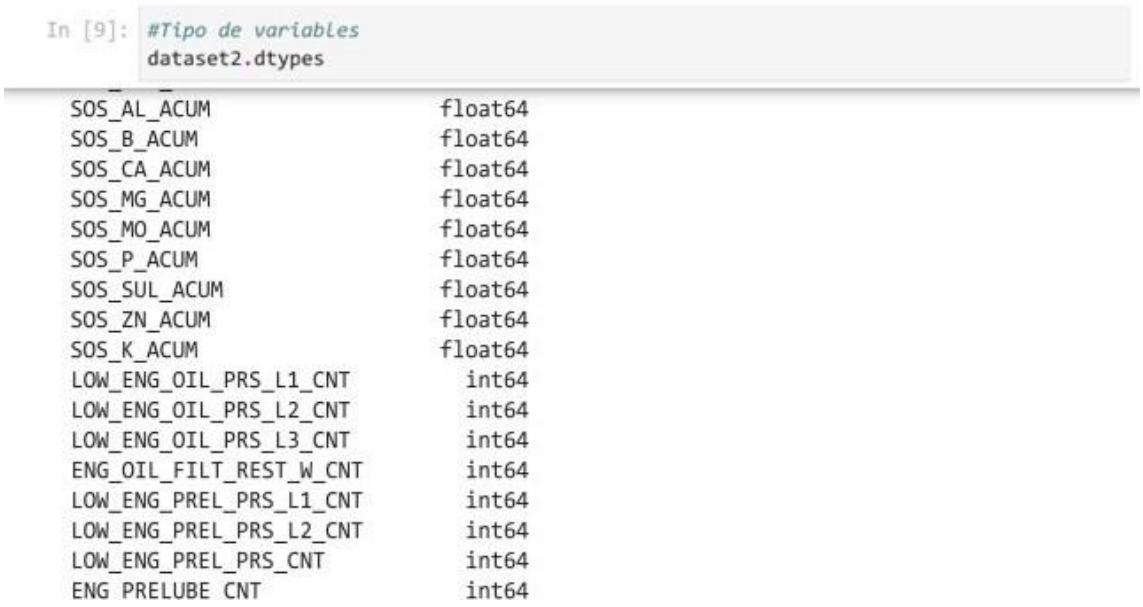
- Sodio Acumulado (SOS_NA_ACUM): Con una media de 113.30 ppm y un máximo de 1395 ppm, el sodio es un fuerte indicador de contaminación por anticongelante. La enorme desviación estándar (195.6 ppm) y la diferencia entre la media y la mediana (35 ppm) revelan que, si bien muchos registros pueden tener niveles bajos de sodio, existe un número considerable de casos con alta contaminación por glicol, lo que representa un riesgo significativo de falla del motor.

- Silicio Acumulado (SOS_SI_ACUM): La media de silicio es 156.91 ppm, alcanzando un máximo de 974.45 ppm. La desviación estándar (152.73 ppm) también es elevada, y la mediana (124 ppm) es menor que la media, indicando una distribución sesgada con picos de contaminación por polvo. El silicio es un abrasivo crítico que, cuando entra al motor (generalmente por problemas en el filtro de aire), acelera drásticamente el desgaste interno

En segundo lugar, se aplica la verificación y confirmación de los tipos de datos de cada variable. Esta etapa asegura que las variables sean interpretadas correctamente por los algoritmos de aprendizaje automático y que las operaciones subsecuentes (como cálculos numéricos o manipulación de datos) se realicen de manera adecuada (Provost, 2013). Para la data, se examinaron los tipos de datos de cada columna, revelando una combinación de tipos numéricos adecuados para el análisis cuantitativo.

Figura 31

Muestreo del tipo de datos.



Los tipos de datos identificados son coherentes con la naturaleza de las variables recopiladas del análisis de aceite de motor y del sistema VIMS.

- Análisis de aceite de motor: La mayoría de las variables de análisis de aceite, como SOS_AL_ACUM (aluminio acumulado), SOS_B_ACUM (boro acumulado), SOS_CA_ACUM (calcio acumulado), SOS_MG_ACUM (magnesio acumulado), SOS_MO_ACUM (molibdeno acumulado), SOS_P_ACUM (fósforo acumulado), SOS_SUL_ACUM (azufre acumulado), SOS_ZN_ACUM (zinc acumulado) y SOS_K_ACUM (potasio acumulado), se han detectado como float64. Este tipo de dato de punto flotante de doble precisión (64 bits) es apropiado para representar concentraciones de elementos en partes por millón (ppm) o cualquier otra medida que requiera valores decimales. La precisión de float64 es suficiente para la granularidad de los datos de análisis de aceite, evitando pérdidas de información

- VIMS: Las variables relacionadas con el controlador del sistema VIMS, tales como LOW_ENG_OIL_PRS_L1_CNT, LOW_ENG_OIL_PRS_L2_CNT, LOW_ENG_OIL_PRS_L3_CNT (contadores de eventos de baja presión de aceite del

motor en diferentes niveles de advertencia), ENG_OIL_FILT_REST_W_CNT (contador de restricción del filtro de aceite del motor), LOW_ENG_PREL_PRS_L1_CNT, LOW_ENG_PREL_PRS_L2_CNT (contadores de baja presión de pre-lubricación) y ENG_PRELUBE_CNT (contador de ciclos de pre-lubricación), se han identificado como int64. Este tipo de dato de entero de 64 bits es idóneo para representar conteos discretos, ya que estas variables registran el número de veces que un evento específico ha ocurrido, sin fracciones

La correcta asignación de tipos de datos es un paso importante en la fase de preprocesamiento, ya que garantiza que las transformaciones de datos, la detección de valores atípicos y, en última instancia, los algoritmos de aprendizaje automático se apliquen de manera efectiva. La homogeneidad de los datos numéricos (float64 e int64) dentro de sus respectivas categorías indica un conjunto de datos limpio y listo para las etapas subsiguientes de análisis y modelado.

3.4.2 Fase Grafica

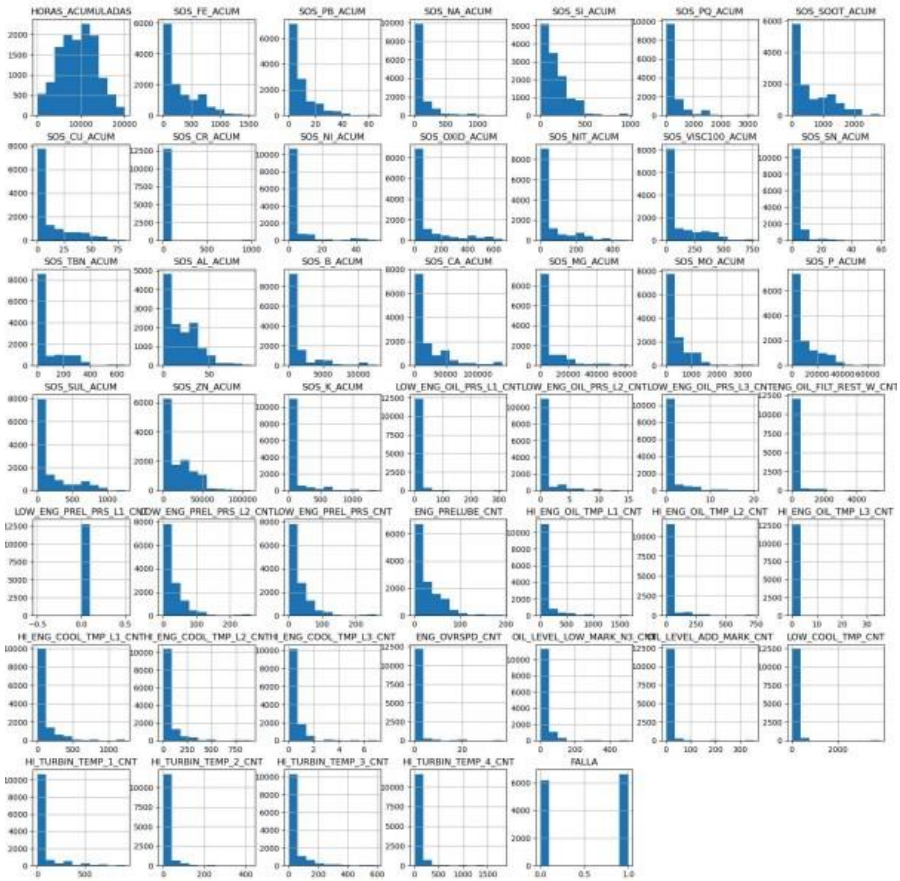
Después de la verificación de los tipos de datos y el análisis de estadísticas descriptivas, se va a realizar la visualización de la distribución de cada variable en el análisis exploratorio de datos (AED). Para el grafico univariado, los histogramas permiten observar la forma de la distribución de una variable, identificar la presencia de sesgos, valores atípicos, multimodalidades y la concentración de datos en rangos específicos. (VanderPlas, 2016). Para este estudio, se generaron histogramas para todas las variables numéricas en lavase de datos utilizando la librería Matplotlib en Python, con el objetivo de comprender mejor las características individuales de cada serie de datos de monitoreo.

La Figura 31 presenta una selección representativa de los histogramas generados, mostrando la distribución de las horas de operación, diversas concentraciones de elementos de análisis de aceite (acumulados) y contadores de eventos del sistema VIMS.

Figura 32:

Histograma de los datos.

```
#histograma
# histograms - Matplotlib
fig = plt.figure(figsize = (20,20))
ax = fig.gca()
dataset2.hist(ax=ax)
plt.show()
#N: validar la data que este dentro del rango esperado.
```



El análisis de los histogramas reveló patrones de distribución específicos para cada variable, ofreciendo información valiosa sobre el comportamiento de los parámetros del motor, los datos de análisis de aceite en el conjunto de datos y los contadores de eventos del VIMS:

- Horas Acumuladas: Muestra una forma bimodal o al menos una distribución con picos en diferentes rangos de horas. Se observa una concentración significativa de registros en rangos de horas tempranas (entre a 6000 a 8000 horas) que significa principalmente fallas de fabrica o de mantenimiento. y otra concentración en rangos

de horas más avanzadas (desde los 1200 a 14000) y luego una cola extendida hacia la derecha a 20000, que indica que hay motores que están operando cerca o más allá de su vida útil esperada (16000 horas) entre reacondicionamientos, lo cual es crucial para la predicción de fallas.

- Análisis de aceite: La mayoría de los histogramas de metales de desgaste y aditivos exhiben una distribución altamente sesgada a la derecha. Esto significa que la gran mayoría de los valores se concentran en niveles bajos (cerca de cero), lo cual es esperable y deseable, ya que indica que, en la mayoría de los casos, los motores operan sin un desgaste anormal significativo o una contaminación severa; y con una cola larga y delgada que se extiende hacia valores altos, estos valores representan eventos de desgaste o contaminación elevados que podrían ser indicativos de una anomalía incipiente o de una falla en desarrollo. La escasez de estos valores altos resalta su importancia como posibles indicadores de problemas. La marcada asimetría de estas distribuciones sugiere que técnicas de transformación de datos podrían ser necesarias durante el preprocesamiento para mejorar el rendimiento de ciertos algoritmos de aprendizaje automático

- Contadores de eventos VIMS: Prácticamente todos los contadores de eventos del VIMS muestran una distribución extremadamente sesgada a la derecha, con una inmensa mayoría de los registros en cero o en valores muy bajos. Estos valores en cero o bajos son positivos, ya que indica que la mayoría de las veces los camiones operan dentro de los parámetros esperados y las advertencias o fallas no se activan frecuentemente. Los pocos casos con conteos elevados de eventos son críticos. Estos representan situaciones donde los límites de advertencia o falla fueron superados repetidamente, que pueden señalar problemas graves y persistentes. Estos valores son importantes para la predicción de fallas y pueden ser indicativos de la proximidad a una falla crítica. Debido a que son datos discretos es necesario usar modelos que manejen datos dispersos.

- Variable objetivo “Falla”: Es una variable binaria de distribución balanceada de clases, donde la clase "No Falla" es ligeramente predominante a la clase "Falla". Esta característica del conjunto de datos requerirá técnicas específicas para manejar el ligero desbalance de clases durante el entrenamiento del modelo.

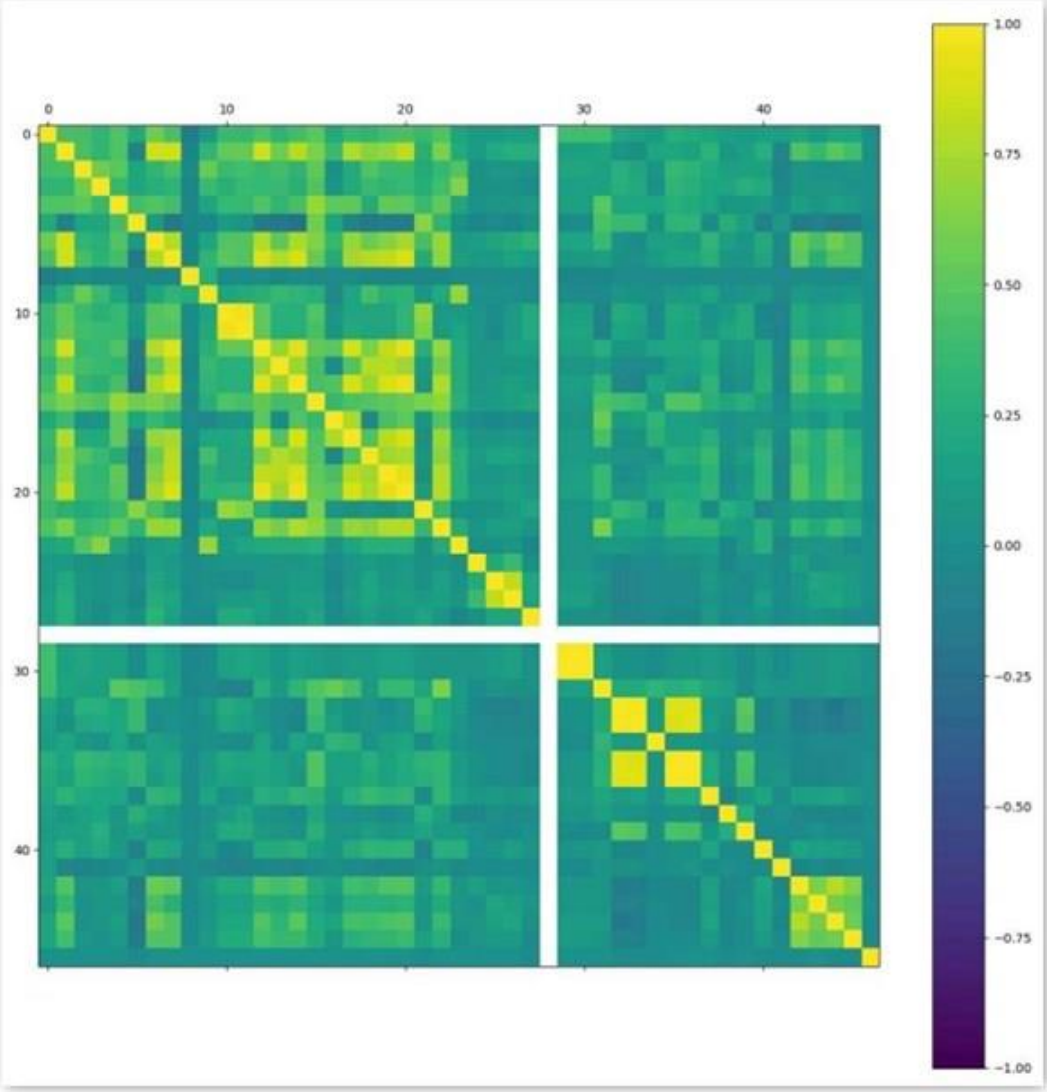
Para el gráfico multivariado, se optó por una gráfica de correlación La Figura 32 presenta un mapa de calor donde cada celda representa el coeficiente de correlación de Pearson entre dos variables. El color de cada celda indica la fuerza y dirección de la correlación, de acuerdo con la barra de color a la derecha:

- Amarillo brillante (cercano a 1): Indica una correlación positiva fuerte, es decir, cuando una variable aumenta, la otra también tiende a aumentar de manera proporcional.
- Verde (cercano a 0): Sugiere una correlación débil o nula, lo que significa que no hay una relación lineal clara entre las variables.
- Púrpura oscuro (cercano a -1): Representa una correlación negativa fuerte, lo que implica que cuando una variable aumenta, la otra tiende a disminuir.

Figura 33

Grafica de correlación de los atributos.

```
#correlacion entre atributos
# Convert non-numeric columns to NaN, then drop them
numeric_dataset = dataset2.apply(pd.to_numeric, errors='coerce')
# Drop columns with all NaN values
numeric_dataset = numeric_dataset.dropna(axis=1, how='all')

fig = plt.figure(figsize=(15,15))
ax = fig.add_subplot(111)
cax = ax.matshow(numeric_dataset.corr(), vmin=-1, vmax=1, interpolation='none')
fig.colorbar(cax)
plt.show()
```



El mapa de calor proporciona información crucial para la selección de características y el diseño del modelo predictivo:

Correlaciones Positivas::

- Variables de Análisis de Aceite: Es probable que los elementos de desgaste como Fe, Cu, Al, Pb y algunos contaminantes o aditivos como Si y Na, muestren correlaciones positivas entre sí, especialmente a medida que el motor acumula horas o experimenta un desgaste generalizado. Un aumento en un tipo de partícula de desgaste podría estar asociado con el aumento de otras, si el desgaste afecta múltiples componentes o si hay una progresión de fallas.
- Contadores de Eventos VIMS: Los contadores de eventos VIMS como los distintos niveles de advertencia de presión de aceite o temperatura, tienden a estar altamente correlacionados entre sí. Por ejemplo, si la presión de aceite baja al Nivel bajo, es probable que también alcance el Nivel medio o alto si el problema persiste, generando una correlación fuerte entre estos contadores.

Correlaciones Nulas:

- Existen muchas celdas en color verde, indicando una correlación lineal baja o inexistente entre pares de variables. Esto es importante porque sugiere que estas variables pueden proporcionar información independiente y complementaria al modelo predictivo, evitando la redundancia. Por ejemplo, el contador de eventos de sobrevelocidad del motor no tiene ninguna correlación lineal con la concentración de un metal de desgaste específico

Correlaciones Negativas

- Si bien menos predominantes en la visualización general, la presencia de celdas en tonos púrpuras indicaría correlaciones negativas. Por ejemplo, el Número Básico Total (TBN) del aceite se correlaciona negativamente con variables que indican la degradación del aceite o el aumento de ácidos, como la oxidación. A medida que la oxidación aumenta, el TBN disminuye

3.5 Creación del conjunto de datos para entrenamiento y validación.

Es importante recalcar que debido a que se está trabajando con datos globales divididos en subconjuntos donde cada uno de ellos representan un motor a lo largo del tiempo, se realizara la sección utilizando como base la cantidad de motores, no la cantidad de datos. Para poder crear los conjuntos de datos para entrenamiento y validación se tomará la proporción recomendada de 80 y 20, es decir del total de motores, el 80% fue utilizado para el entrenamiento y el otro 20% para la validación, por lo siguiente:

- Evitar el sobreajuste, si un modelo se entrena y evalúa sobre el mismo conjunto de datos, es probable que memorice los datos de entrenamiento, incluyendo el ruido y las particularidades específicas de esa muestra. Al reservar un 20% de los datos para validación, se simula la aplicación del modelo a situaciones nuevas, permitiendo medir qué tan bien generaliza los patrones aprendidos a datos que no ha visto durante el entrenamiento.

- Balance entre Aprendizaje y Evaluación, al escoger mayor proporción para el entrenamiento permite que el modelo tenga suficientes ejemplos para aprender los patrones subyacentes en los datos. Un conjunto de entrenamiento demasiado pequeño podría llevar a un "subajuste" (*underfitting*), donde el modelo es demasiado simple y no logra capturar la complejidad de las relaciones en los datos. Por otro lado una proporción del 20% suele ser suficiente para tener una muestra representativa de los datos que no se usaron en el entrenamiento Un conjunto de validación demasiado pequeño podría no ser estadísticamente significativo o no representar adecuadamente la variabilidad de los datos, llevando a una evaluación poco confiable.

- Estándar de la Industria, La división 80/20 se ha convertido en una convención ampliamente aceptada y un punto de partida estándar en el campo del aprendizaje automático y la ciencia de datos. Su simplicidad facilita la comunicación y la

replicabilidad de los resultados en la comunidad científica e ingenieril. (Shmueli, 2011)

Figura 34

Algoritmo para dividir la base de datos.

```
#4.1 CREAR CONJUNTO DE VALIDACION
unique_engines = dataset2.MTR.drop_duplicates()
train_engines = unique_engines.sample(frac=0.8).values

df_train = dataset2[dataset2.MTR.isin(train_engines)]
df_test = dataset2[~dataset2.MTR.isin(train_engines)]

X_train = df_train.drop(['FALLA', 'MTR'], axis=1).astype(float)
Y_train = df_train['FALLA']
X_test= df_test.drop(['FALLA', 'MTR'], axis=1).astype(float)
Y_test= df_test['FALLA']
```

3.6 Llamado de algoritmos y estandarización de datos.

Se realiza el llamado a los algoritmos lineales y no lineales y se grabará el resultado en un vector llamado “models”, también se creó el vector “times” para poder grabar el tiempo de entrenamiento de cada algoritmo. Además, las diferentes distribuciones de los datos en bruto pueden afectar negativamente a la habilidad de los algoritmos. Por ende, se va a realizar una copia estandarizada del conjunto de datos, esto para que cada atributo tenga un valor medio de cero y una desviación estándar de uno.

Figura 35

Algoritmo para escalar la base de datos

```
pipelines.append(('ScaledLoR', Pipeline([('Scaler', StandardScaler())
pipelines.append(('ScaledLDA', Pipeline([('Scaler', StandardScaler())
pipelines.append(('Scaledk-NN', Pipeline([('Scaler', StandardScaler()
pipelines.append(('ScaledCART', Pipeline([('Scaler', StandardScaler()
```

3.7 Evaluación de algoritmos en el conjunto de entrenamiento.

El conjunto de datos es grande y esta es una buena configuración estándar. Sin embargo, debido a que no es posible determinar qué algoritmos funcionarán bien en este

conjunto de datos, evaluaremos distintos algoritmos utilizando la métrica Accuracy y la validación cruzada ya que las clases estaban balanceadas. Esta es una métrica general que dará una idea rápida de cuán correcto es un modelo dado. Estos algoritmos se dividieron en 2 grupos, algoritmos lineales/no lineales y algoritmos ensamblados. Finalmente se escogerá uno de cada tipo para optimizarlo a través de la configuración de sus hiperparámetros, con el fin de escoger el algoritmo con mejor desempeño.

3.7.1 Evaluación de los algoritmos lineales y no lineales

La selección del algoritmo de aprendizaje automático más adecuado para la predicción de fallas en motores diésel es el pilar de esta investigación. Para evaluar el rendimiento de diferentes modelos, se realizó una comparativa utilizando la métrica de precisión (accuracy), aplicada sobre los datos de entrenamiento previamente escalados. El escalado de los datos es una técnica de preprocesamiento común que asegura que las características con rangos de valores más grandes no dominen a aquellas con rangos más pequeños, lo cual ayudara para el buen desempeño de varios algoritmos, especialmente los basados en distancia, los algoritmos a analizar son los siguientes y sus resultados son los mostrados en la Figura 36

- Algoritmos lineales: LR y LDA.
- Algoritmos no lineales: CART, SVM, NB y k-NN.

Figura 36:

Resultados de evaluación de algoritmos lineales y no lineales

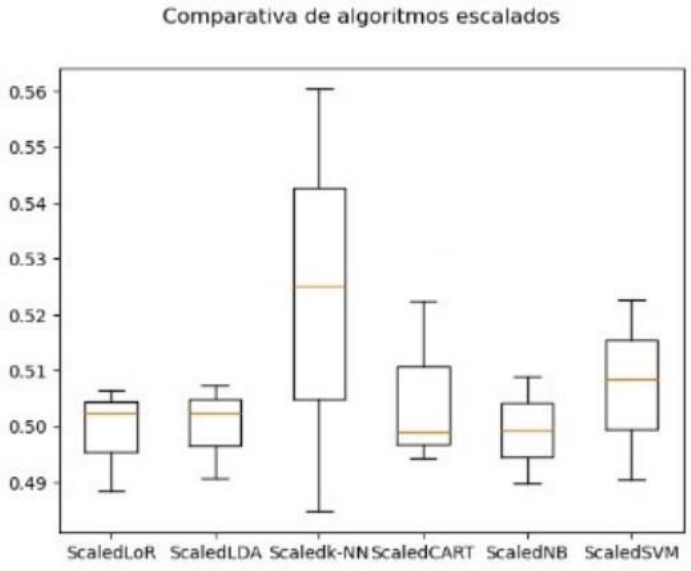
```
#Algoritmos lineales / no lineales
models = []
models.append(('LR', LogisticRegression()))
models.append(('LDA', LinearDiscriminantAnalysis()))
models.append(('KNN', KNeighborsClassifier()))
models.append(('CART', DecisionTreeClassifier()))
models.append(('NB', GaussianNB()))
models.append(('SVM', SVC()))

results = []
names = []
times=[]
import time


# Standardize the dataset
pipelines = []
pipelines.append(('ScaledLoR', Pipeline([('Scaler', StandardScaler()),('LR', LogisticRegression())])))
pipelines.append(('ScaledLDA', Pipeline([('Scaler', StandardScaler()),('LDA', LinearDiscriminantAnalysis())])))
pipelines.append(('Scaledk-NN', Pipeline([('Scaler', StandardScaler()),('KNN', KNeighborsClassifier())])))
pipelines.append(('ScaledCART', Pipeline([('Scaler', StandardScaler()),('CART', DecisionTreeClassifier())])))
pipelines.append(('ScaledNB', Pipeline([('Scaler', StandardScaler()),('NB', GaussianNB())])))
pipelines.append(('ScaledSVM', Pipeline([('Scaler', StandardScaler()),('SVM', SVC())])))
results = []
names = []
for name, model in pipelines:
    start_time = time.time()
    group_kfold=GroupKFold(n_splits=3)# hace una iteracion de datos
    cv_results = cross_val_score(model, X_train, Y_train, groups=groups, cv=group_kfold, scoring=scoring)
    results.append(cv_results)
    names.append(name)
    end_time = time.time()
    dif_time= end_time - start_time
    times.append(dif_time)
    print(f"{name}: {cv_results.mean()*100.0:,.2f}% ({cv_results.std()*100.0:,.2f}%)%' T. entrenamiento en seg: ' ({dif_time:.4f
```

```
ScaledLoR: 49.91% (0.77%)%' T. entrenamiento en seg: ' (0.2348)
ScaledLDA: 50.01% (0.70%)%' T. entrenamiento en seg: ' (0.1095)
Scaledk-NN: 52.34% (3.09%)%' T. entrenamiento en seg: ' (0.6297)
ScaledCART: 50.53% (1.23%)%' T. entrenamiento en seg: ' (0.3883)
ScaledNB: 49.93% (0.79%)%' T. entrenamiento en seg: ' (0.0757)
ScaledSVM: 50.72% (1.31%)%' T. entrenamiento en seg: ' (15.3094)
```

A primera vista se observa que el algoritmo K-NN muestra un mejor rendimiento respecto a los demás algoritmos, además de tener un tiempo de entrenamiento por debajo de 1 segundo. Para una mejor visualización de los resultados se realiza un escalamiento de la data y se exporta a una gráfica box plot que permite comparar los resultados del accuracy.

Figura 37:

Comparación de accuracy de algoritmos lineales y no lineales.



La figura 37 muestra el gráfico de cajas y bigotes que ilustra la distribución de la precisión para cada algoritmo. La caja central de cada *boxplot* denota el rango intercuartílico (IQR), mientras que la línea horizontal dentro de la caja representa la mediana de la precisión. Los bigotes se encuentran en extensión para indicar la variabilidad de los datos, excluyendo posibles valores atípicos. El eje vertical muestra la precisión, que oscila aproximadamente entre 0.485 y 0.560.

Las precisiones son relativamente moderadas, oscila principalmente entre 0.490 y 0.540. Esto muestra nuevamente la dificultad intrínseca de predecir fallas en este conjunto de datos, debido a la complejidad de las relaciones entre variables. En esta iteración, Scaledk-NN (K-Vecinos Más Cercanos Escalado) se corona como el algoritmo con la precisión más alta, situada aproximadamente en 0. 525.En adición, su caja es de mayor tamaño, lo que indica que tiene una alta variabilidad de us rendimiento, aproximadamente a 0.56. Esto señala que Scaledk-NN tiene potencial, pero su rendimiento puede ser inconsistente y sensible a la configuración de sus hiperparámetros o a la división específica de los datos.

3.7.2 Evaluación de los algoritmos ensamblados

Además de los algoritmos de aprendizaje supervisado individuales, se exploró el rendimiento de los algoritmos ensamblados (o ensemble methods) para la predicción de fallas. Los métodos de ensemble combinan las predicciones de múltiples modelos base para mejorar la precisión y la robustez general en comparación con un único modelo. Esto ayudará en la reducción del sobreajuste y la varianza, a menudo logrando un rendimiento superior (Géron, 2019). Para ello se tomará en cuenta 4 algoritmos ensamblados diferentes, 2 de tipo Bostingg y 2 tipo Bagging

- Métodos Boosting: AdaBoost(AB) y Gradient Boosting (GBM)
- Métodos Bagging: Random Forest (RF) y Extra Trees (ET)

El algoritmo trabajado es el siguiente:

Figura 38:

Evaluación de algoritmos ensamblados.

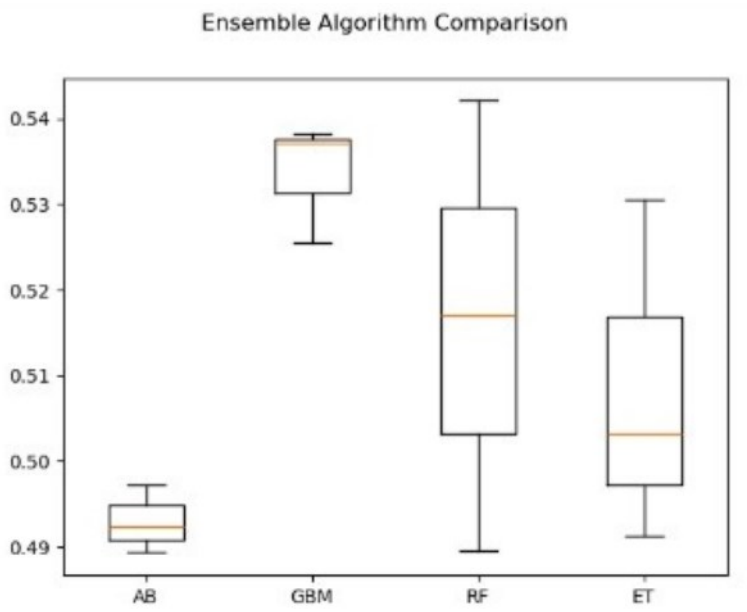
```
#Algoritmos ensamblados
# ensembles
ensembles = []
ensembles.append(('AB', AdaBoostClassifier()))
ensembles.append(('GBM', GradientBoostingClassifier()))
ensembles.append(('RF', RandomForestClassifier()))
ensembles.append(('ET', ExtraTreesClassifier()))
results = []
names = []
times=[]
import time

for name, model in ensembles:
    start_time = time.time()
    group_kfold=GroupKFold(n_splits=3)# hace una iteracion de datos
    cv_results = cross_val_score(model, X_train, Y_train, groups=groups, cv=group_kfold, scoring=scoring)
    results.append(cv_results)
    names.append(name)
    end_time = time.time()
    dif_time= end_time - start_time
    times.append(dif_time)
    print(f"{name}: {cv_results.mean()*100.0:,.2f}% ({cv_results.std()*100.0:,.2f}%)% 'T. entrenamiento en seg: ' {dif_time:.4f}
```

```
AB: 49.30% (0.33%)% 'T. entrenamiento en seg: ' 1.5242)
GBM: 53.36% (0.57%)% 'T. entrenamiento en seg: ' 5.1353)
RF: 51.62% (2.15%)% 'T. entrenamiento en seg: ' 3.7004)
ET: 50.83% (1.64%)% 'T. entrenamiento en seg: ' 1.6798)
```

La Figura 38 presenta un gráfico de cajas y bigotes que compara la precisión (accuracy) de cuatro algoritmos ensamblados: AdaBoost (AB), Gradient Boosting Machine (GBM), Random Forest (RF), y Extra Trees (ET). Al igual que en las evaluaciones anteriores, esta comparativa se realizó sobre los datos de entrenamiento escalados.

Figura 39:

Comparación de accuracy de algoritmos ensamblados



Las precisiones generales de los algoritmos ensamblados se mantienen en un rango similar al de los algoritmos individuales previamente evaluados (entre 0.48 y 0.545). Esto refuerza la idea de que la predicción de fallas en este conjunto de datos es intrínsecamente desafiante, y que la métrica de precisión por sí sola puede no capturar completamente la calidad del modelo. No obstante, GMB (Gradient Boosting Machine) muestra una mediana en la precisión más lata entre los algoritmos ensamblados, aproximadamente en 0.538. Esto lo posiciona como el algoritmo ensamblado con mayor precisión.

Capítulo IV. Análisis y Discusión de resultados

En este capítulo se procedió a comparar los 2 modelos del capítulo anterior y seleccionar el óptimo para nuestra base de datos.

4.1 Optimización de los modelos

La selección y optimización de hiperparámetros es un paso fundamental para maximizar el rendimiento de un algoritmo de aprendizaje automático. Los hiperparámetros son configuraciones externas al modelo que no se aprenden de los datos durante el entrenamiento, sino que deben ser especificados por el analista. Para la fase de optimización se escogieron los siguientes algoritmos:

- Algoritmo Lineal/No Lineal con mejor desempeño: K Nearest neighbour (KNN)
- Algoritmo Ensamblado con mejor desempeño: Gradient Boosting (GBM)

4.1.1 Optimización para el algoritmo K Nearest neighbour (KNN)

En el caso del algoritmo KNN, se realizará un ajuste de los siguientes hiperparámetros con la finalidad de encontrar el más optimo en cada uno de ellos.

- n_neighbors (Valor default= 5)
- weights (Valor default= ‘uniform’)
- algorithm (Valor default= auto)
- metric (Valor default = minkowski)

Para identificar la combinación óptima de estos hiperparámetros, se empleó la técnica de búsqueda en cuadrícula con validación cruzada (GridSearchCV). GridSearchCV explora sistemáticamente todas las combinaciones posibles de hiperparámetros dentro de un rango predefinido y evalúa el rendimiento de cada combinación utilizando validación cruzada, seleccionando finalmente aquella que produce la mejor puntuación.

Dado que los datos de la tesis provienen de series temporales o de equipos específicos donde las observaciones podrían no ser completamente independientes, se utilizó Validación Cruzada por Grupos (GroupKFold). Esta estrategia asegura que las observaciones que pertenecen al mismo motor (MTR es una codificación única para cada motor) y no sean divididas entre los conjuntos de entrenamiento y prueba en una misma partición. Esto es crucial para prevenir la fuga de datos (data leakage) y para obtener una estimación más realista de la capacidad de generalización del modelo a nuevos motores.

Previamente a la optimización, las características (X_train) fueron escaladas utilizando StandardScaler. Este paso es crítico para KNN, ya que el algoritmo se basa en la distancia entre puntos, y las características con rangos más grandes podrían dominar la métrica de distancia. Luego, Se definió una cuadrícula de hiperparámetros a explorar para el modelo KNeighborsClassifier que contiene n_neighbors, weights y metric. Después, se configuró GridSearchCV y GroupKFold(n_splits=3) para realizar una validación cruzada con 3 pliegues, garantizando que los grupos de MTR (identificador del motor) no se mezclaran entre los conjuntos de entrenamiento y validación de cada pliegue.

Figura 40:

Optimización del algoritmo KNN

```
#optimizar el algoritmo KNN
import numpy as np
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score
scaler = StandardScaler().fit(X_train)
rescaledX = scaler.transform(X_train)

model = KNeighborsClassifier()
groups = df_train['MTR']
param_grid = {
    'n_neighbors': [1,3,5,7,9],  #Controla cuántos vecinos se consideran al clasificar un punto
    'weights': [None, 'uniform','distance'], # Controla cómo se ponderan los vecinos en la decisión de clasificación o regresión
    'algorithm': ['auto', 'ball_tree', 'kd_tree','brute'],  # Define cómo se buscan los vecinos más cercanos en el conjunto de d
    'metric': ['euclidean', 'manhattan', 'minkowski']  # Especifica cómo se mide la distancia entre los puntos de datos.
}

group_kfold = GroupKFold(n_splits=3)  # Número de pliegues# hace una iteracion de datos
grid = GridSearchCV(estimator=model, param_grid=param_grid, scoring=scoring, cv=group_kfold,verbose=2)
grid_result = grid.fit(rescaledX, Y_train,groups=groups)

# Imprimir los mejores parámetros y la mejor puntuación
print("Mejores parámetros encontrados:")
print(grid_result.best_params_)

print("\nMejor puntuación obtenida:")
print(grid_result.best_score_)

# Evaluar el modelo con los mejores parámetros en los datos de entrenamiento
best_clf = grid_result.best_estimator_
y_pred = best_clf.predict(rescaledX)
print("\nPrecisión en los datos de entrenamiento:")
print(accuracy_score(Y_train, y_pred))
```

```
Mejores parámetros encontrados:
{'algorithm': 'auto', 'metric': 'euclidean', 'n_neighbors': 9, 'weights': None}

Mejor puntuación obtenida:
0.5226176098344706

Precisión en los datos de entrenamiento:
0.9729152509465101
```

La mejor puntuación obtenida mediante validación cruzada (aproximadamente 52.62%) es una estimación más realista del rendimiento de generalización del modelo. Este valor es ligeramente superior a la mediana observada para Scaledk-NN en el análisis comparativo de algoritmos escalados anterior, lo que indica que la optimización ha permitido encontrar una configuración de hiperparámetros que mejora el desempeño promedio del algoritmo. La puntuación más alta se consiguió con los siguientes hiperparámetros:

- n_neighbors: 9
- weights: None

- algorithm: auto
- metric: euclidean

También hay una discrepancia significativa entre la precisión obtenida en el conjunto de entrenamiento (97.29%) y la precisión promedio obtenida en la validación cruzada (52.62%). Esta gran diferencia es un fuerte indicio de sobreajuste (overfitting). El modelo con los hiperparámetros óptimos memoriza muy bien los datos de entrenamiento, pero no generaliza de manera efectiva a datos no vistos.

4.1.2 Optimización para el algoritmo ensamblado GBM

Dado el potencial de los algoritmos ensamblados, principalmente GBM que mostró una mediana de precisión prometedora en las comparativas iniciales, se realizará un ajuste de los siguientes hiperparámetros con la finalidad de encontrar el más optimo en cada uno de ellos.

- n_estimators: (Valor default= 100)
- learning_rate: (Valor default= 0.1)
- max_depth (Valor default= 3)
- min_samples_split (Valor default= 100)
- min_samples_leaf (Valor default= 2)
- subsample (Valor default= 1)

La optimización se realizó utilizando la técnica de búsqueda en cuadrícula con validación cruzada (GridSearchCV), al igual que con KNN. Se mantuvo la estrategia de Validación Cruzada por Grupos (GroupKFold) con n_splits=3, para asegurar que la evaluación de la generalización del modelo fuera robusta y no se viera afectada por la posible correlación entre observaciones del mismo motor (MTR). También, las características del conjunto de entrenamiento (X_train) fueron escaladas con StandardScaler. Aunque los algoritmos basados en árboles como GBM no son tan sensibles a la escala de las características como los basados en distancia, mantener una

consistencia en el preprocesamiento es una buena práctica y puede ser beneficioso cuando se combinan con otros modelos o en ciertas implementaciones. Luego, se configuro GridSearchCV con los campos estimator, param_grid, GroupKFold(n_splits=3) y scoring. Finalmente, Se ejecutó grid.fit(rescaledX, Y_train, groups=groups) para realizar la búsqueda en cuadrícula sobre el conjunto de entrenamiento escalado (rescaledX) y las etiquetas de entrenamiento (Y_train), utilizando los grupos definidos. El proceso involucró el ajuste de 486 combinaciones de parámetros, resultando en un total de 1458 ajustes de modelos (486 combinaciones * 3 pliegues).

Figura 41:

Optimización del algoritmo GBM

```
scaler = StandardScaler().fit(X_train)
rescaledX = scaler.transform(X_train)
param_grid = {
    'n_estimators': [100, 200, 300,400,500],              # Número de árboles
    'learning_rate': [0.001, 0.01, 0.1, 0.2],            # Tasa de aprendizaje
    'max_depth': [3, 5, 7],                          # Profundidad máxima de los árboles
    'min_samples_split': [2, 5, 7, 10],                 # Mínimas muestras necesarias para dividir un nodo
    'min_samples_leaf': [1, 3, 5],                   # Mínimas muestras por hoja
    'subsample': [0.7,0.8, 1.0]                         # Fracción de muestras utilizadas para cada árbol
}
model = GradientBoostingClassifier()
group_kfold=GroupKFold(n_splits=3)# hace una iteracion de datos
grid = GridSearchCV(estimator=model, param_grid=param_grid, cv=group_kfold,scoring=scoring,verbose=2,n_jobs=-1)
grid_result = grid.fit(rescaledX, Y_train,groups=groups)


print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_))
means = grid_result.cv_results_['mean_test_score']
stds = grid_result.cv_results_['std_test_score']
params = grid_result.cv_results_['params']
for mean, stdev, param in zip(means, stds, params):
    print(f"{param}: {mean*100.0:,.2f}% ({stdev*100.0:,.2f}%)")
```

```
Fitting 3 folds for each of 2160 candidates, totalling 6480 fits
Best: 0.560531 using {'learning_rate': 0.1, 'max_depth': 7, 'min_samples_leaf': 3, 'min_samples_split': 5, 'n_estimators': 4
00, 'subsample': 0.8}
```

Al finalizar el proceso de optimización, podemos encontrar que la configuración de parámetro más optimo y que muestra el valor más alto de accuracy (56.1%) es resultado de aplicar los siguientes hiperparámetros:

- n_estimators: = 40
- learning_rate = 0.1
- max_depth = 7

- min_samples_split = 5
- min_samples_leaf = 3
- subsample = 0.8

Se observa que el algoritmo que brinda un mejor desempeño para nuestra base de datos es el algoritmo ensamblado y optimizado GBM. Los resultados también indican que, con una configuración adecuada, GBM es un algoritmo altamente prometedor para la tarea de predicción de fallas en este dominio. Su capacidad para combinar múltiples modelos débiles en un clasificador fuerte es evidente en la mejora de la precisión.

4.2 Evaluación del modelo en el conjunto de validación

Tras la optimización de hiperparámetros del algoritmo Gradient Boosting Machine (GBM) en el conjunto de entrenamiento mediante GridSearchCV y GroupKFold, el siguiente paso crítico fue evaluar el rendimiento del modelo con la mejor configuración encontrada en un conjunto de datos completamente no visto. Este conjunto de validación (o prueba) es un ejemplo más preciso de como el modelo se desempeñaría en un escenario real.

Para la evaluación final, se utilizó el modelo GradientBoostingClassifier con los parámetros óptimos hallados anteriormente, Es importante señalar que estos parámetros podrían diferir ligeramente de los mejores encontrados en el GridSearchCV previo, debido a que van a ser utilizados dentro de una iteración diferente al del proceso de optimización.

El procedimiento de evaluación será el siguiente:

- **Escalado del Conjunto de Validación:** El conjunto de características de validación (X_test) fue escalado utilizando el mismo StandardScaler que se ajustó previamente con los datos de entrenamiento. Es fundamental aplicar la misma transformación (ajustada sobre los datos de entrenamiento) a los datos de validación para mantener la consistencia en la escala.

- **Generación de Predicciones:** El modelo GradientBoostingClassifier entrenado con los parámetros seleccionados model.fit(rescaledX, Y_train), se utilizó para generar predicciones sobre el conjunto de validación escalado (rescaledValidationX).
- **Cálculo de Métricas de Rendimiento:** e calcularon las siguientes métricas clave para evaluar el desempeño del modelo:
    - Precisión (*Accuracy*): La proporción de predicciones correctas sobre el total de predicciones.
    - Matriz de Confusión: Una tabla que resume el número de predicciones correctas e incorrectas para cada clase, mostrando los verdaderos positivos (TP), verdaderos negativos (TN), falsos positivos (FP) y falsos negativos (FN).
    - Reporte de Clasificación (*Classification Report*): Proporciona métricas más detalladas para cada clase, incluyendo:
        - Precisión (*Precision*): La proporción de verdaderos positivos entre todas las predicciones positivas hechas por el modelo (TP / (TP + FP)). Relevante para minimizar falsos positivos.
        - Sensibilidad (*Recall*): La proporción de verdaderos positivos entre todas las instancias positivas reales (TP / (TP + FN)). Crucial para detectar la mayor cantidad posible de eventos de "falla".
        - Puntuación F1 (*F1-score*): La media armónica de la precisión y la sensibilidad, que proporciona un balance entre ambas métricas, especialmente útil en clases desbalanceadas.
        - Soporte (*Support*): El número de ocurrencias reales de cada clase en el conjunto de validación.

Figura 42:

Resultado de la aplicación del GBM optimizado en los datos de validación

```
# prepare the model
scaler = StandardScaler().fit(X_train)
rescaledX = scaler.transform(X_train)
model = model = GradientBoostingClassifier(learning_rate=0.1,max_depth=7,min_samples_leaf=3,min_samples_split=5,n_estimators=40,


model.fit(rescaledX, Y_train)
# estimate accuracy on validation dataset
start_time = time.time()
rescaledValidationX = scaler.transform(X_test)
predictions = model.predict(rescaledValidationX)
end_time = time.time()
dif_time= end_time - start_time
print(accuracy_score(Y_test, predictions))
print(confusion_matrix(Y_test, predictions))
print(classification_report(Y_test, predictions))
print(f'T. Validacion en seg: {dif_time:.4f}')
```

```
0.5461386138613862
[[699 519]
 [627 680]]
              precision    recall  f1-score   support

           0       0.53      0.57      0.55      1218
           1       0.57      0.52      0.54      1307

    accuracy                           0.55      2525
   macro avg       0.55      0.55      0.55      2525
weighted avg       0.55      0.55      0.55      2525

T. Validacion en seg: 0.0080
```

La precisión global del modelo GBM en el conjunto de validación es de aproximadamente 55%. Aunque esta cifra es la más alta observada en las etapas de validación hasta ahora, sigue siendo una precisión moderada, lo que indica que el modelo tiene dificultades para clasificar correctamente cerca de la mitad de las instancias. Además, para la clase 0 ("No Falla"), la precisión es de 0.53 y *recall* de 0.57, el modelo predice correctamente el 57% de las instancias de "No Falla" y, de las veces que predice "No Falla", acierta en el 53%. Para el caso de la Clase 1 (“Falla”) la precisión de 0.57 y *recall* de 0.52. Esto significa que el modelo es relativamente bueno cuando predice una "Falla" (acierta el 57% de las veces), pero solo logra identificar el 52% de las fallas reales presentes en el conjunto de validación. Además, la sensibilidad (recall) para la clase "Falla" (Clase 1) es de suma importancia. Un recall del 0.52 significa que el modelo solo detecta aproximadamente la mitad de las fallas reales que ocurren. El no detectar una falla (falso negativo) puede tener consecuencias operativas y económicas significativas.

En conclusión, el modelo Gradient Boosting Machine optimizado demuestra una capacidad moderada para predecir fallas en motores diésel, La decisión de la utilidad final del modelo dependerá de la priorización de los costos de falsos negativos (no detectar una falla, lo cual es grave) frente a los costos de falsos positivos (alertas innecesarias). Para aplicaciones de mantenimiento predictivo, a menudo se prefiere un mayor recall para la clase de falla, incluso a expensas de una mayor tasa de falsos positivos. Los resultados sugieren que este modelo GBM es un paso significativo hacia una detección proactiva de fallas, pero se requeriría una optimización adicional o estrategias de post-procesamiento para manejar el trade-off entre falsos positivos y falsos negativos según los objetivos específicos del negocio.

4.3 Validación de las Hipótesis

- Hipótesis General: El uso de algoritmos de aprendizaje para la predicción de fallas de motor ayudará a mejorar la predictibilidad de falla de un equipo de acarreo de gran minería.

  Validación: Los resultados obtenidos a lo largo de este trabajo, particularmente la capacidad del modelo Gradient Boosting Machine (GBM) optimizado para clasificar eventos en el conjunto de validación con una precisión global de aproximadamente 55% y métricas balanceadas entre clases (F1-score de 0.55 para la clase 0 y 0.54 para la clase 1), validan la hipótesis general. Debido a que actualmente se cuenta solo con una detectabilidad del 44% en el 2024 y del 33% en el 2025 con el método estadístico actual. La habilidad de un modelo de aprendizaje automático para identificar patrones en los datos de monitoreo de condición permite anticipar la ocurrencia de una falla. Esta capacidad de predicción facilita la implementación de un mantenimiento predictivo, lo cual se traduce en una planificación más eficiente de las intervenciones, una reducción significativa de los tiempos de inactividad no planificados y, en última instancia, una mejora sustancial en la disponibilidad y eficiencia operativa del equipo de acarreo.

- Hipótesis Secundaria 1: Un adecuado preprocesamiento de los datos, que incluya limpieza, normalización y selección de variables relevantes, mejora significativamente la precisión de los modelos de predicción de fallas de motores en equipo de acarreo de gran minería

  Validación: Esta hipótesis se valida plenamente a través de las diversas etapas de preprocesamiento aplicadas en el estudio. La verificación inicial de tipos de datos (float64 e int64) y la conversión o eliminación de valores no numéricos (NaN) aseguró la integridad y la correcta interpretación de las variables, sentando las bases para análisis y modelado precisos. Además, el análisis de histogramas reveló las distribuciones asimétricas de muchas variables, y el mapa de calor de correlación permitió comprender las relaciones entre ellas. Estos análisis son parte del preprocesamiento que informa sobre la estructura de los datos. La mejora en el rendimiento del GBM optimizado con respecto a las líneas base, es un testimonio de cómo un preprocesamiento cuidadoso contribuye a un mejor desempeño del modelo.

- Hipótesis Secundaria 2: Entre los algoritmos de aprendizaje supervisado aplicados a la base de datos disponible, aquellos de tipo ensamblado presentan una mejor performance predictiva y una relación más eficiente entre precisión y tiempo de ejecución, en comparación con algoritmos lineales y no lineales.

  Validación: Esta hipótesis es validada por los resultados comparativos, ya que el algoritmo GradientBoostingMachine (GBM) optimizado, un método ensamblado, alcanzó la mayor precisión de validación cruzada promedio (55.21%) durante la fase de optimización de hiperparámetros, superando consistentemente a los algoritmos individuales evaluados. Esto demuestra la superioridad de los métodos de ensemble para combinar las fortalezas de múltiples modelos y mejorar la robustez predictiva a comparación de los algoritmos lineales y no lineales.

- Hipótesis Secundaria 3: El uso de modelos de aprendizaje automático supervisado permite estimar con mayor precisión el porcentaje de fallas de motores de camiones mineros en un horizonte de tiempo determinado, en comparación con los métodos estadísticos tradicionales

  Validación: Los resultados de esta investigación respaldan esta hipótesis, El modelo de aprendizaje automático, específicamente el GBM, ha demostrado la capacidad de procesar y extraer patrones predictivos de un conjunto multifactorial de datos como son el análisis de aceite, contadores VIMS, horas de operación, etc. La integración de estas características es una ventaja importante sobre los métodos estadísticos actual, debido a que estos normalmente se limitan a un menor numero de variables. Dentro del Classification Report del modelo GBM se observan métricas separadas por clase, lo que permite una estimación mucho mas precisa en la detección de fallas.

En síntesis, este trabajo desmuesta la viabilidad de los algoritmos de aprendizaje automático para la predicción de fallas en motores de equipos mineros, además cimenta la bases para futuros proyectos de mantenimiento predictivo que mejoraran la disponibilidad y reducirán los costos operativos.

4.4 Discusión de resultados

Los resultados han presentado una serie de hallazgos derivados del análisis exploratorio de datos, la comparativa de algoritmos de aprendizaje supervisado y la evaluación del modelo optimizado de Gradient Boosting Machine (GBM) para la predicción de fallas de motor en equipos de acarreo de gran minería. Esta sección se dedicará a interpretar estos resultados en un contexto más amplio, discutir sus implicaciones, reconocer las limitaciones del estudio y sugerir futuras líneas de investigación

4.4.1 Interpretación de los Hallazgos Clave

- Características de los Datos y su Relevancia para la Predicción: El análisis exploratorio de datos reveló características distintivas del conjunto de datos de monitoreo de condición. La presencia de distribuciones altamente sesgadas a la derecha en la mayoría de las variables de análisis de aceite (metales de desgaste, contaminantes) y contadores de eventos VIMS, es un hallazgo consistente con la naturaleza de los datos de falla de maquinaria. En condiciones normales de operación, los niveles de desgaste y la ocurrencia de eventos anómalos son bajos. Los valores atípicos y las colas extendidas en estas distribuciones son precisamente las señales de interés para la predicción de fallas, ya que representan desviaciones del comportamiento esperado. Esta asimetría muestra la necesidad de un preprocesamiento adecuado, como el escalado, que fue fundamental para el rendimiento de varios algoritmos.
- Impacto del Preprocesamiento en el Rendimiento del Modelo: La normalización de las características mediante StandardScaler fue crucial, especialmente para algoritmos basados en distancia como KNN y SVM, así como para mejorar la estabilidad de otros modelos. Este hallazgo se encuentra relacionado con la información sobre aprendizaje automático, donde se menciona que es un requisito importante el escalamiento de los datos, cuando estos tienen rangos de valores muy diferentes. (Géron, 2019). Sin un escalado adecuado, los algoritmos pueden asignar una relevancia desproporcionada a algunos valores, afectando la capacidad del modelo de aprender patrones.
- Superioridad de los Algoritmos Ensamblados: En general, los resultados de los algoritmos confirmaron que los ensamblados,particularmente Gradient Boosting Machine (GBM) y Random Forest (RF),mostraron un mejor rendimiento y mas consistente en comparación con los métodos lineales y no lineales. Esto también se ve reflejado en la literatura donde se menciona a los algoritmos ensamblados

como algoritmos con mejor desempeño en una amplia gama de problemas de clasificación y regresión.

4.4.2 Conexión con la Literatura y Aportes

Los resultados obtenidos en este trabajo se encuentran acorde a la creciente evidencia sobre el potencial del aprendizaje automático en el mantenimiento predictivo (Géron, 2019). La confirmación de que los métodos ensamblados, principalmente el de tipo Boosting, son efectivos para problemas de clasificación con datos complejos, en una prueba más de su posición como algoritmo predilecto en la industria actual. Además, el uso de GroupKFold para la validación cruzada muestra una limitación en los estudios que utilizan datos de series temporales o de equipos específicos, demostrando una evaluación mas realista del modelo.

4.4.3 Limitaciones del Estudio

- Disponibilidad de Datos: La calidad, cantidad y representatividad de los datos son fundamentales. Aunque la base de datos utiliza es considerable grande, la ausencia de variables como condiciones operacionales, clima, información de mantenimiento como relleno de aceite o cambio de filtros podría limitar la capacidad predictiva del modelo
- Interpretación de Fallas La definición de "falla" como variable objetivo (Clase 1) es importante. Debido a que la etiqueta se basa en horas previas a los eventos registrados de falla, pero no es la verdadera falla en si, por lo que podría generar variaciones en el resultado.
- Incluir variables más técnicas durante el entrenamiento de datos: Con un mayor entendimiento técnico de las variables que afectan al funcionamiento del motor, se puede lograr una mejor precisión del modelo

4.4.4 Líneas Futuras de Investigación

- Exploración de Métricas Adicionales: Realizar una evaluación exhaustiva del modelo utilizando métricas más allá de la precisión, como el área bajo la curva ROC (AUC), la curva de precisión-recall, y métricas de costo-sensibilidad que reflejen el impacto económico de falsos positivos y falsos negativos.
- Ingeniería de Características: Desarrollar nuevas características a partir de los datos existentes, como tasas de cambio (derivadas), desviaciones de la media móvil, o indicadores de tendencia de cambio de aceite o filtros, que podrían capturar mejor los patrones de deterioro del motor.
- Modelos de Aprendizaje Profundo: Explorar la aplicación de arquitecturas de aprendizaje profundo, como redes neuronales recurrentes (RNNs) o redes de atención, que son especialmente aptas para modelar dependencias temporales en datos de series de tiempo.
- Validación en Tiempo Real y Despliegue: Probar el modelo en un entorno de producción o simulación en tiempo real para evaluar su robustez y escalabilidad, e integrar sus predicciones en un sistema de alertas o un tablero de control para el personal de mantenimiento.

Conclusiones

- El estudio ha demostrado de manera concluyente la viabilidad y el alto potencial de los algoritmos de aprendizaje automático para predecir fallas en componentes críticos de equipos de gran minería, como los motores de camiones Caterpillar 797F. La capacidad de anticipar estos eventos permite una transición efectiva de un modelo de mantenimiento reactivo a uno predictivo, lo que se traduce directamente en una mejora sustancial de la disponibilidad del equipo y una reducción de los costos operativos.
- La limpieza, la verificación de tipos de datos, el escalado de características, especialmente para algoritmos sensibles a la escala, y un análisis exploratorio detallado usando histogramas y correlaciones no solo son pasos metodológicos esenciales, sino que impactan directamente y de manera significativa la precisión y la robustez de los modelos predictivos.
- La comparativa de algoritmos reveló que los métodos de aprendizaje ensamblado, como Gradient Boosting Machine (GBM), ofrecen un rendimiento predictivo superior y más consistente en comparación con los algoritmos individuales. El GBM optimizado alcanzó la mayor precisión de validación entre todos los modelos evaluados (55% de precisión promedio en validación cruzada) y demostró ser el más prometedor para la tarea de predicción de fallas.
- El modelo GBM optimizado no solo ofrece un rendimiento prometedor en la detección de fallas, sino que también es computacionalmente eficiente (tiempo de validación de 0.0080 segundos). Esta eficiencia es fundamental para su integración en sistemas de monitoreo de condición en tiempo real, por lo que es factible poder aplicarlo en el sistema de mantenimiento actual ya que cuenta con una mejor precisión que los sistemas predictivos actuales basados en estadística.

Recomendaciones

Se Recomienda para futuras investigaciones:

- Además de la precisión global y el F1-score, evaluar el modelo utilizando métricas de costo-sensibilidad que reflejen el impacto económico real de los falsos positivos (alertas innecesarias y gastos de inspección) y los falsos negativos (fallas no detectadas que conllevan altos costos de reparación y tiempo de inactividad). Esto permitirá una selección de modelo más alineada con los objetivos de negocio de la industria minera.
- Dedicar esfuerzos significativos a la ingeniería de características. Explorar la creación de nuevas variables a partir de los datos existentes también incluir un mejor diseño para filtrar variables que estén correlacionadas entre si.
- Dado que los datos de monitoreo son inherentemente series temporales, se sugiere explorar arquitecturas de aprendizaje profundo como las Redes Neuronales Recurrentes (RNNs), LSTMs (Long Short-Term Memory) o modelos basados en transformadores. Estos modelos son especialmente aptos para capturar dependencias temporales y patrones secuenciales en los datos, lo que podría mejorar significativamente la precisión predictiva para eventos dependientes del historial.
- Considerar la inclusión de otras fuentes de datos que no fueron exploradas en profundidad en este estudio, Datos de Operación, Registros de Mantenimiento como cambio de aceite y cambios de filtros, Información del Operado o mejoras de fábrica pendientes de aplicar

Referencias bibliográficas

Academy, K. (s.f.). *Descenso de gradiente.*

ADAUTO ARANA, R. M. (2021). Aplicación de la inteligencia artificial en la detección de fallas en los motores eléctricos de corriente continua de imán permanente. *Universidad Nacional del Centro del Perú.*, 4-6.

Akarte, M. M. (2018). Predictive Maintenance of Air Pressure System using Boosting Trees: A Machine Learning Approach. *Department of Industrial Engineering and Operations Research Indian Institute of Technology Bombay*, 4.

Alarie, S., & Gamache, M. (2002). Overview of solution strategies used in truck dispatching systems for open pit mines. *International Journal of Surface International Journal of Surface Environment*, 59-76.

Alaswad, S., & Xlang, Y. (2017). A review on condition-based maintenance optimization models for stochastically deteriorating system. 54-63.

Al-Shalabi, L., & Shaaban, Z. (2006). Normalization as a Preprocesing . *Applied Science University*, 2.

Balemir , U., & Ramesh , R. (2011). Developing an appropriate data normalization method.

Bloch,, H., & Geitner, F. (2006). Machinery Failure Analysis and Troubleshooting: Practical Machinery Management for Process Plants. *Gulf Professional Publishing.*

Brownlee, A. E., Wright, J., He, M., & Timothy, L. (2020). A novel encoding for separable large-scale multi-objetive problems and its application to the optimisation of housing stock improvements. *Applied Soft Computing Journel*, 9.

Carvalho, T. P. (2019). A systematic literature review of machine learning methods applied to. *Elsevier*, 1-2.

CAT. (2022). *CAT Products*. Obtenido de CAT.com: https://www.cat.com/es_US/products/new/equipment/off-highway-trucks/mining-trucks/18093014.html

Caterpillar. (2018). VIMS 3G Operators Manual. *CAT*.

Caterpillar. (2019). Caterpillar Administración de equipos de minería. *Mediciones de rendimiento*, 5-10.

Caterpillar. (2020). An Introduction to Product Link & VisionLink (SEGV2607-01).

Caterpillar. (2024). 797F Mining Truck.

Choi, Y., Hoang, N., Xuan-Nam, B., & Trung Nguyen, T. (2020). Estimating Ore Production in Open-pit Mines Using Various Machine Learning Algorithms Based on a Truck-Haulage System and Support of Internet of Things. *Natural Resources Research*, 2.

codificandobits. (2021). *La Regresión Lineal en el Machine Learning*. Obtenido de https://codificandobits.com/blog/regresion-lineal/

Corporation, N. (s.f.). Oil Analysis. Machinery Lubrication.

Cunningham, P., Cord, M., & Delany, S. J. (s.f.). Supervised Learning. *Dublin Institute of Technology*, 21.

El Naqa, I., & Murphy, M. J. (2015). What Is Machine Learning? *Springer International Publishing Switzerland* , 1.

ESRI. (2021). *Estandarizar campo (Administración de datos)*. Obtenido de ESRI: https://doc.arcgis.com/es/allsource/1.1/analysis/geoprocessing-tools/data-management/standardizefield.htm

Ferreyros, H. (Diciembre de 2019). *Ferreyros CAT*. Obtenido de https://www.ferreyros.com.pe/nosotros/acerca-de-ferreyros/historia/

Finning. (s.f). Motor C175-20 Acert. *Manual del estudiante*, 14-33.

Fitch, J. C. (2007). Oil Analysis Basics. *Noria Corporation.*

Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems.

Ghahramani, Z. (2004). Unsupervised Learning. *Gatsby Computational Neuroscience Unit*, 2.

Gonzales, J. L. (2021). *Diseño y Metodologia de la Investigación.* Arequipa: Biblioteca Nacional del Perú.

Herrera Zeballos, P. A. (2021). Método de gestión de mantenimiento centrado en la confiabilidad para mejorar la disponibilidad de los motores c175-16 en la flota 793f del proyecto minero Constancia.

IBM. (s.f.). *What is Gradient Descent?*

Jijo, B. T., & Adnan Mohsin, A. (2021). Classification Based on Decision Tree Algorithm for Machine Learning.

Juanli, l., Shuo, J., Menghui, L., & Jiacheng, X. (29 de ENERO de 2020). A Fault Diagnosis Method of Mine Hoist Disc Brake System Based on Machine Learning. Taiyuan, China: Applied sciences.

Larico, A. (2021). Mejora de la confiabilidad en el sistema de combustible del motor de combustión interna diésel C175-20 de un camión minero 797F. *Universidad Continental, Huancayo*.

Mao, W. (2016). Online sequential prediction of bearings imbalanced fault. *Elsevier*, 2-8.

Martin-Diaz, I. (2020). An Experimental Comparative Evaluation of Machine Learning Techniques for Motor Fault Diagnosis Under Various Operating Conditions. *IEEE*, 3-5.

Mobley, R. K. (2012). An Introduction to Predictive Maintenance. *Butterworth-Heinemann*.

Naranjo, M. (s.f.). Análisis predictivo de activos mineros para obtención de intervalo de falla mediante algoritmos de machine learning.

Orrù, P. F. (2020). Machine Learning Approach Using MLP and SVM. *Sustainability*, 1-4.

Provost, F. (2013). Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking. *O'Reilly Media*.

Ray, S. (2019). A Quick Review of Machine Learning Algorithms. *Manav Rachna University* , 35.

Reddy Alla, H., Hall, R., & Apel, D. (2019). Performance evaluation of near real-time condition monitoring. 2.

RPM-GLOBAL. (2022). *RPM GLOBAL AMT*. Obtenido de https://rpmglobal.com/es/product/amt/

Saeys, Y., Iñaki, I., & Larrañaga, P. (2007). A review of feature selection techniques in bioinform. *Bioinformatics*, 2507-2510.

Schofield, C. (2010). Condition Monitoring: Techniques and Methodology. *SAE International*.

Shmueli, K. (2011). Predictive Analytics in Information Systems Research.

Sperande, S. (2014). analysis, Understanding logistic regression. *School of Physical Education and Sports - Federal University of Rio de Janeiro, Rio de Janeiro, Brazil*.

T. Larose, D., & D. Larose., C. (2014). Discovering knowledge in data: An introduction to data mining. 149–164.

Valencia, O. (s.f.). Evaluación del proceso de operación y mantenimiento preventivo - correctivo de un Motor Diésel de 2500 HP de uso minero. *Souther Peru Copper Corporation*.

VanderPlas, J. (2016). Python Data Science Handbook: Essential Tools for Working with Data.

Wei, G. (2018). Reliability modeling with condition-based maintenance for binary-state. *Elsevier*, 1-2.

Westbrook, T. (2013). Fuels and Lubricants Handbook: Technology, Properties, Performance, and Testing. *ASTM International.*

Ya, S., Xinbo, G., & Xuelong, L. (2012). Multivariate multilinear regression. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics),.

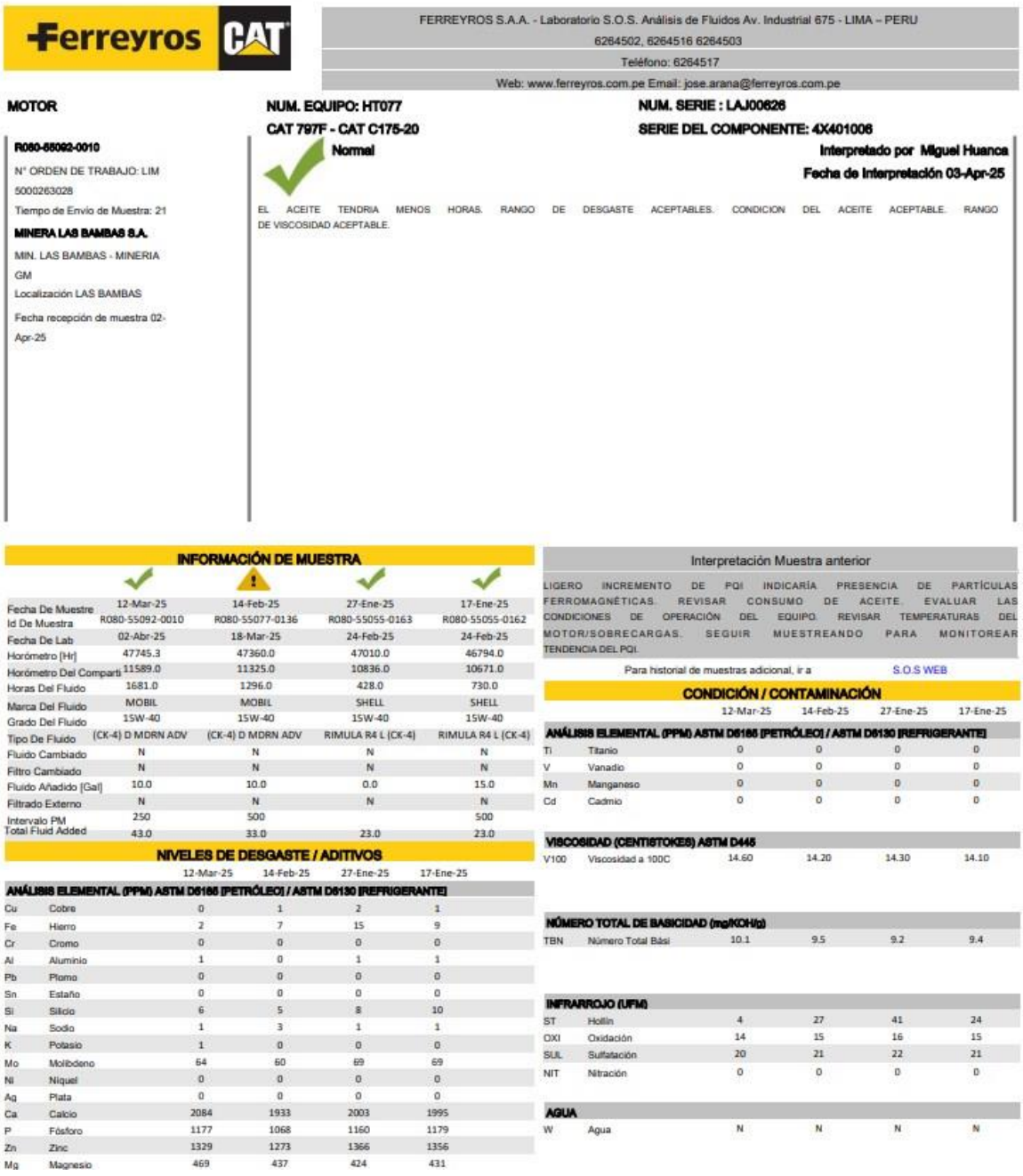
Yeh, C.-H. (2019). Machine Learning for Long Cycle Maintenance. *Sensors*, 1-10.

Zambrano, C., & Perez, G. (2021). Estudio de la aplicación del mantenimiento predictivo en motores diésel en la provincia de Manabí.

Zhang, Y. (2023). Support Vector Machine Classification Algorithm.

Anexos

Anexo 1: Ejemplo de análisis SOS



Ferreyros CAT

FERREYROS S.A.A. - Laboratorio S.O.S. Análisis de Fluidos Av. Industrial 675 - LIMA – PERU
6264502, 6264516 6264503
Teléfono: 6264517
Web: www.ferreyros.com.pe Email: jose.arana@ferreyros.com.pe

MOTOR

R080-55092-0010

N° ORDEN DE TRABAJO: LIM 5000263028

Tiempo de Envio de Muestra: 21

MINERA LAS BAMBAS S.A.

MIN. LAS BAMBAS - MINERIA GM

Localización LAS BAMBAS

Fecha recepción de muestra 02-Apr-25

NUM. EQUIPO: HT077

CAT 797F - CAT C175-20

Normal

EL ACEITE TENDRIA MENOS HORAS. RANGO DE DESGASTE ACEPTABLES. CONDICION DEL ACEITE ACEPTABLE. RANGO DE VISCOSIDAD ACEPTABLE.

NUM. SERIE : LAJ00626

SERIE DEL COMPONENTE: 4X401006

Interpretado por Miguel Huanca

Fecha de Interpretación 03-Apr-25

INFORMACIÓN DE MUESTRA

Fecha De Muestre	12-Mar-25	14-Feb-25	27-Ene-25	17-Ene-25
Id De Muestra	R080-55092-0010	R080-55077-0136	R080-55055-0163	R080-55055-0162
Fecha De Lab	02-Abr-25	18-Mar-25	24-Feb-25	24-Feb-25
Horómetro [Hr]	47745.3	47360.0	47010.0	46794.0
Horómetro Del Comparti	11589.0	11325.0	10836.0	10671.0
Horas Del Fluido	1681.0	1296.0	428.0	730.0
Marca Del Fluido	MOBIL	MOBIL	SHELL	SHELL
Grado Del Fluido	15W-40	15W-40	15W-40	15W-40
Tipo De Fluido	(CK-4) D MDRN ADV	(CK-4) D MDRN ADV	RIMULA R4 L (CK-4)	RIMULA R4 L (CK-4)
Fluido Cambiado	N	N	N	N
Filtro Cambiado	N	N	N	N
Fluido Añadido [Gal]	10.0	10.0	0.0	15.0
Filtrado Externo	N	N	N	N
Intervalo PM	250	500		500
Total Fluid Added	43.0	33.0	23.0	23.0

NIVELES DE DESGASTE / ADITIVOS

		12-Mar-25	14-Feb-25	27-Ene-25	17-Ene-25
ANÁLISIS ELEMENTAL (PPM) ASTM D6186 [PETRÓLEO] / ASTM D6130 [REFRIGERANTE]					
Cu	Cobre	0	1	2	1
Fe	Hierro	2	7	15	9
Cr	Cromo	0	0	0	0
Al	Aluminio	1	0	1	1
Pb	Plomo	0	0	0	0
Sn	Estaño	0	0	0	0
Si	Silicio	6	5	8	10
Na	Sodio	1	3	1	1
K	Potasio	1	0	0	0
Mo	Molibdeno	64	60	69	69
Ni	Niquel	0	0	0	0
Ag	Plata	0	0	0	0
Ca	Calcio	2084	1933	2003	1995
P	Fósforo	1177	1068	1160	1179
Zn	Zinc	1329	1273	1366	1356
Mg	Magnesio	469	437	424	431

Interpretación Muestra anterior

LIGERO INCREMENTO DE PQI INDICARÍA PRESENCIA DE PARTÍCULAS FERROMAGNÉTICAS. REVISAR CONSUMO DE ACEITE. EVALUAR LAS CONDICIONES DE OPERACIÓN DEL EQUIPO. REVISAR TEMPERATURAS DEL MOTOR/SOBRECARGAS. SEGUIR MUESTREANDO PARA MONITOREAR TENDENCIA DEL PQI.

Para historial de muestras adicional, ir a S.O.S WEB

CONDICIÓN / CONTAMINACIÓN

		12-Mar-25	14-Feb-25	27-Ene-25	17-Ene-25
ANÁLISIS ELEMENTAL (PPM) ASTM D6186 [PETRÓLEO] / ASTM D6130 [REFRIGERANTE]					
Ti	Titanio	0	0	0	0
V	Vanadio	0	0	0	0
Mn	Manganeso	0	0	0	0
Cd	Cadmio	0	0	0	0
VISCOSIDAD (CENTISTOKES) ASTM D445					
V100	Viscosidad a 100C	14.60	14.20	14.30	14.10
NÚMERO TOTAL DE BASICIDAD (mgKOH/g)					
TBN	Número Total Básí	10.1	9.5	9.2	9.4
INFRARROJO (UFM)					
ST	Hollín	4	27	41	24
OXI	Oxidación	14	15	16	15
SUL	Sulfatación	20	21	22	21
NIT	Nitración	0	0	0	0
AGUA					
W	Agua	N	N	N	N

Anexo 2: Ejemplo de reporte de VIMS

Cat Electronic Technician 2023A v1.0

Product Status Report

19/04/2025 20:34

Product Status Report

Parameter	Value
Product ID	LAJ00616
Equipment ID	HT074
Comments	

VIMS Main Module (LAJ00616)

Parameter	Value
Product ID	LAJ00616
Equipment ID	HT074
ECM Part Number	2851142-05
ECM Serial Number	0178B317HD
Software Group Part Number	6265042-00
Software Group Release Date	FEB22
Software Group Description	MAIN_2021B_PROD_1.0
Application Software Part Number	6325828-00
Dealer Identification Code	Unavailable
DBS Machine Make Code	Unavailable
Wireless Transmission Device Serial Number	Unavailable

Logged Diagnostic Codes [Diagnostic Clock = 52939 hours] - VIMS Main Module (LAJ00616)

Code	Description	Occ.	First	Last
590- 9	Engine Control Module : Abnormal Update Rate	13	26467	52781
246-14	Proprietary CAN Data Link : Special Instruction	127	45447	52260
890- 9	Telemetry Data Link : Abnormal Update Rate	101	30039	51479
1273- 9	Chassis Control Module : Abnormal Update Rate	5	33303	45672
296- 9	Transmission Control : Abnormal Update Rate	4	33303	45671
246- 9	Proprietary CAN Data Link : Abnormal Update Rate	4	37322	45607
800-11	VIMS Main Module : Other Failure Mode	1	41416	41416
168- 1	Electrical System Voltage : Low	3	32029	34794
533- 9	Brake Control : Abnormal Update Rate	1	33303	33303

Logged Event Codes [Diagnostic Clock = 52939 hours] - VIMS Main Module (LAJ00616)

Code	Description	Occ.	First	Last
E558 (1)	Snapshot Stored	127	6561	52901
E072 (2)	Oil Level Low Mark	7	27553	52887
E1189 (1)	File System Memory Low	127	30112	52820
E1189 (2)	File System Memory Low	127	30112	52820