UNIVERSIDAD NACIONAL DEL ALTIPLANO

FACULTAD DE INGENIERIA MECÁNICA ELÉCTRICA, ELECTRONICA Y SISTEMAS

ESCUELA PROFESIONAL DE INGENIERIA MECÁNICA ELÉCTRICA

MANTENIMIENTO PREDICTIVO USANDO ALGORITMOS DE MACHINE LEARNING APLICADO A BOMBAS DE AGUA

TESIS

PRESENTADA POR:

Bach. ALDAIR RODRIGO BENAVENTE

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO MECÁNICO ELÉCTRICO

PUNO – PERÚ

2024

ALDAIR RODRIGO BENAVENTE

MANTENIMIENTO PREDICTIVO USANDO ALGORITMOS DE MACHINE LEARNING APLICADO A BOMBAS DE AGUA

Universidad Nacional del Altiplano

Detalles del documento

Identificador de la entrega
trn:oid:::8254:416594655

Fecha de entrega
16 dic 2024, 12:19 p.m. GMT-5

Fecha de descarga
16 dic 2024, 12:23 p.m. GMT-5

Nombre de archivo
MANTENIMIENTO PREDICTIVO USANDO ALGORITMOS DE MACHINE LEARNING APLICADO A BO....docx

Tamaño de archivo
9.1 MB

131 Páginas

20,394 Palabras

115,579 Caracteres

turnitin Página 1 of 142 - Portada Identificador de la entrega trn:oid:::8254:416594655

13% Similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para ca...

Filtrado desde el informe

- Bibliografía
- Texto citado
- Texto mencionado
- Coincidencias menores (menos de 10 palabras)

Fuentes principales

9% Fuentes de Internet
3% Publicaciones
9% Trabajos entregados (trabajos del estudiante)

Marcas de integridad

N.° de alertas de integridad para revisión

No se han detectado manipulaciones de texto sospechosas.

Los algoritmos de nuestro sistema analizan un documento en profundidad para buscar inconsistencias que permitirían distinguirlo de una entrega normal. Si advertimos algo extraño, lo marcamos como una alerta para que pueda revisarlo.

Una marca de alerta no es necesariamente un indicador de problemas. Sin embargo, recomendamos que preste atención y la revise.

JOSE MANUEL RAMOS CUTIPA
ING MECANICO ELECTRICISTA
CIP 78419

M.Sc. Felipe Condori Chambill.
SUBDIRECTOR DE INVESTIGACIÓN
EPIME

turnitin Página 2 of 142 - Descripción general de integridad Identificador de la entrega trn:oid:::8254:416594655

DEDICATORIA

Dedico este trabajo a mis amigos y amigas, quienes han sido una fuente de apoyo, alegría y fortaleza durante todo este proceso. A mi familia, que siempre ha estado a mi lado, brindándome su amor y confianza, sin los cuales no hubiera llegado hasta aquí.

Aldair Rodrigo Benavente

AGRADECIMIENTOS

A lo largo del camino para la realización de este trabajo, he recibido el apoyo de muchas personas a las cuales me gustaría expresar mi más sincero agradecimiento. A mi familia, por su constante apoyo, cariño y comprensión incondicional durante todo este proceso. A mis amigos y amigas, tanto los que están cerca como aquellos que, aunque estén lejos, siempre han estado presentes de alguna manera para brindarme su ánimo y aliento.

Quiero extender un agradecimiento especial al Ing. John Caballero T., por su guía, paciencia y valiosa colaboración en la culminación de este proyecto. Asimismo, mi gratitud va hacia mi universidad, que me brindó la formación académica necesaria y las herramientas para crecer como profesional.

No puedo dejar de mencionar a los docentes que han marcado mi trayectoria académica, como el Ing. Mateo S. M., el Ing. Walter P. P., y el Ing. Armado Tito C. C., quienes, con su dedicación y enseñanza, han sido una fuente de inspiración y motivación para alcanzar este objetivo. A todos ellos, mi más sincero agradecimiento por haber sido parte fundamental en esta etapa de mi vida.

Aldair Rodrigo Benavente

ÍNDICE GENERAL

Pág.

Área: Control de procesos

Tema: Aprendizaje Automático con Random Forest y Regresión lineal múltiple

Fecha de sustentación: 27 de diciembre del 2024

ÍNDICE DE TABLAS

Pág.

ÍNDICE DE FIGURAS

Pág.

ÍNDICE DE ANEXOS

Pág.

ACRÓNIMOS

IA:	Inteligencia Artificial
RMSE:	Root Mean Square Error
MAE:	Mean Absolute Error
FFT:	Fast Fourier Transform
TAN:	Total Acid Number
TBN:	Total Basic Number
ARIMA:	AutoRegressive Integrated Moving Average
RNN:	Recurrent Neural Network

RESUMEN

Esta tesis aborda la aplicación de algoritmos de Machine Learning, específicamente Random Forest y Regresión lineal Múltiple para el mantenimiento predictivo de bombas de agua multietapa. El objetivo principal es anticipar es estado futuro de las máquinas a partir de datos históricos, optimizando la gestión de fallas y el mantenimiento predictivo particularmente en equipos críticos. La metodología del estudio se fundamenta en un enfoque observacional, descriptivo y explicativo, donde se analizan datos cuantitativos y cualitativos. Primero se recolecto datos en formato CSV, se cargó a la plataforma Google Colab para el respectivo desarrollo del código en Python, preprocesamiento de datos, entrenamiento de algoritmos y análisis de resultados mediante las métricas RMSE y MAE, finalmente se comparó el desempeño para identificar el algoritmo más adecuado. Como resultado, según las métricas con una estimación para los datos normales y con 4 días más de entrenamiento para regresión lineal múltiple el RMSE fue de 0.070 y el MAE de 0.044 para 4 días más de entrenamiento mientras que para Random Forest con 4 días más de entrenamiento las métricas RMSE fue de 0.037 y un MAE de 0.006 mostrando un mejor desempeño para el algoritmo de Random Forest. Cabe mencionar que se usó la validación Walk-Forward para manejo de series temporales con Random Forest. Como conclusión, se logró implementar los algoritmos para el mantenimiento predictivo, destacando a Random Forest como la opción más eficiente. Esta tesis es útil para optimizar la gestión de paradas y el mantenimiento de bombas de agua, aunque identificar modos específicos de falla requerirá estudios adicionales con sensores especializados como acelerómetros.

Palabras clave: Análisis de datos, Machine learning, Mantenimiento predictivo, Python.

ABSTRACT

This thesis addresses the application of Machine Learning algorithms, specifically Random Forest and Multiple Linear Regression, for the predictive maintenance of multistage water pumps. The main objective is to anticipate the future state of the machines based on historical data, optimizing failure management and predictive maintenance, particularly for critical equipment.

The methodology of the study is based on an observational, descriptive, and explanatory approach, where both quantitative and qualitative data are analyzed. First, data was collected in CSV format, uploaded to the Google Colab platform for the respective development of Python code, data preprocessing, algorithm training, and result analysis using the RMSE and MAE metrics. Finally, the performance was compared to identify the most suitable algorithm. As a result, according to the metrics, with an estimate for normal data and an additional 4 days of training for multiple linear regression, the RMSE was 0.070 and the MAE was 0.044 for 4 more days of training, while for Random Forest, with 4 more days of training, the RMSE was 0.037 and the MAE was 0.006, showing better performance for the Random Forest algorithm. It is worth mentioning that Walk-Forward validation was used for handling time series with Random Forest. In conclusion, the algorithms were successfully implemented for predictive maintenance, with Random Forest standing out as the most efficient option. This thesis is useful for optimizing the management of shutdowns and maintenance of water pumps, although identifying specific failure modes will require further studies with specialized sensors such as accelerometers.

Keywords: Data analysis, Machine learning, Predictive maintenance, Python.

CAPÍTULO I

INTRODUCCIÓN

En los últimos años vemos que el uso de las diferentes IA está en aumento, son herramientas muy útiles y potentes encargadas de diferentes tareas, existen IA con aplicaciones en muchas áreas de la sociedad y la industria ya sea para automatizar procesos o hacer el proceso más eficiente, hay un margen dentro de todo esto en la aplicación de tales herramientas ya que la sociedad cree y los medios muestran que algunas IA hacen trabajos de forma más rápida que un humano, más eficiente, es cierto de alguna manera ya que una IA es entrenada para hacer un trabajo dentro de una industria u organización, tales casos según mi punto de vista no son malos ya que un objetivo de las IA es hacer el trabajo humano y por lo tanto tiene que ser algo semejante o incluso mejor.

Cada año esta rama de la ciencia avanza más viendo aplicaciones de estas técnicas no solo a nivel industrial o científico, vemos que se crean IA para mejorar la jugabilidad en videojuegos como el caso de AlphaStar, como DALL-E que es una IA que es capaz de crear arte con solo ingresar una descripción textual de lo que se quiere, entre muchos ejemplos más, el factor importante dentro de todo esto es que es una herramienta que ¨aprende¨ literalmente, hasta el momento no podemos hablar de recrear la creatividad humana.

Con todo esto la aplicación de las IA dentro de la ingeniería es actualmente muy recurrente, como en al análisis de datos, en otros sectores de la ingeniería también se desarrollan herramientas con IA como en este caso.

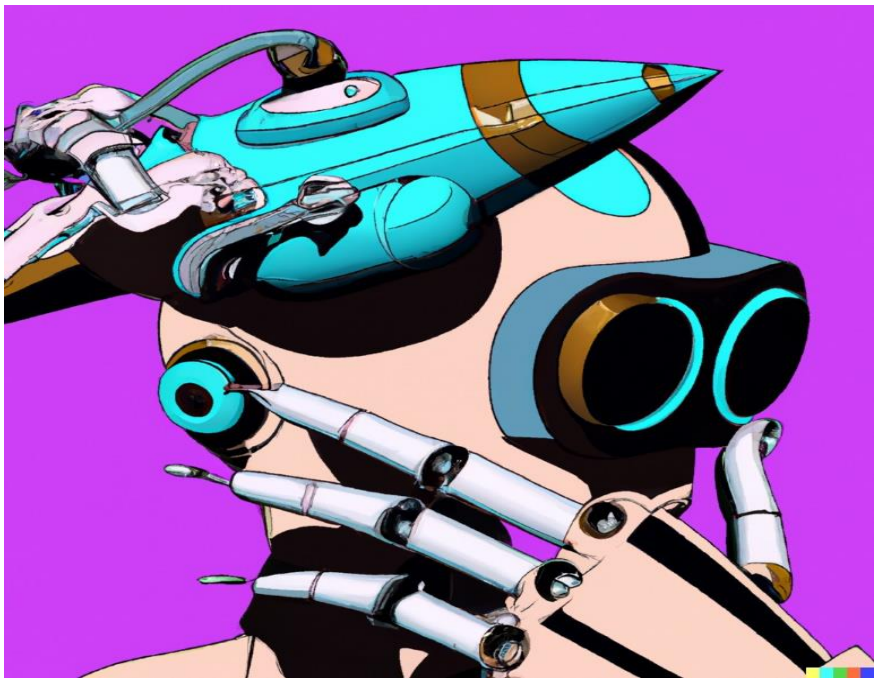
Una parte muy importante dentro de los procesos industriales es garantizar una producción constante, o en tal caso tener interrupciones que no perjudiquen la producción significativamente. Dentro de las áreas de la ingeniería está el mantenimiento, este a su vez tiene ciertas clasificaciones según el momento a ejecutarlo, tenemos el mantenimiento correctivo el cual se aplica ocurrida la falla, el mantenimiento preventivo el cual tomando recomendaciones de los fabricantes se ejecuta, a modo de prevenir una falla sin que esta sea notoria, por último, está el mantenimiento predictivo el cual se entiende como un análisis y uso de técnicas para detectar posibles fallas y defectos iniciales en las maquinas.

Esta clase de mantenimiento es diferente de las demás clases en el sentido que es necesario hacer un análisis de datos de funcionamiento de los equipos, en tal caso muchas veces esta labor la realiza una persona encargada que mediante análisis estadísticos puede predecir una falla en un equipo, esta labor requiere que el equipo necesite una cantidad considerable de instrumentos de medición, sensores que brinden información necesaria para este fin.

La IA tiene protagonismo en esta clase de mantenimiento ya que es posible entrenar una IA para que analice, detecte y haga una predicción en base a los datos, el objetivo general de este trabajo es comprobar que una IA es capaz de hacer las acciones antes mencionadas, los objetivos específicos será aplicar esta herramienta en equipos reales demostrando de esta manera la utilidad de esta misma.

Figura 1

Representación de la inteligencia artificial en un futuro lejano.



Nota. Creado por DALL – E.

1.1. PLANTEAMIENTO DEL PROBLEMA

Las inteligencias artificiales tienen la ventaja de trabajar con bastantes datos y de forma autónoma una vez programadas, pudiendo aprender según el objetivo del programador, en ese sentido, el mantenimiento predictivo necesita de un análisis de datos recolectado por un encargado de mantenimiento, esta labor al ser constante y repetitiva en algunas ocasiones puede ser analizada con una IA. Con la interpretación correcta de los resultados del análisis de Machine Learning a los datos podemos predecir tal cual el mantenimiento predictivo requiere y estudia. Según mi perspectiva el tiempo ganado por el análisis de Machine Learning es útil tanto para el profesional encargado como para la industria que aplica esta técnica de IA, además que Machine Learning es capaz de reforzar las predicciones al paso del tiempo según la precisión de los datos recolectados lo que

conlleva que los errores disminuirán mientras más casos analice y mientras más avance el tiempo.

1.2. FORMULACIÓN DEL PROBLEMA

1.1.1. Problema general

Es necesario hacer un análisis de datos de mantenimiento para obtener resultados para el mantenimiento predictivo, estos resultados serán de utilidad en el periodo que se hagan, además que es necesario un encargado especializado en el área.

1.1.2. Problemas específicos

Machine Learning es capaz de funcionar con varios algoritmos, escoger el más óptimo dependerá de los resultados que se tenga para cada algoritmo, esto se aplica también a un problema real, contrastar resultados en este trabajo como en una aplicación real es muy importante.

1.3. HIPÓTESIS DE LA INVESTIGACIÓN

1.3.1. Hipótesis general

La aplicación de los algoritmos de regresión lineal y random forest al mantenimiento predictivo permitirá una mejora significativa en la precisión y eficiencia en la detección temprana de fallas en equipos industriales en comparación con los métodos tradicionales empleados en la industria. Además, que al usar los dos métodos sabremos cuál de los dos es el más óptimo según la estructura de datos que se tiene.

1.4. JUSTIFICACIÓN DEL ESTUDIO

1.1.3. Justificación práctica

Las empresas están en la libertad de implementar estrategias que sean más rentables, seguros y sofisticados.

1.1.4. Justificación tecnológica

Vemos en estos días que la revolución de las inteligencias artificiales abarca no solo al sector informático, es la aplicación en entornos reales lo interesante de esta tecnología, en tal caso una implementación temprana con un estudio de monitoreo respectivo al o a los algoritmos de Machine Learning usados pondrían en cierta ventaja a la empresa que opta por la implementación de esta colaboración IA y mantenimiento predictivo.

1.1.5. Justificación teórica - científico

Los avances en informática son muy constantes y en el caso de la IA es muy notoria la diferencia que hubo dentro de pocos años, esto es una muestra de que, así como en tan poco tiempo se llegó a avances tan significativos, las aplicaciones industriales de las diversas herramientas de IA que hay y que habrá dentro de unos años serán más que suficientes y necesarias para un óptimo y rentable proceso productivo.

1.5. OBJETIVOS DE LA INVESTIGACIÓN

1.5.1. Objetivo general

Lograr el aprendizaje automático según los datos del equipo para lograr predecir el estado de la maquina en el futuro inmediato.

1.5.2. Objetivos específicos

- Entrenar algoritmos de Machine Learning (regresión lineal y random forest) para las labores de mantenimiento predictivo.
- Analizar los resultados y hacer comparaciones entre algunos algoritmos.
- Revisar cuál de los dos algoritmos tiene mejor desempeño.

CAPÍTULO II

REVISIÓN DE LITERATURA

2.1. ANTECEDENTES DE LA INVESTIGACIÓN

2.1.1. Antecedentes internacionales

Soto (2021) en su trabajo presenta una aplicación de machine Learning, usando una bomba centrifuga simulada, menciona que es una estrategia de utilización de la IA para el uso en mantenimiento predictivo. Usa simulaciones de las cuales extrae datos como el torque, presión y caudal de una bomba centrifuga. Estos datos son tomados con un rango en RPM de 1780 a 6200 para un funcionamiento normal y para la simulación de fallas usa el rango de 6201 a 9500 RPM. Presenta dos modelos uno está basado en la regresión lineal de Bayes y el otro en el modelo kNN como medio para predecir los datos nuevos que brinde el funcionamiento de la bomba. Menciona que, aunque su trabajo sea una simulación presenta una estrategia para el mantenimiento predictivo que puede ser usado en entornos reales.

Vilema (2022) en su trabajo de investigación creo un modelo predictivo de Machine Learning supervisado usando el algoritmo de Random Forest en el software Python. Los datos usados son del repositorio de machine learning de la Universidad de California, Irvine (UCI), uso el 75% del total de datos para entrenamiento y 25% para prueba. De un total de 6 pasos seguidos obtiene el resultado que el modelo aumenta el rendimiento cuando utiliza un conjunto de características seleccionadas e hiperparámetros optimizados, en su conclusión muestra que el modelo funcionó con un buen rendimiento para la detección de

fallas. Como punto final recomienda que una buena preparación de datos es responsable de un buen resultado.

Reveco (2019) en su trabajo recalca que la industria minera está investigando nuevas tecnologías para monitorear a los equipos, con el objetivo de realizar modelos predictivos capaces de predecir el momento de falla de los equipos evitando de esta manera mantenimiento no programado y, por lo tanto, costos excesivos. En ese sentido junto a Anglo American se hizo un modelo para los motores Diesel de la marca Cummins, modelo QSK60 HPI de la flota de camiones Komatsu 930E usando datos de las muestras de análisis de aceites para el entrenamiento de algoritmos de Machine Learning. En la parte final de su trabajo de investigación vemos que los algoritmos de clasificación usados fueron Multiclass Decision Jungle y Forest obteniendo buenos resultados.

Sánchez (2021) usó datos reales proporcionados por la empresa Scania con el objetivo de minimizar una función del coste, en su investigación menciona que los datos que uso contenían bastantes errores de medición, para esto uso técnicas de filtrado e imputación. Escogió random forest como algoritmo de Machine Learning ya que menciona que este algoritmo tiene un rendimiento superior, en la parte final de su trabajo comparo los resultados que obtuvo con los resultados que otros investigadores obtuvieron para el mismo problema.

Bartolomé (2018) en su trabajo de investigación hace una comparación entre el algoritmo de regresión logística y Support vector Machines con el objetivo de saber cuál es el modelo más eficiente, como se mostrara en este trabajo hay varios modelos o algoritmos que podemos usar para la creación de estos modelos, como conclusión de su trabajo menciona que depende de los objetivos un modelo

puede ser o no adecuado, también cabe mencionar que tal como las otras referencia un objetivo del mantenimiento predictivo es reducir costes por mantenimientos repentinos aprovechando las herramientas de la industria 4.0

2.1.2. Antecedentes nacionales

Contreras (2020) menciona el avance tecnológico que el mundo está pasando, Perú no deja de ser ajeno a estos cambios y se están dando de forma lenta pero segura, dentro de estas tecnologías esta la IA, trabajo colaborativo, Big data, internet de las cosas, estas tecnologías digitales son parte fundamental de la industria 4.0. Menciona también la importancia dentro de las industrias del motor de inducción, en tal sentido plantea implementar un método de mantenimiento predictivo basado en IA con el fin de monitorizar el estado del motor y así predecir el momento adecuado de cambio de sus elementos evitando paradas inesperadas de planta, de esta manera su objetivo es hacer que la industria local o la que implemente estos sistemas de gestión de mantenimiento sean más competitivas, Para su trabajo uso el aprendizaje supervisado de Machine Learning para la predicción del estado de los rodamientos de los motores a inducción.

Albornoz (2021) en su tesis muestra la mejora en fiabilidad y eficiencia conseguida cuando se aplica el aprendizaje automático supervisado en el mantenimiento predictivo de motores eléctricos usados en las unidades mineras del Perú, menciona que la fiabilidad y la eficiencia del mantenimiento predictivo es muy importante dentro de las empresas mineras debido al nivel de competitividad en la que están involucradas. También menciona que se prefiere el mantenimiento predictivo ya que esta técnica aprovecha al máximo el tiempo de vida útil de todos los componentes del motor, su trabajo se enfoca en aplicar

los análisis matemáticos relacionados al nivel de aislamiento del bobinado del estator y las vibraciones mecánicas a la que está sometido el motor, finalmente se hace comparaciones entre los dos métodos usados en mantenimiento predictivo, el tradicional el cual consiste en sistemas de control conformados por elementos computarizados y usando aprendizaje automático supervisado que vendría a ser Machine Learning.

Huamán (2021) establece los fundamentos para implementar modelos predictivos basados en inteligencia artificial aplicados al diagnóstico técnico del índice de salud en interruptores de potencia de una empresa de transmisión de energía eléctrica. Su investigación resalta al eficacia y eficiencia de los modelos entrenados con Machine Learning en comparación con métodos tradicionales como la lógica difusa. A diferencia de esta, donde las reglas de parametrización y criterios de inferencia son definidos manualmente el modelo propuesto emplea parámetros técnicos como variables predictoras y desarrolla un modelo matemático propio a partir de análisis computacionales. Los resultados obtenidos muestran un diagnóstico con un 99.27% de efectividad y una reducción del 76.19% en las horas hombre, logrando de esta manera optimizar significativamente el proceso y mejorando la precisión de los diagnósticos.

Fosca (2019) explora las aplicaciones de herramientas de Machine Learning en el campo financiero, centrándose en el pronóstico de acciones, índices bursátiles y commodities. Su trabajo utiliza diversos algoritmos para comparar y analizar los resultados obtenidos estableciendo un marco que sirve como base para investigaciones futuras. El principal objetivo de su tesis fue desarrollar un modelo predictivo basado en Machine Learning para pronosticar el precio del cobre

destacando la importancia de estas herramientas en la mejora de los análisis financieros.

Sanchez y Rivera (2024) describe la aplicación de un modelo de Machine Learning basado en redes neuronales (Autoencoder) para la detección temprana de anomalías en chancadoras primarios, equipos muy importantes en la industria minera. En la tesis se detalla la implementación de técnicas avanzadas de machine learning para mejorar la detección y prevención de fallos, además de su impacto positivo en las operaciones mineras. El modelo que uso fue validado permitiéndole monitorear y planificar las anomalías de manera oportuna reduciendo paradas no planificadas. El modelo fue implementado en una mina de Cobre lo que permitió un monitoreo constante de los chancadores y una planificación proactiva del mantenimiento. Este enfoque demostrado en este trabajo tiene el potencial de ser replicado y adaptado a otras áreas de la industria, mostrando como usar Machine Learning puede mejorar la confiabilidad y optimizar las operaciones mineras

2.2. MARCO TEÓRICO

2.2.1. Inteligencia artificial

La IA tiene varios enfoques para poder ser definida, el concepto general que se maneja es bastante superficial ya que si nos enfocamos en el término “Inteligencia” vemos que tiene una definición no muy clara y es diferente según el punto de vista que se le dé. Ahora en el área de la informática no es aún posible expresar la inteligencia o el conocimiento en una ecuación que sea aplicada a la resolución de un problema o a su propio análisis ya que en el aspecto humano de la inteligencia se ve bastantes expresiones de esta última en las diferentes formas

de creatividad como, por ejemplo, un músico, un pintor, un matemático, un deportista, un ingeniero, etc.

En 2010, Ponce Cruz nos dice que inteligencia artificial se "podría considerar como un dialecto simbólico constituido por cadenas de caracteres que representan el mundo real". Menciona que esto es producto de que la humanidad desde sus inicios represento mediante símbolos el mundo real lo que constituye la base del lenguaje humano.

De forma general según lo que vemos para la informática la inteligencia artificial es un modelo desarrollado capaz de imitar un aspecto de la inteligencia humana.

2.2.2. Machine Learning

El Machine Learning es una rama de la IA enfocada en la creación de aplicaciones que aprenden de los datos y mejoran su precisión con el tiempo sin ser programados para hacerlo (IBM Cluod education, 2020).

En la definición de IA se menciona que la inteligencia humana no puede ser modelada, descrita o representada mediante un código único dentro de una computadora, aun no es posible, lo que sí es posible es lograr representaciones de algunas características de la inteligencia, uno de estos es la creación, aprendizaje, suposición, entre otros. En tal sentido Machine Learning es una aplicación de la IA usando la característica de aprender que tenemos los humanos y los seres vivos en general.

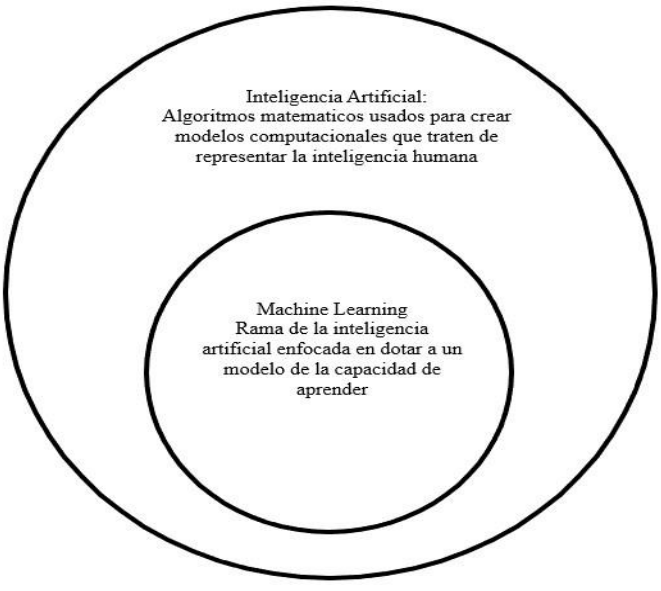
Una aplicación directa de esta característica de la inteligencia humana se describe en Mohri y Talwalkar, (2018). El aprendizaje automático puede definirse

ampliamente como métodos computacionales que utilizan la experiencia para mejorar el rendimiento o hacer predicciones precisas.

En la definición anterior vemos una descripción más enfocada al uso que se le dará al Machine Learning en este trabajo, los métodos computacionales son los algoritmos de Machine Learning los cuales tienen sus diferencias y sus áreas a ser aplicados, la experiencia por lo general son datos recopilados durante bastante tiempo los cuales serán clasificados en datos de entrenamiento y datos a comparar lo que se llama training y testing para demostrar si el modelo usado cumple con lo requerido, y en la parte final menciona la mejora del rendimiento y las predicciones precisas, esto último como resultado del proceso de aprendizaje.

Figura 2

Diferencias entre Machine Learning y la inteligencia artificial.



.

Nota: Orden jerarquico entre Inteligencia Artificial y Machine Learning.

La figura 2, muestra solo una parte de la IA en general, ya que como una categoría más profunda en temas de aprendizaje está el Deep Learning, pero este tema no será tratado en este trabajo.

2.2.3. Aplicaciones de Machine Learning

Para hacer las aplicaciones del uso de Machine Learning hice un resumen de los diferentes enfoques que se le pueda dar, pero lo esencial dentro de la IA y machine Learning es poder contar con datos a analizar, a esta gran cantidad de datos se le conoce como big data. Las aplicaciones más comunes son las siguientes:

- Reconocimiento de imágenes.
- Traducción de idiomas automático.
- Recomendaciones de servicios y productos.
- Autos autónomos.
- Filtrado de spam.
- Detección de estafas.
- Banca.
- Predicción del tráfico vehicular.

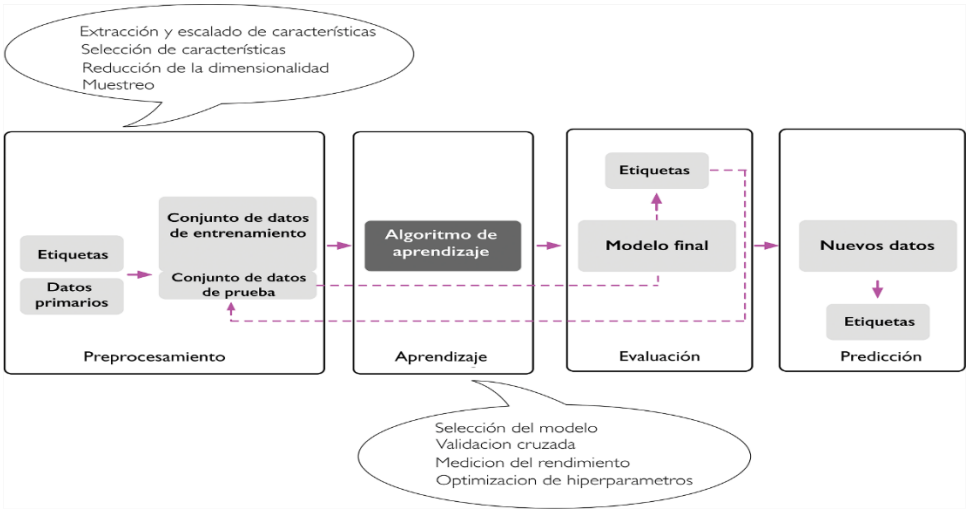
Estos son solo algunas de las aplicaciones que tiene Machine Learning, en (Mohri y Talwalkar, 2018) se menciona aún más aplicaciones.

2.2.4. Pasos a seguir para crear un sistema de aprendizaje automático

Es importante mencionar los pasos que se tiene que seguir para crear un sistema de Machine Learning. Estos pasos siguen un flujo de trabajo típico, aunque los autores asocian los pasos de manera distinta el procedimiento para modelado predictivo es el siguiente:

Figura 3

Hoja de ruta para crear sistemas de aprendizaje automático



Nota: Procedimiento para crear sistemas de machine learning tanto para regresion como para clasificación (Raschka y Mirjalili, 2019)

2.2.5. Tipos de aprendizaje automático

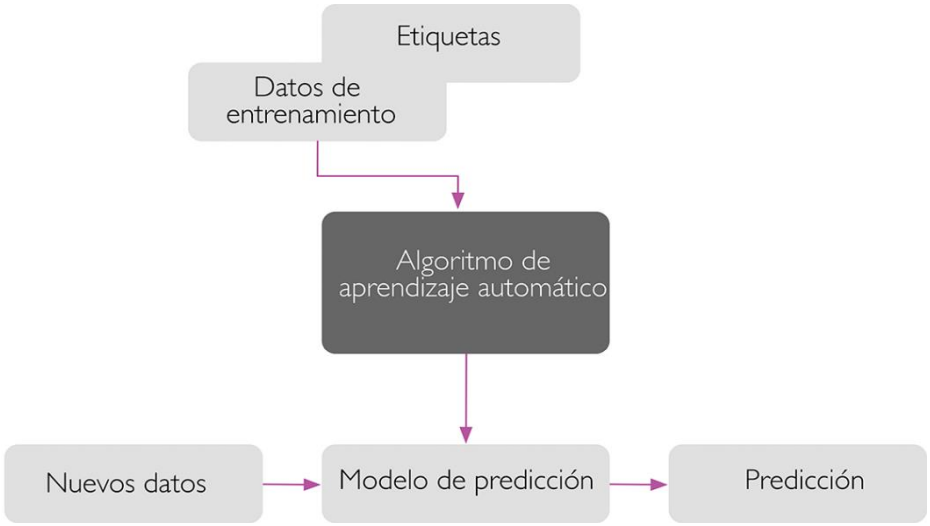
Otro punto a aclarar es que Machine Learning tiene formas de aprender, está el aprendizaje supervisado, el aprendizaje no supervisado y el aprendizaje reforzado.

2.2.6. Aprendizaje supervisado.

El objetivo del aprendizaje supervisado es aprender un modelo, partiendo de datos de entrenamiento etiquetados, que nos permite hacer predicciones sobre datos futuros. Aquí, el termino supervisado se refiere a un conjunto de muestras donde las señales de salida deseadas (etiquetas) ya se conoce (Raschka y Mirjalili, 2019).

Figura 4

Hacer predicciones con el aprendizaje supervisado



Nota: Proceso del aprendizaje automatico supervisado para regresiones (Raschka y Mirjalili, 2019)

De todos estos datos un porcentaje se usará para el entrenamiento del modelo (training) y el porcentaje restante se usará como prueba (testing) para contrastar los datos que el modelo predijo.

El aprendizaje supervisado tiene dos problemas importantes, el primero es el sobreajuste (overfiting) y luego está la correlación (correlation).

Contreras, (2020). El sobreajuste ocurre cuando el modelo usado no generaliza los datos de entrenamiento ya que está muy ajustado a los datos que se le dio. Esto es un problema grave ya que los algoritmos de machine learning buscan patrones entre los datos de entrenamiento con el fin de inferir datos nuevos, como una extrapolación de datos para la predicción. Esto ocurre cuando el modelo de aprendizaje se entrena demasiado o lo hace con datos de entrenamiento anómalos produciendo patrones que no son generales a un funcionamiento normal, también ocurre un sobreajuste cuando no se dispone de la cantidad

suficiente de datos. La correlación se presenta cuando en el modelo a entrenar, se hallan variables relacionados con otras de igual valor, por ejemplo, las variables kilogramos y gramos, entre ellas existe una alta correlación. Se pueden encontrar casos en la cual una variable puede influir casi un 100% en el resultado final. (p. 31).

2.2.7. Aprendizaje no supervisado

Teniendo un conjunto de datos que no están etiquetados, para este caso donde no se sabe la clasificación se busca una relación entre estos, buscando una estructura para encontrar relaciones y poder agruparlos. En 2020, Contreras Alvarez, p. 32, menciona que este tipo de aprendizaje es utilizado en deep learning y en un futuro relegara al machine learning.

El código recibe exclusivamente datos de entrenamiento sin etiquetar y hace predicciones para todos los puntos invisibles. Dado que, en general, no hay ningún ejemplo etiquetado disponible en ese entorno, puede resultar difícil evaluar cuantitativamente el desempeño de un alumno. La agrupación y la reducción de dimensionalidad son ejemplos de problemas de aprendizaje no supervisados (Mohri y Talwalkar, 2018).

No se profundizará en los conceptos del aprendizaje no supervisado ya que para este trabajo se dispone de datos etiquetados.

2.2.8. Aprendizaje reforzado

El aprendizaje reforzado en machine Learning es un enfoque que implica la enseñanza de un modelo a través de la experiencia y la retroalimentación (feedback) en un entorno de prueba y error. En este proceso, la máquina aprende

a tomar decisiones óptimas para lograr un objetivo específico a través de la interacción con su entorno.

El aprendizaje reforzado está basado en que la máquina debe aprender de sus errores y recibir una retroalimentación positiva o negativa, dependiendo de si sus acciones lo acercan o lo alejan del objetivo deseado. Mediante recompensas o castigos, el modelo ajusta su comportamiento y aprende a tomar mejores decisiones.

Este enfoque se ha utilizado en una amplia variedad de aplicaciones de Machine Learning, como en la robótica, los juegos y el control de procesos industriales. A medida que la tecnología avanza, el aprendizaje reforzado se está volviendo cada vez más importante para mejorar la automatización y la eficiencia en diversas industrias.

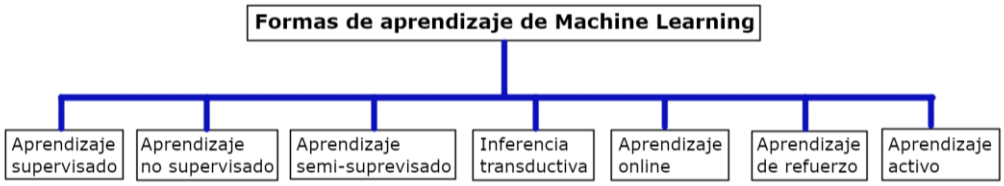
Este tipo de aprendizaje se asemeja al aprendizaje supervisado con la diferencia que la respuesta no es la etiqueta que se espera o que es correcto, generalmente la respuesta es una medida de la forma de reacción del modelo por parte de la función de recompensa. A medida que transcurre las interacciones con su entorno de entrenamiento el modelo maximizara la recompensa en una serie de prueba y error planificada.

2.2.9. Otros tipos de aprendizaje

En Mohri y Talwalkar, (2018), vemos aún más tipos de aprendizaje automático, en total los diferentes tipos de aprendizaje encontrado son los siguientes:

Figura 5

Tipos de aprendizaje en machine learning.



Nota: Otras formas de aprendizaje para Machine Learning.

2.2.10. Algoritmos usados en Machine Learning

2.2.10.1. Regresión lineal

El objetivo de la regresión lineal es modelar la relación entre una o múltiples características y una variable de destino continua (Raschka y Mirjalili, 2019).

Pero la regresión lineal tiene varias subsecciones las cuales veremos a continuación:

2.2.10.2. Regresión lineal simple

Esta clase de regresión es la más común de todas, consta de modelar con una característica sencilla la relación entre una característica de una variable explicativa ubicada en el eje x y la correspondiente respuesta continua o variable de destino situada en el eje y.

$$y = w_0 + w_1 x$$

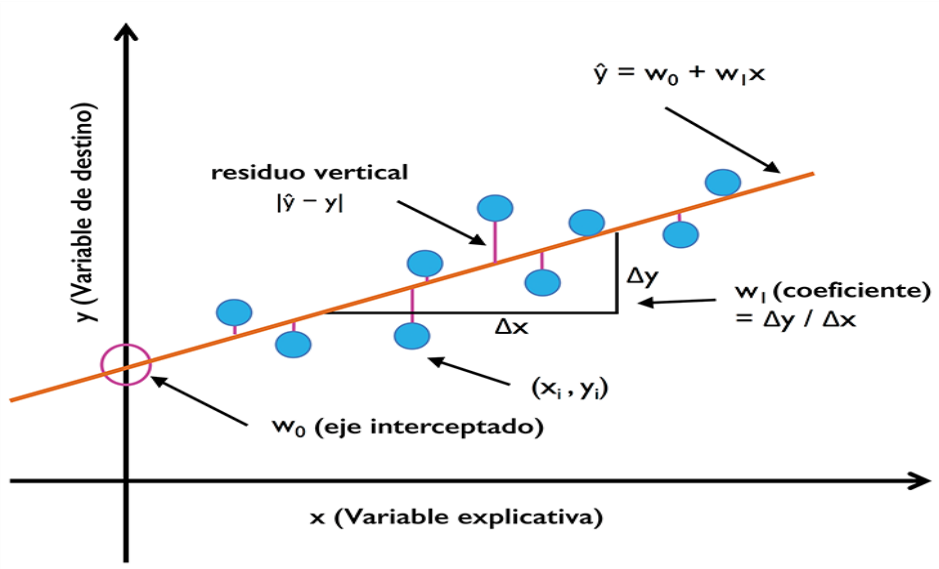
Esta ecuación se representa de varias formas en los diferentes libros citados, pero todas representan la ecuación de una recta solo que en la

teoría de Machine Learning representa la relación de variables, w_0 representa el eje y interceptado, sabemos que un valor positivo de este valor hace que la recta tienda al infinito positivo del eje y, uno negativo lo contrario, en la teoría de Machine Learning este término es llamado bias term o intercept term (Géron, 2019).

El termino w_1 representa la pendiente en una ecuacion de la recta pero en este caso representa el peso de la variable, mayormente llamado weight. El objetivo de este algoritmo es aprender estos pesos para describir la relacion entre la variable explicativa y la variable de destino, estas se usaran para hacer predicciones de las respuestas de variables explicativas nuevas.

Figura 6

Regresión lineal



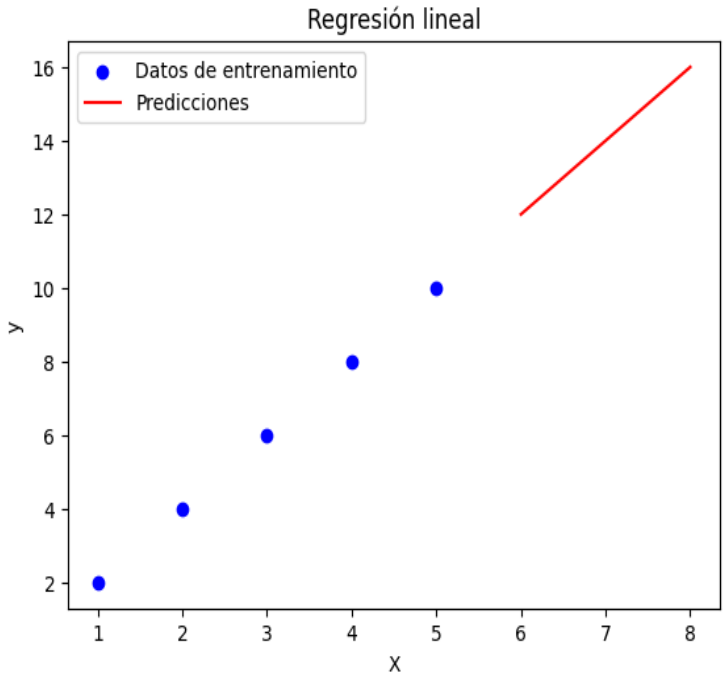
Nota. Ejemplo ilustrativo que incluye variables, ecuación de la recta y otros componentes clave del modelo (Raschka y Mirjalili, 2019)

Este grafico es familiar en estadística, la línea o recta de regresión es la que se ajusta menor a los datos y es la manera en que el algoritmo concluye que los datos tenderán a futuro, la distancia de los puntos a la línea de regresión es llamado offset o residuos y representan los errores de predicción.

La forma en que predice este algoritmo lo veremos a continuación:

Figura 7

Predicciones usando Regresión lineal.



Nota: Ejemplo de prediccion en base a tendencia de los datos con regresion lineal.

Pero esto representa solo una porción del algoritmo complejo utilizado en contextos del mundo real. En la práctica, es poco común encontrar o presenciar una regresión lineal que involucre solamente una variable, como la que acabamos de explorar. Esta es la razón por la cual se recurre a la regresión lineal múltiple.

2.2.10.3. Regresión lineal múltiple

La regresión lineal múltiple constituye una estrategia de análisis estadístico empleada para anticipar el valor de una variable dependiente, considerando diversas variables independientes. A diferencia de la regresión lineal simple, que se limita a una sola variable independiente, la regresión lineal múltiple evalúa múltiples variables independientes que podrían influir en la variable dependiente.

La ecuación que describe a este método es la siguiente:

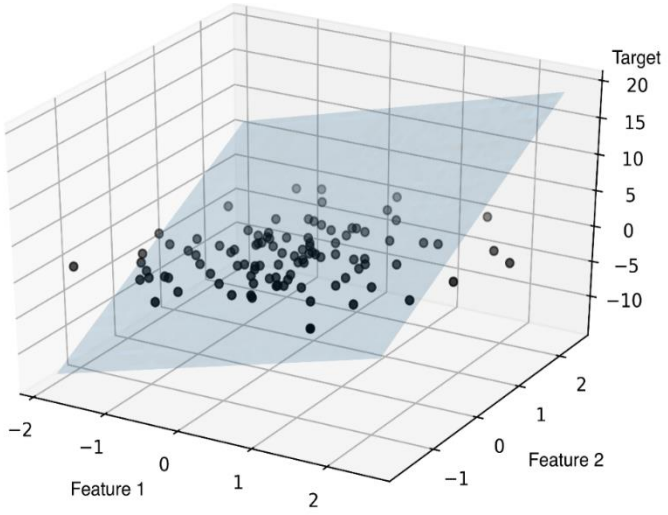
$$y = w_0 x_0 + w_1 x_1 + \cdots + w_m x_m = \sum_{i=0}^{m} w_i x_i = w^T x$$

En este caso, w_0 es el eje y que intercepta con $x_0 = 1$ (Raschka & Mirjalili, Python Machine Learning, 2019, p. 333).

En una regresión lineal simple el grafico representativo es facil de entender, pero, en la regresión lineal multiple al tener mas de dos variables el grafico que representa la relación entre variables puede ser confuso.

Figura 8

Regresión lineal múltiple.

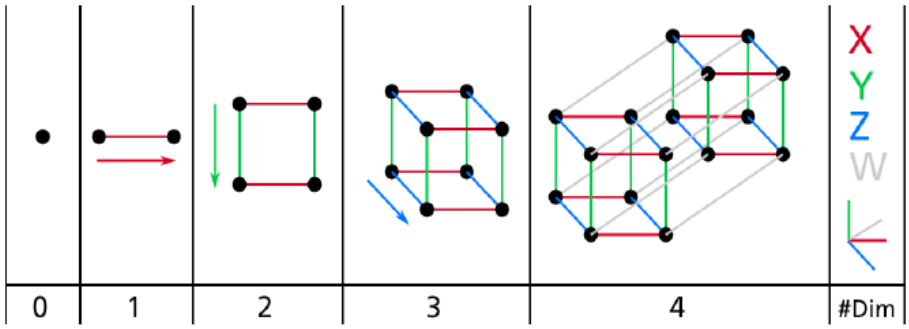


Nota: Representación tridimensional de una regresion lineal múltiple mostrando (Raschka & Mirjalili, 2019)

Para el caso de dimensiones superiores a la tercera es muy difícil entender el grafico, tal como veremos.

Figura 9

Punto, segmento, cuadrado, cubo y teseracto (0D a 4D hipercubos)



Nota: Aumento de dimensionalidad (Géron, 2019)

La manera en que el algoritmo trabaja con los datos es usando matrices, podemos representar la ecuación anterior de la siguiente manera:

$$Y = XW$$

En esta expresión cada letra corresponde a una matriz en específico, estas matrices son:

$$X = \begin{bmatrix} x_{01} & \dots & x_{0m} \\ x_{11} & \dots & x_{1m} \\ x_{21} & \dots & x_{2m} \end{bmatrix}$$

$$Y = \begin{bmatrix} y_0 \\ y_1 \\ y_m \end{bmatrix}$$

$$W = [w_0 \quad \dots \quad w_m]$$

El uso de matrices hace que el entrenamiento del algoritmo sea eficiente.

Al inicio vimos que en la regresión lineal simple se genera una recta o línea que pasa a través de los datos del modelo, en una regresión lineal múltiple al trabajar con dimensiones superiores lo que se genera es un hiperplano entre características, pero, el modelo no puede ser perfecto por lo que la presencia de errores será más que recurrente, la forma más simple de explicarlo es mediante el residuo vertical que se puede ver en la figura 6, aquí vemos la resta entre el valor real de la muestra de datos y el valor que predijo el algoritmo, el error cuadrático medio es una medida de error que será detallada en lo posterior pero en base a esto y con la derivada del error cuadrático medio vectorial es que el algoritmo logra el mejor ajuste a los datos, la expresión para esto es:

$$W = (X^T X)^{-1} X^T Y$$

Que corresponde a la ecuación de coste minimizada, a este método se le llama el método de cuadrados ordinarios, pero existen otras funciones de coste y otros métodos los cuales se aplican según la estructura de los datos, la otra opción más común es el método iterativo del descenso del gradiente, pero para este trabajo solo se usara el método de mínimos cuadrados ordinario.

Como paso preliminar, se realizará un análisis de la relación entre las variables mediante una matriz de correlación. Esta matriz guarda relación con la matriz de covarianza, y en líneas generales, puede considerarse como una versión ajustada de esta última, ya que se encuentra reescalada.

2.2.10.4. Árbol de decisión (Decisión Tree) y Bosque Aleatorio (Random Forest)

El árbol de decisión aplicado a la regresión constituye una metodología dentro del aprendizaje automático que facilita la representación de conexiones intrincadas entre las variables de entrada y la variable objetivo de naturaleza continua. En contraste con su empleo en clasificación, la finalidad del árbol de decisión para regresión es la predicción de valores numéricos en lugar de la asignación de categorías. La estructura de este árbol se fundamenta en la subdivisión recursiva del espacio de características en subespacios más pequeños y homogéneos con respecto a la variable objetivo. Cada nodo del árbol corresponde a una

interrogante o prueba relacionada con una característica del conjunto de datos.

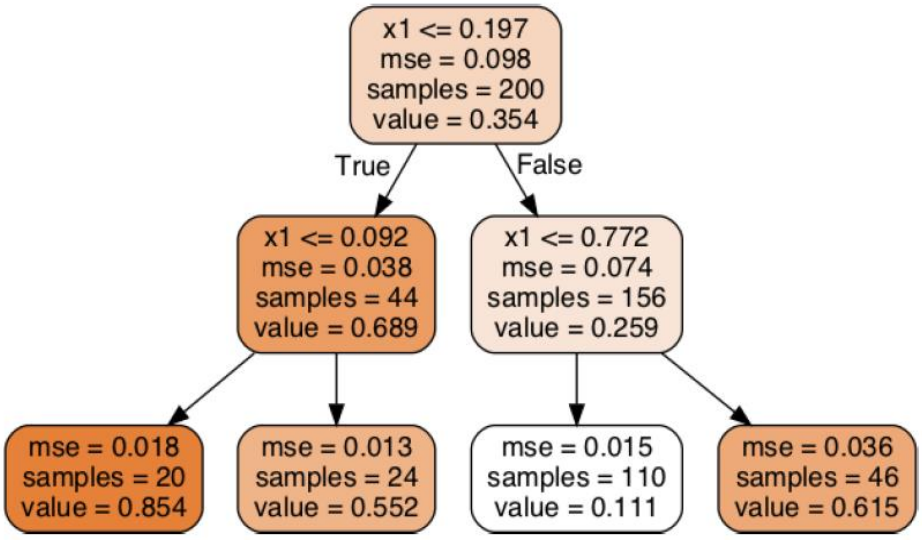
La construcción del árbol se realiza mediante un enfoque de búsqueda recursiva, donde en cada paso se elige la característica y el punto de corte que maximiza la reducción en la varianza o el error en los subconjuntos resultantes. Esta iteración continúa hasta que se satisfaga un criterio de detención, como alcanzar una profundidad máxima o una cantidad mínima de muestras en un nodo. Al realizar una predicción con un árbol de decisión para regresión, los datos de prueba siguen las ramas del árbol de acuerdo con las condiciones establecidas en cada nodo. La estimación final se obtiene calculando la media o la mediana de los valores de la variable objetivo en el nodo hoja correspondiente.

Destacando entre sus atributos, el árbol de decisión para regresión tiene la capacidad de representar relaciones no lineales y de capturar interacciones complejas entre variables. A pesar de ello, se deben tener en cuenta ciertas limitaciones. Por ejemplo, si no se gestiona de manera adecuada, el árbol podría ajustarse demasiado a los datos de entrenamiento, lo que podría afectar negativamente su desempeño con nuevos datos. Para mitigar este problema, se recurre a estrategias como la poda del árbol y la imposición de límites en la profundidad máxima.

Unifiqué ambas definiciones ya que el bosque aleatorio (Random Forest) simplemente implica la aplicación de múltiples árboles de decisión. Cada árbol de decisión posee una estructura de mapa conceptual, como se ilustra en el siguiente ejemplo:

Figura 10

Árbol de decisión para regresión



Nota: Ejemplo de árbol de decisión para regresión, ilustrando sus nodos, ramas u divisiones basadas en los datos (Géron, 2019)

La predicción será el error cuadrático medio (MSE) de todas las instancias, tanto en el uso en clasificación como en regresión el algoritmo de árbol de decisión usa el algoritmo CART, pero el enfoque es distinto, en regresión lo que busca la implementación del algoritmo CART es minimizar el MSE del grupo de entrenamiento (training set), la función de costo CART para regresión es la siguiente:

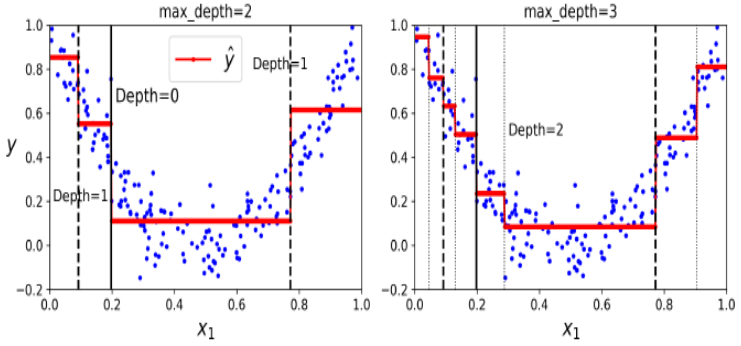
$$j(k, t_k) = \frac{m_{left}}{m} MSE_{left} + \frac{m_{right}}{m} MSE_{right} \quad donde \begin{cases} MSE_{node} = \sum_{i \in node} (\hat{y}_{node} - y^{(i)})^2 \\ \hat{y}_{node} = \frac{1}{m_{node}} \sum_{i \in node} y^{(i)} \end{cases}$$

Lo que hace este algoritmo es dividir el grupo de entrenamiento en dos subgrupos usando una característica k y un umbral t_k buscando los subgrupos mas puros para este fin según la ponderación del tamaño, una vez dividido el primer subgrupo hace lo mismo con los nuevos sub-

subgrupos recursivamente hasta alcanzar la profundidad máxima o hasta que encuentre una división que reduzca la impureza.

Figura 11

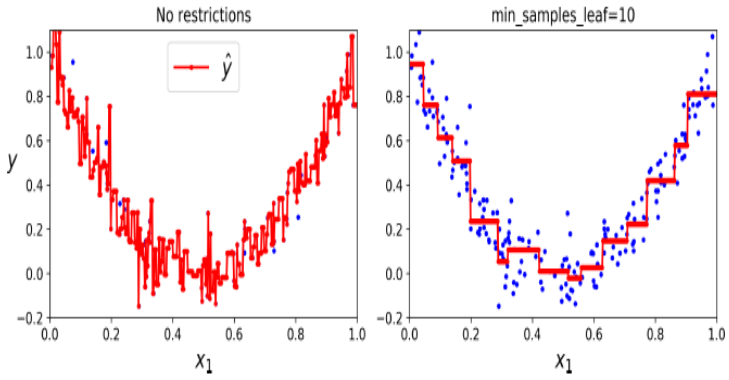
Predicción del árbol de decisión con valor de profundidad 2 y 3.



Nota: Ajuste del algoritmo con niveles de profundidad de 2 y 3 mostrando como el modelo simplifica divisiones al adaptarse a los datos de manera más general (Géron, 2019).

Figura 12

Predicción con valor de profundidad 10 y sin restricciones.



Nota: Ajuste de algoritmo con una profundidad de 10 y sin restricciones, evidenciando un mayor nivel de ajuste a los datos incluyendo tendencia de sobreajuste (Géron, 2019).

Como vemos en los graficos los contornos de decision cambian según la profundidad del modelo, el error dependera del hiperparametro llamado “profundidad”, esta es la desventaja que tiene este algoritmo ya

que como se mencionó anteriormente podemos incurrir en un sobreajuste para el modelo con un valor de profundidad erroneo, ahora se hará una definicion del bosque aleatorio que no es más que la union de varios árboles de decision.

Bosque aleatorio (Random forest)

El Bosque Aleatorio, un algoritmo de aprendizaje automático, se emplea en problemas tanto de clasificación como de regresión. Su fundamento radica en la idea de combinar diversos árboles de decisión para lograr predicciones más precisas y robustas. En líneas generales, el Random Forest construye un conjunto de árboles de decisión independientes y los fusiona para obtener un resultado final. Cada árbol de decisión se entrena con una muestra aleatoria del conjunto de datos de entrenamiento, y luego las predicciones de cada árbol se promedian o ponderan para obtener la predicción final.

Cuando se enfrenta a problemas de regresión, el Random Forest se emplea para anticipar valores numéricos continuos a partir de variables de entrada. Por ejemplo, podría aplicarse en situaciones tales como la estimación de precios de viviendas, considerando características como el tamaño, la ubicación, el número de habitaciones, entre otros.

Una de las ventajas fundamentales del Random Forest en problemas de regresión reside en su habilidad para abordar la alta dimensionalidad y las relaciones no lineales entre las variables. Asimismo, demuestra eficacia en el manejo de valores faltantes y atípicos.

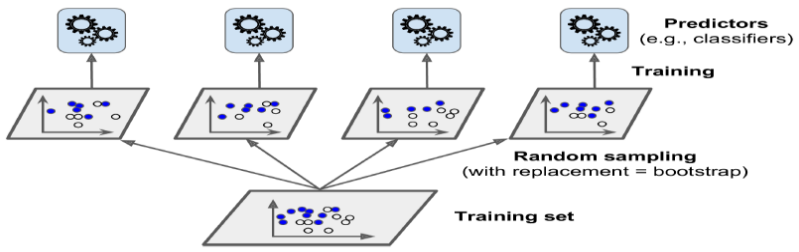
Como hemos comentado, un bosque aleatorio es un conjunto de árboles de decisión, generalmente entrenados mediante el método de bagging (o, a veces, pegado) (Géron, 2019).

La técnica de bagging implica la utilización del mismo algoritmo de entrenamiento para cada predictor. Sin embargo, cada predictor se entrena con diferentes subgrupos aleatorios del conjunto de entrenamiento caracterizándose por el reemplazo del muestreo. A diferencia de esto, el método de pasting no realiza reemplazo durante el muestreo.

Tanto el método bagging como pasting hacen que las instancias de entrenamiento se entrenen muchas veces en varios predictores, solo el método bagging hace que las instancias de entrenamiento se entrenen en un mismo predictor, una vez entrenado los predictores, el conjunto hace predicciones para una nueva instancia agregando las predicciones de todos los predictores, en el caso de la aplicación para clasificacion usa el valor más frecuente y para nuestro caso en regresion se usa el promedio de los valores.

Figura 13

Bagging/Pasting muestreo y entrenamiento del Training set.



Nota: Proceso de muestreo y entrenamiento del conjunto de datos mediante Bagging o Pasting (Géron, 2019)

2.2.11. Preprocesamiento de datos y métricas de evaluación

El preprocesamiento de datos se refiere a un conjunto de técnicas y procedimientos aplicados a los datos antes de su empleo en el entrenamiento de un modelo de aprendizaje automático. El propósito del preprocesamiento es convertir los datos de entrada en un formato más apropiado para el modelo de aprendizaje automático y mejorar su calidad. Este proceso abarca tareas como la depuración de datos, la normalización, la selección de características y la reducción de dimensionalidad.

En cuanto a las métricas de evaluación, estas representan indicadores utilizados para valorar la calidad y precisión de un modelo de regresión. Constituyen un componente esencial del proceso de aprendizaje automático, ya que permiten determinar la utilidad del modelo para abordar el problema en cuestión. Algunas métricas de evaluación comúnmente empleadas en regresión abarcan el error cuadrático medio (MSE), la raíz del error cuadrático medio (RMSE), el error absoluto medio (MAE), el coeficiente de determinación (R^2) y el error porcentual absoluto medio (MAPE).

2.2.12. Limpieza de datos

La depuración de datos se presenta como un procedimiento fundamental en el análisis de datos. Implica la identificación, corrección y eliminación de errores y duplicados en los datos. Este proceso resulta crucial para asegurar la calidad de los datos y evitar decisiones incorrectas basadas en información errónea o incompleta.

El proceso de limpieza de datos implica varias etapas, como la identificación de datos faltantes, eliminación de datos duplicados, corrección de

errores tipográficos, normalización de datos y la detección y eliminación de valores atípicos.

Uno de los primeros pasos en la limpieza es la identificación de datos faltantes. Los datos faltantes pueden ocurrir por diversas razones, como la falta de ingreso de datos por parte del usuario o un error en la adquisición de datos. La forma de manejar los datos faltantes dependerá del conjunto de datos en cuestión y de la cantidad de datos faltantes. A veces es posible reemplazar los datos faltantes por estimaciones razonables basadas en los datos disponibles, mientras que otras veces puede ser necesario eliminar los registros que contienen datos faltantes.

La eliminación de datos duplicados es otra tarea importante en la limpieza de datos. Los datos duplicados pueden ocurrir debido a errores en la entrada de datos o debido a la naturaleza de los datos recopilados. La presencia de datos duplicados tiene el potencial de distorsionar los resultados del análisis, por lo cual resulta fundamental identificar y eliminar los registros duplicados.

La corrección de errores tipográficos también es importante en la limpieza de datos. Los errores tipográficos pueden ocurrir cuando los datos se ingresan manualmente o cuando se copian y pegan datos de una fuente a otra. La corrección de errores tipográficos puede implicar la comparación de los datos con una lista de valores válidos o la utilización de técnicas de limpieza de texto para corregir los errores de ortografía y gramática.

Finalmente, la identificación y eliminación de valores atípicos representan un paso crucial en el proceso de limpieza de datos. Estos valores inusuales, que pueden surgir debido a errores en la introducción de datos o a datos que se sitúan

fuera del rango normal, tienen el potencial de distorsionar los resultados del análisis. La detección de valores atípicos implica identificar aquellos que se ubican fuera de un rango definido o eliminar los valores que están significativamente alejados de la media.

2.2.13. Normalización

En el ámbito del aprendizaje automático, la normalización constituye una etapa esencial de preprocesamiento de datos. Su propósito es transformar las características o variables de entrada a una escala uniforme, con el fin de impedir que algunas de estas características ejerzan un peso desproporcionado en el modelo debido a su magnitud original. Este proceso también contribuye a mejorar la convergencia del modelo durante el entrenamiento y puede ser particularmente relevante al emplear algoritmos de optimización sensibles a la escala de las características, como en el caso de la regresión lineal, KNN, SVM, árbol de decisión y bosque aleatorio, aunque estos últimos admiten trabajar con datos no normalizados.

Existen diversas técnicas de normalización que son ampliamente empleadas en el ámbito del aprendizaje automático, entre ellas, la normalización min-max y la normalización Z-score. La normalización min-max ajusta la escala de los datos a un intervalo específico, generalmente entre 0 y 1. La fórmula es:

$$x_{norm}^{(i)} = \frac{x^{(i)} - x_{min}}{x_{max} - x_{min}}$$

En este caso, $x^{(i)}$ es una muestra concreta, x_{min} es el valor más pequeño en una columna de características, y x_{max} es el valor más grande (Raschka & Mirjalili, Python Machine Learning, 2019, p. 143).

La normalización Z-score también es conocida como estandarización, escala los datos para que tengan una media de 0 y una desviación estándar de 1, la fórmula es la siguiente:

$$x_{std}^{(i)} = \frac{x^{(i)} - \mu_x}{\sigma_x}$$

Aquí, μ_x es la muestra media de una columna de características concreta y σ_x es la correspondiente desviación estándar (Raschka & Mirjalili, Python Machine Learning, 2019, p. 144).

2.2.14. Selección de caracteristicas

Se trata de un procedimiento que conlleva la identificación y elección de un subconjunto de características o variables consideradas relevantes para abordar el problema en cuestión. El objetivo de llevar a cabo la selección de características es reducir la complejidad inherente al modelo y, al mismo tiempo, mejorar su capacidad de generalización.

En general, existen dos tipos de técnicas de selección de características: las basadas en filtros y las basadas en envoltorios.

Las técnicas basadas en filtros evalúan la relevancia de cada característica de manera independiente, empleando medidas estadísticas como la correlación o la prueba t, con el propósito de determinar la importancia de cada característica. Posteriormente, se procede a clasificar las características según su relevancia y se elige un subconjunto de características más destacadas para la construcción del modelo.

Por otro lado, las técnicas basadas en envoltorios evalúan la relevancia de conjuntos de características en conjunción con el modelo de aprendizaje automático. En este enfoque, se generan diversos modelos utilizando diferentes subconjuntos de características, y se mide su rendimiento utilizando criterios tales como la precisión o el error de validación cruzada. El subconjunto de características que demuestra el mejor rendimiento es seleccionado para conformar el modelo final.

Además de las técnicas de selección de características basadas en filtros y envoltorios, también existen enfoques basados en incrustaciones, cuyo objetivo es aprender de manera automática una representación óptima de características mediante la creación de un modelo de aprendizaje automático.

Si observamos que un modelo funciona mucho mejor en un subconjunto de datos de entrenamiento que en un conjunto de datos de prueba, dicha observación es un fuerte indicador de sobreajuste (Raschka & Mirjalili, Python Machine Learning, 2019, p. 145).

Esto se soluciona con la funcion cross validation, pero esto lo veremos en el desarrollo del modelo.

2.2.15. Reducción de dimensionalidad

La reducción de dimensionalidad es un procedimiento que busca disminuir el número de variables o características en un conjunto de datos, procurando retener la mayor cantidad de información relevante posible. El propósito fundamental de la reducción de dimensionalidad es simplificar el modelo, mitigar el ruido o redundancia en los datos, optimizar la eficiencia del proceso de

entrenamiento y evaluación, y fortalecer la capacidad de generalización del modelo a datos no vistos o nuevos.

La reducción de dimensionalidad puede ser útil en muchas aplicaciones de análisis de datos, incluyendo la regresión, la clasificación, el clustering y la visualización de datos. Existen varias técnicas para reducir la dimensionalidad, que incluyen el Análisis de Componentes Principales (PCA), el Análisis de Discriminante Lineal (LDA), la Factorización de Matrices, y el t-SNE.

2.2.16. Outliers y boxplot

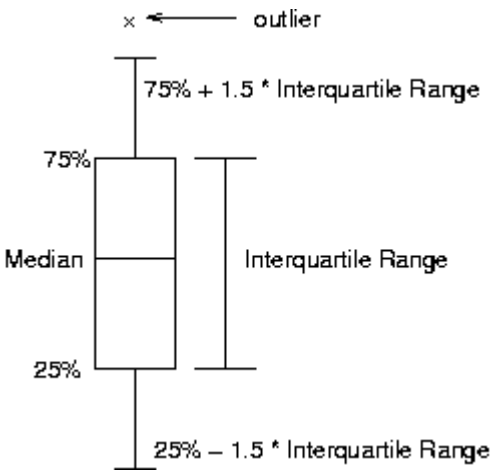
Los outliers o valores atípicos son observaciones que están a una distancia anormal de otros valores de la muestra total o aleatoria del conjunto de datos analizado, para la detección de outliers se tiene técnicas como usar boxplot o scatterplot.

En este caso usamos boxplot para ver la presencia de outliers, aunque esto dependerá de la distribución de los datos, los boxplot se usan para datos con distribuciones normales.

El boxplot representa gráficamente el comportamiento de los datos en el medio y en los extremos de las distribuciones usando para esto la mediana y los cuartiles inferior y superior (25% y 75%), los datos que estén dentro del rango intercuartil serán los datos normales y los que estén lejos de ese rango serán los outliers.

Figura 14

Explicación de Boxplot



Nota: Ejemplo de Boxplot que ilustra la distribución, mediana y posibles valores atípicos en un conjunto de datos (https://math.stackexchange.com/q/2491589).

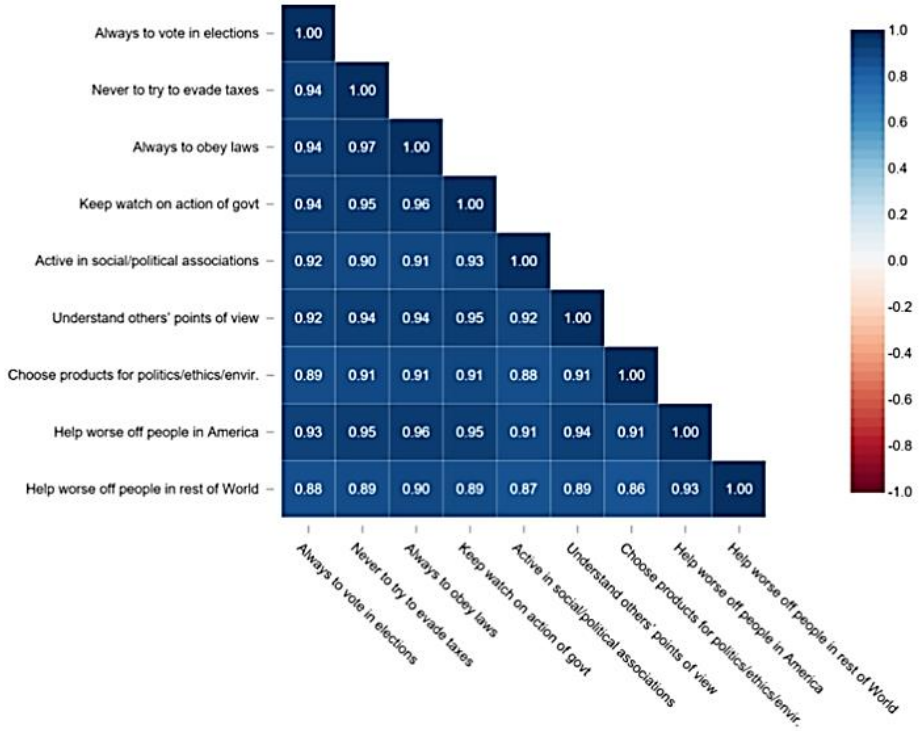
2.2.17. Correlación

La correlación representa una medida de la relación existente entre las variables independientes y la variable dependiente en la base de datos. Para evaluar dicha correlación, se pueden utilizar distintos métodos, como la correlación de Pearson, la correlación de Spearman y la correlación de Tau de Kendall.

Esta correlación se representa generalmente en una matriz, esta matriz es una tabla que representa los coeficientes calculados con los métodos anteriores, es muy usado para resumir los datos, las relaciones entre estos para descartar las variables más significativas de las menos significativas y hacer un análisis de datos más ligero para el procesador usado.

Figura 15

Matriz de correlación



Nota: Matriz de correlacón con mapa de calor, mostrando las relaciones entre variables con valores entre -1 y 1 (https://www.displayr.com/what-is-a-correlation-matrix/).

Generalmente, incluye el coeficiente de correlación de Pearson, que evalúa la dependencia lineal entre las características. Este coeficiente, con un rango de -1 a +1, se calcula mediante la covarianza entre dos características (x, y), donde el numerador es la covarianza y el denominador es el producto de sus desviaciones estándar.

$$r = \frac{\sum_{i=1}^{n}\left[\left(x^{(i)} - \mu_x\right)\left(y^{(i)} - \mu_y\right)\right]}{\sqrt{\sum_{i=1}^{n}\left(x^{(i)} - \mu_x\right)^2}\sqrt{\sum_{i=1}^{n}\left(y^{(i)} - \mu_y\right)^2}} = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$$

En esta ecuación μ representa la media de las muestras de características, σ_{xy} es la covarianza y σ_x, σ_y son las desviaciones estándar.

Se aconseja utilizar esta matriz para evaluar la relevancia de las variables con respecto a la variable objetivo. En caso de que una variable no sea considerada importante, es común eliminar su columna correspondiente, creando así una nueva base de datos con las variables más significativas. Esta decisión suele depender del criterio del modelador del algoritmo.

La correlación empleada en este estudio es la correlación de Spearman, ya que esta medida se utiliza en situaciones donde hay variables numéricas y categóricas. La correlación de Spearman, también conocida como correlación de rangos, explora la relación entre dos variables y actúa como la versión no paramétrica de la correlación de Pearson, sin requerir en este caso una distribución normal de los datos.

La ecuación que representa esta correlación es la siguiente:

$$r_s = 1 - \frac{6 * \sum d_i^2}{n * (n^2 - 1)}$$

Donde d es la diferencia de rangos entre las dos variables y n es el numero de casos. Este coeficiente de correlación varia de -1 a 1 donde:

Para un valor de correlación de 0.0 < 0.1 no se tiene correlación, para 0.1 < 0.3 hay poca correlación, para 0.3 < 0.5 hay correlación media, para 0.5 < 0.7 hay correlación alta y para 0.7 < 1 la correlación es muy alta.

Ahora veremos las métricas de evaluación usadas, para este modelo usaremos solo las siguientes métricas:

2.2.18. Error cuadrático medio (RMSE).

Representa la desviación estándar de los valores de error derivados de las predicciones. El RMSE actúa como indicador del grado de dispersión de los valores residuales, ofreciendo información acerca de la concentración de datos en torno a la línea que mejor se adapta a la tendencia de los datos a lo largo del tiempo. La fórmula de esta métrica de evaluación es la siguiente:

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)}{n}}$$

En esta ecuación $\hat{y}_i$ son los valores que predijo el algoritmo, y_i son los valores observados y n es el número de observaciones.

2.2.19. Error absoluto medio (MAE)

El MAE (Error Absoluto Medio) es una métrica de evaluación para modelos de regresión, la cual calcula el promedio de los errores absolutos entre las predicciones del modelo y los valores reales. El MAE se expresa en las mismas unidades que los datos originales y constituye una medida de la precisión del modelo.

La fórmula del MAE es la siguiente:

$$MAE = \frac{1}{n}\sum_{t=1}^{n} |x_t - \hat{x_t}|$$

Aquí, x_t es un valor real (testing) y $\hat{x_t}$ es un valor predicho.

El MAE es una métrica simple y de fácil interpretación que proporciona una medida directa del error promedio del modelo. Sin embargo, el MAE no tiene

en cuenta la dirección del error (es decir, si el modelo está subestimando o sobreestimando los valores reales), y puede no ser adecuado para conjuntos de datos con valores atípicos o distribuciones asimétricas.

2.2.20. Mantenimiento

El mantenimiento es toda actividad encaminada a conservar las propiedades físicas de una institución o empresa a fin de que esté en condiciones para operar en forma satisfactoria y a un costo razonable (Medrano et al, 2017).

El mantenimiento comprende el conjunto de acciones y procedimientos realizados para preservar un bien o equipo. Su propósito es garantizar el funcionamiento óptimo y extender la vida útil del mismo. En otras palabras, el mantenimiento implica una serie de intervenciones preventivas y correctivas orientadas a asegurar la operación segura y eficiente de máquinas, equipos, instalaciones o infraestructuras.

Existen varios tipos de mantenimiento, entre los más comunes se encuentran:

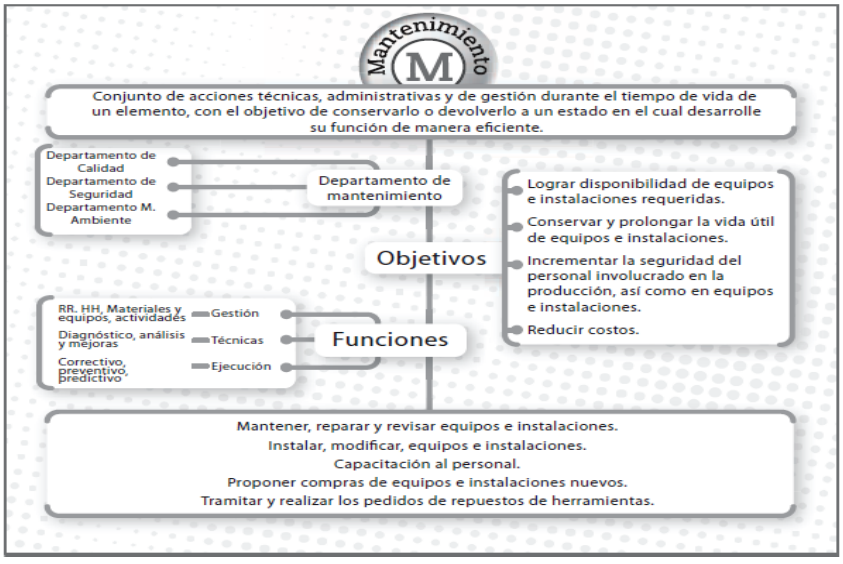
- Mantenimiento preventivo: Realizado de manera periódica y planificada, con el fin de detectar y corregir posibles fallas antes de que se produzcan. El objetivo es evitar las averías y reducir el riesgo de fallos inesperados.
- Mantenimiento correctivo: Corresponde al mantenimiento ejecutado una vez que ha ocurrido una avería o fallo en un equipo, con la finalidad de corregir la falla y restablecer el equipo a su estado original.
- Mantenimiento predictivo: Basado en el monitoreo continuo del estado de los equipos y la identificación de posibles fallos antes de que se produzcan.

Se utilizan técnicas de análisis de datos y de medición para predecir el momento en que se va a producir una falla.

La relevancia del mantenimiento en una industria resulta esencial. La gestión adecuada de equipos y maquinaria garantiza la continuidad fluida del proceso productivo, previene paros no planificados y disminuye el riesgo de accidentes laborales. Asimismo, el mantenimiento apropiado contribuye a extender la vida útil de los equipos, a reducir los costos asociados a reparaciones y reemplazos, y a potenciar tanto la eficiencia energética como la calidad del producto final. En resumen, el mantenimiento representa una inversión que asegura la rentabilidad y competitividad de una industria.

Figura 16

Esquema del concepto del mantenimiento industrial.



Nota: Representación gráfica que organiza los conceptos principales del mantenimiento industrial (Medrano et al, 2017)

Dentro de los tres mantenimientos solo desarrollare el mantenimiento predictivo al ser el objetivo de este trabajo.

2.2.21. Mantenimiento predictivo

El mantenimiento predictivo es una técnica de mantenimiento no muy común ya que requiere de equipo especializado y de personal calificado, sobre todo en análisis de datos, el objetivo del mantenimiento predictivo es predecir un comportamiento en un determinado equipo, estructura o bien material dentro de una empresa, pero por lo general se asigna un plan de mantenimiento predictivo a equipos importantes, indispensables dentro de un proceso productivo ya que, de ser el caso que estos tengan una falla perjudiquen de forma significativa la producción y generen altos costos de reparación.

El mantenimiento predictivo o mantenimiento basado en la condición se apoya en un conjunto de actividades que permiten predecir y prevenir el desarrollo de fallas en equipos e instalaciones. La aplicación de técnicas especializadas ayuda a detectar con anticipación un desperfecto en el equipo, el mal funcionamiento o el cambio de estado de un equipo o maquina durante su operación (Medrano et al, 2017).

Pero el mantenimiento predictivo en si parte de diferentes fuentes de análisis, estos análisis se realizan con equipos especializados dentro o de forma externa en los equipos. Esto forma parte de las desventajas del mantenimiento predictivo, pero, una buena implementación de este tipo de mantenimiento trae ventajas tanto productivas como económicas a largo plazo.

Estos análisis son los siguientes:

2.2.22. Medición y análisis de vibraciones

La evaluación de vibraciones constituye una técnica de mantenimiento predictivo empleada para la detección y diagnóstico de problemas en maquinaria y equipos. Este método se basa en la medición y análisis de las vibraciones que emiten durante su operación normal. Dichas vibraciones pueden señalar la presencia de fallas o inconvenientes que, de no identificarse y corregirse oportunamente, podrían resultar en daños significativos. El análisis de vibraciones se lleva a cabo con el uso de dispositivos especializados, como acelerómetros y analizadores de vibraciones, que miden las vibraciones en diversas frecuencias y direcciones. Los datos obtenidos se recopilan y procesan mediante software dedicado al análisis de vibraciones, permitiendo la detección de patrones y tendencias que sugieren posibles fallos. Este enfoque posibilita la identificación temprana de problemas como desalineación, desequilibrio, holgura mecánica, complicaciones en la lubricación, daño en rodamientos, entre otros. Al detectar estos problemas de manera oportuna, es posible planificar intervenciones de mantenimiento preventivo que eviten fallas catastróficas y extiendan la vida útil de los equipos.

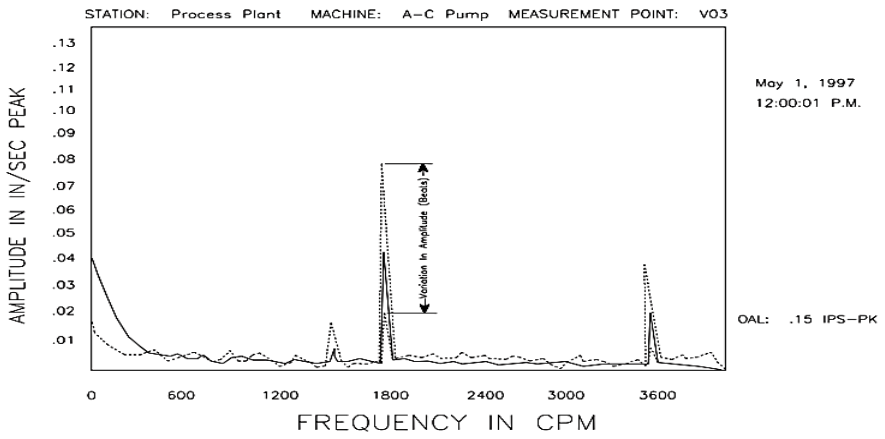
La orientación para la toma de medidas en el análisis abarca las direcciones horizontal, vertical y axial. Estas mediciones se llevan a cabo en los puntos de apoyo, como las chumaceras, y se recomienda realizarlas al menos una vez al mes. Diversas formas de interpretar los resultados incluyen la evaluación de los niveles de vibración total, el análisis de las frecuencias de vibración, la realización de análisis de espectro FFT y la observación de la señal en el dominio del tiempo.

Hay que aclarar que las diferentes formas de analizar los resultados nos dan diferentes resultados según el problema que se está buscando solucionar, en muchos casos cuando se requiera por ejemplo saber si hay un desbalance es necesario usar otras formas de analizar el resultado, será necesario ver la forma de la onda obtenida del análisis y sobre todo saber interpretar los armónicos presentes.

Este método de recolección de datos, enfocado en los niveles de vibración total, destaca por su rapidez al evaluar el estado de equipos rotatorios. Sin embargo, presenta limitaciones, ya que no puede capturar señales de vibración en frecuencias bajas y carece de la capacidad para identificar la fuente específica de vibración excesiva. El patrón típico de los análisis de vibraciones se presenta de la siguiente manera:

Figura 17

La vibración es dinámica y las amplitudes cambian constantemente.



Nota: Gráfica de vibraciones que muestra cómo las amplitudes varían en función de la frecuenca de manera dinámica (Mobley, 2002).

En Mobley, (2002) menciona que los datos de vibración al ser dinámicos necesitan ser promediados para su análisis.

No se ahondará más en este tema ya que es bastante profundo y el objetivo de este trabajo no es la implementación de la mejor técnica de análisis de vibración para bombas de agua.

2.2.23. Ultrasonido

La utilización del ultrasonido como método de mantenimiento predictivo implica la aplicación de ondas sonoras de frecuencia elevada para la detección de problemas en equipos y maquinaria. Este enfoque se fundamenta en la premisa de que las fallas mecánicas en los equipos generan vibraciones y sonidos que producen ondas sonoras de alta frecuencia, las cuales pueden ser identificadas y analizadas.

El equipo utilizado para el análisis de ultrasonido es un detector de ultrasonidos, que mide las ondas sonoras emitidas por los equipos y las convierte en señales audibles o visibles. Estas señales se utilizan para detectar problemas en los equipos, como fugas de aire comprimido, holguras mecánicas, rodamientos defectuosos, entre otros. El ultrasonido es una técnica muy efectiva para la detección temprana de problemas mecánicos, ya que permite detectar fallas que no son visibles a simple vista y que pueden pasar desapercibidas en otros tipos de análisis. Además, esta técnica es muy útil en ambientes ruidosos, donde otros tipos de análisis pueden verse afectados por el ruido de fondo. Una aplicación de este tipo de análisis a este trabajo es su uso para detectar fracturas en la estructura que compone al equipo, se puede hacer un análisis a los soportes de la bomba para ver su rigidez o ver que no tenga fracturas en su interior, además también serviría su uso para ver el desgaste por cavitación en los impulsores de la bomba.

2.2.24. Tribología

El mantenimiento predictivo basado en la tribología implica llevar a cabo mediciones y evaluaciones de las particularidades y propiedades de los materiales, superficies y lubricantes utilizados en los equipos. Esto posibilita la detección de posibles problemas relacionados con fricción, desgaste y lubricación, así como la identificación de las causas subyacentes de estas fallas.

Facilita la toma de decisiones fundamentadas en relación al requerimiento de mantenimiento preventivo, la elección de lubricantes apropiados y la mejora de las condiciones operativas de los equipos. Asimismo, posibilita alargar la duración de servicio de los equipos y disminuir los gastos vinculados con el mantenimiento y la sustitución de los mismos.

2.2.25. Termografía

La termografía es una técnica de mantenimiento predictivo que se puede utilizar para monitorear el estado de la maquinaria, las estructuras y los sistemas de la planta, no solo los equipos eléctricos (Mobley, 2002).

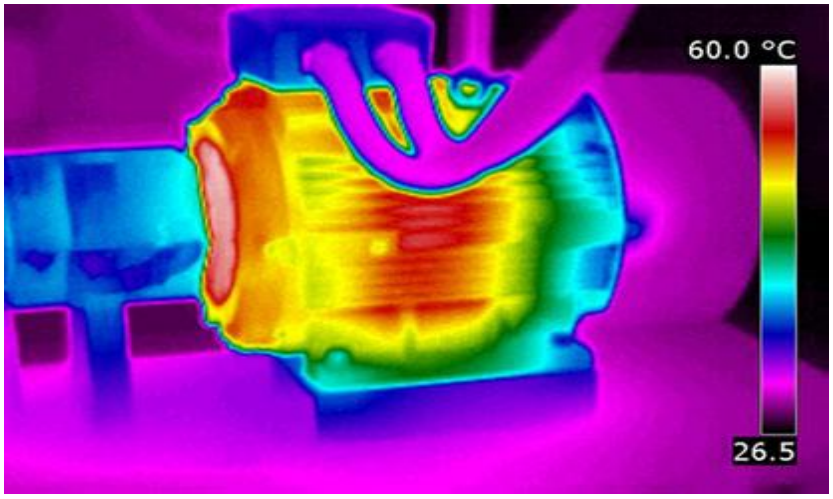
La termografía aplicada al mantenimiento predictivo es una técnica de análisis no destructiva que se utiliza para detectar posibles fallas en equipos y maquinarias a través de la medición de la temperatura de sus componentes.

La termografía se fundamenta en identificar la radiación infrarroja liberada por los objetos a causa de su temperatura. Para aplicar la termografía al mantenimiento predictivo, se emplean cámaras termográficas capaces de registrar la radiación infrarroja emitida por los objetos y transformarla en imágenes

térmicas. Posteriormente, especialistas en termografía analizan estas imágenes para detectar posibles problemas de funcionamiento en los equipos.

Figura 18

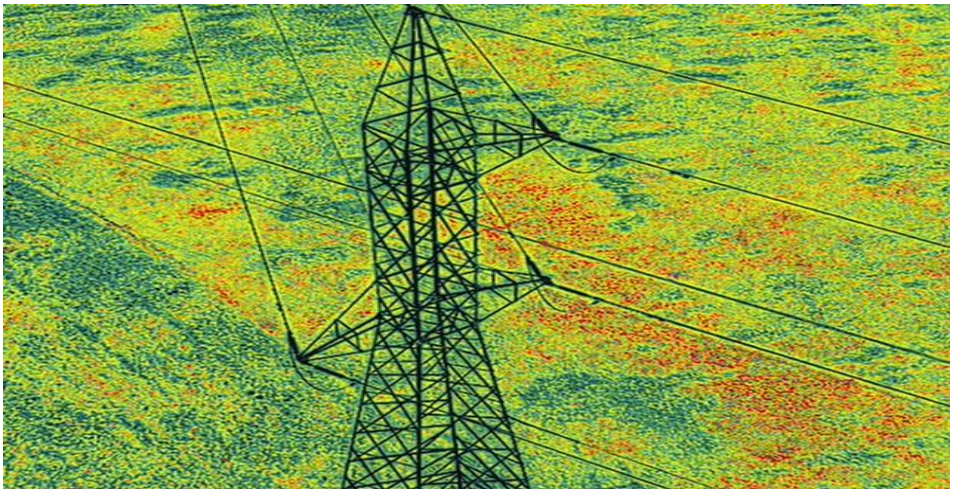
Imagen termográfica de un motor C.A



Nota: Termograma de un motor eléctrico en funcinamiento, destacando las zonas de temperatura para evaluar su estado operativo (https://rulemanesalvear.com.ar/termografia/).

Figura 19

Imagen termográfica de aisladores eléctricos.



Nota: Termograma de una torre de alta tensión (138 Kv), resaltando el estado térmico de los aisladores para detectar posibles fugas de corriente.

La termografía aplicada al mantenimiento predictivo es una herramienta muy útil para detectar fallas incipientes en los equipos, ya que permite detectar cambios en la temperatura de los componentes antes de que se produzcan daños más graves. Además, esta técnica puede ser utilizada en equipos eléctricos, electrónicos y mecánicos, ya que permite detectar posibles problemas de sobrecalentamiento y mal funcionamiento de componentes electrónicos.

La aplicación en nuestro caso va enfocada en el aumento de temperatura en los soportes de la máquina, el criterio lógico en este caso es usar la termografía en los apoyos porque un aumento de temperatura es indicativo de que por ejemplo la lubricación este fallando o se esté desgastando el lubricante generando más fricción entre componentes, también nos puede avisar sobre un aumento de carga no previsto si se enfoca los conductores del motor, entre otras cosas más.

2.2.26. Mantenimiento predictivo en bombas de agua

Como parte final de este resumen es importante saber que técnicas usar para cada tipo de máquina, en muchas ocasiones aplicar todas las técnicas a un equipo no es rentable ni correcto salvo el equipo sea bastante complejo y cuente con varios elementos necesarios donde un análisis de mantenimiento predictivo sea mejor que otro, me refiero a las técnicas que vimos, como ejemplo pongo a la termografía industrial, esta técnica es la más óptima para el caso de tableros eléctricos y en gran medida para equipos eléctricos como motores y transformadores.

En este caso una bomba de agua usada para llevar agua de una parte baja a un alta necesitara en primer lugar al ser una maquina rotativa un análisis vibracional, siendo esta técnica esencial para bombas de agua de gran capacidad,

con el monitoreo continuo de vibraciones generadas por el motor eléctrico de la bomba se puede detectar con anticipación desequilibrios entre ejes, desalineación, holgura en soportes, desgaste o daño en componentes.

Como segundo punto el análisis periódico de aceite lubricante de la bomba puede revelar información sobre el estado de los componentes internos, para esto se deberá realizar pruebas químicas y físicas en el aceite para detectar partículas metálicas, contaminantes y cambios en las propiedades lubricantes TAN y TBN (acidez y alcalinidad), con este análisis podemos detectar generalmente problemas de sellado y desgaste. Como tercer punto la termografía infrarroja es muy importante para identificar puntos calientes, áreas de fricción excesiva o problemas de aislamiento térmico. Estos indicios nos pueden indicar problemas como rodamientos defectuosos o desgastados, fugas de fluidos o mal funcionamiento de los componentes.

El cuarto punto es el monitoreo de la corriente eléctrica que alimenta al motor eléctrico, este monitoreo con el posterior análisis de corriente puede revelar desequilibrios de fase, altas corrientes de arranque, desgaste del aislamiento del motor y problemas de conexión. Como parte final una inspección visual cuidadosa es fundamental en el mantenimiento de bombas de agua de gran capacidad, para esto se puede usar cámaras endoscópicas para examinar internamente componentes de difícil acceso, como los impulsores, sellos y rodamientos ayudando de esta manera a detectar signos de desgaste, corrosión o daños que podrían afectar el rendimiento y la vida útil de la bomba. Estas técnicas deben ser aplicadas regularmente y de forma sistemática, los datos obtenidos deben ser analizados por personal capacitado y especializado y los resultados deben ser usados para la toma de decisiones sobre el mantenimiento preventivo y correctivo

necesario. La implementación de un programa de mantenimiento predictivo (mantenimiento basado en condición) ayuda a maximizar la eficiencia, reducir costos y evitar paradas no planificadas en los equipos (bombas de agua).

2.2.27. Bombas Hidráulicas

Bomba en general es una máquina de fluido, que sirve para comunicar energía al líquido que la atraviesa. Con esta energía puede el líquido remontar el desnivel geodésico existente entre un depósito superior y otro inferior; ser impulsado contra la diferencia de presiones entre la atmosférica y la presión reinante en una caldera, etc. (Mataix, 1975).

Las bombas están basadas en la ecuación de Euler, la altura que transmite el rodete al fluido la describimos con la aplicación de la ecuación de Bernoulli entre la entrada y la salida del rodete:

$$H_{rod} = \frac{p_2 - p_1}{\rho g} + z_2 - z_1 + \frac{c_2^2 - c_1^2}{2g}$$

Esto se aplica a las bombas centrifugas, no se detallará el comportamiento de las bombas de desplazamiento positivo ya que son temas bastante amplios, la aplicación de la bomba usada en este trabajo es para el bombeo de agua potable a reservorios, dentro de toda la clase de bombas como en todos los casos hay una para cada trabajo, pero a grandes rasgos, las ventajas que tienen las bombas centrifugas respecto a las demás son:

- Un acoplamiento más directo con el motor de accionamiento.
- Mayor potencia especifica por unidad de peso lo que conlleva a un menor peso y volumen de la bomba.

- Menor cantidad de partes móviles a excepción del impulsor, esto conlleva a un menor desgaste y una construcción mecánica más simple.
- Se tiene poca o nula cantidad de fuerzas de inercia descompensadas, el rotor está equilibrado estática y dinámicamente (aunque no al 100%).
- Revoluciones especificas más elevadas respecto a los otros tipos de bombas, esto implica la posibilidad de un mayor caudal.
- Flujo continuo del fluido.
- El cierre de la válvula de impulsión no representa un peligro.
- El fluido transportado no se contamina con el lubricante de los cojinetes ya que estos están fuera de la carcasa.

2.2.28. Clasificación de las bombas hidráulicas

Se puede clasificar las bombas hidráulicas en diez categorías distintas, pero las más importantes son las siguientes.

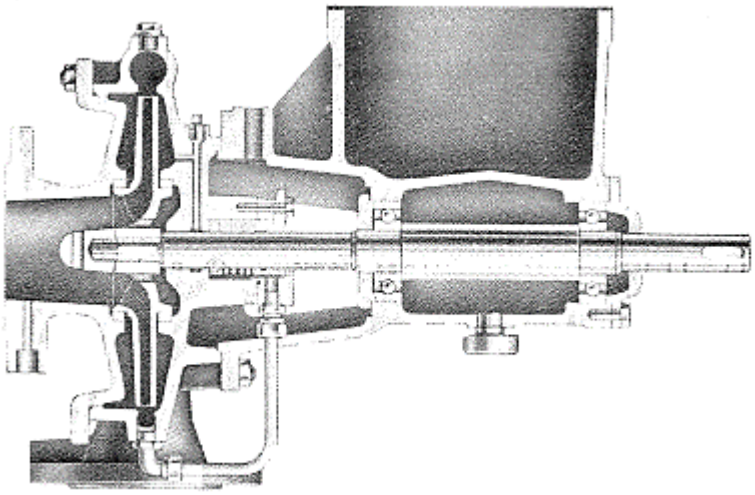
2.2.29. Según la dirección del flujo en el rodete

Se tiene 3 diferentes tipos según esta clasificación:

- Bombas Radiales

Figura 20

Bomba Radial Halberg

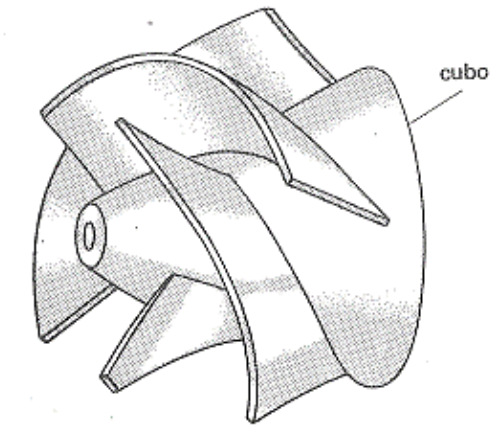


Nota: Corte transversal de una bomba radial, mostrando sus componentes internos como rodamientos, impulsor, etc (Mataix, 1975).

- Bombas Diagonales

Figura 21

Bomba diagonal helicocentrífuga

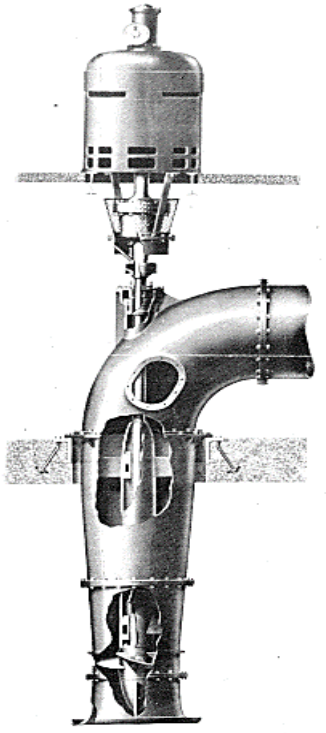


Nota: Vista del impulsor de una bomba diagonal heliocentrífuga, mostrando la forma y estructura del cubo (Mataix, 1975).

- Bombas Axiales

Figura 22

Bomba Axial tipo Kaplan



Nota: Representación de una bomba axial tipo Kaplan con cortes en la tuberia para visualizar algunos componentes internos (Mataix, 1975).

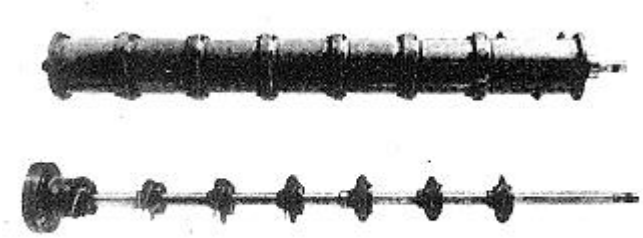
2.2.30. Según la dirección del flujo respecto al eje.

- Bombas simples (De un solo escalonamiento)
- Bombas compuestas (De varios escalonamientos)

En esta clasificación hace referencia a la disposición de los rodetes, ya que pueden estar en serie para más altura y en paralelo para un mayor caudal.

Figura 23

Bomba axial de perforación con siete escalonamientos.



Nota: Imagen de una bomba axial de perforación con siete escalonamientos, destacando la estuctura interna y el recubrimiento de la bomba (Mataix, 1975).

2.2.31. Según el número de flujos

- De simple aspiración o de un flujo
- De doble aspiración o de doble flujo

En esta clasificación vemos las diferentes formas en que una bomba hidráulica aspira el fluido.

2.2.32. Según el tipo de difusor

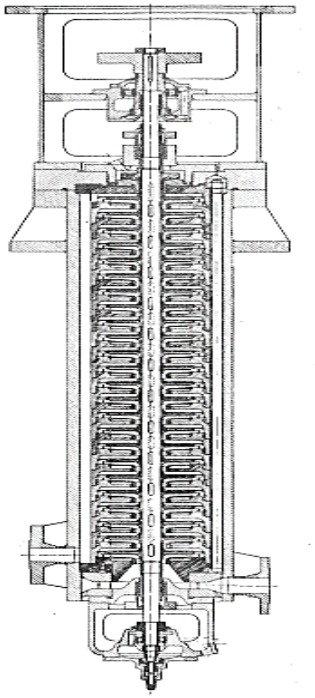
- Bomba hidráulica con corona fija sin alabes y cámara especial.
- Bomba hidráulica con solamente cámara espiral.

2.2.33. Según la posición del eje

- Bomba hidráulica de eje horizontal
- Bomba hidráulica de eje vertical
- Bomba hidráulica de eje inclinado

Figura 24

Bomba vertical de múltiples escalonamientos (sección longitudinal)



Nota: Sección longitudinal de una bomba vertical de múltiples escalonamiento, mostrando su diseño y disposicion interna (Mataix, 1975).

2.2.34. Según la altura o presión que se suministra

Para este tipo de clasificación se tiene bombas de baja presión, media presión y de alta presión.

2.2.35. Según el tipo de accionamiento

Existen bombas accionadas por un motor eléctrico, por motor de gasolina, por un motor Diesel, por turbinas de vapor y por turbinas de gas.

2.2.36. Según el líquido bombeado

Este tipo de bombas es capaz de trabajar con cualquier fluido, puede ser corrosivo, con sólidos en suspensión, agua, aceites, alcoholes, etc.

2.2.37. Según los materiales utilizados en su fabricación

Los aceros más usados son fundición, aceros al Molibdeno, aceros austeníticos, al bronce, etc.

Aunque para casos donde el fluido es abrasivo (cuando tiene solidos como arena) se colocan recubrimientos en el rodete y en la parte interna de la carcasa.

2.2.38. Según el fin a que se destinan

Los usos para los que estará destinado la turbobomba como bomba anti-incendio, riego, suministro de agua, para alimentación de calderas, agrícola, drenaje, marinas, químicas, para los diferentes fluidos usados en la industria, etc.

2.2.39. Fallas más comunes en bombas centrifugas

Las averías que se presentan con mayor frecuencia en las bombas centrífugas son las siguientes

Las bombas centrifugas pueden sufrir obstrucciones en la succión debido a la presencia de sedimentos, residuos o materiales extraños en el agua. Estas obstrucciones pueden afectar el flujo y causar disminución en el rendimiento de la bomba.

El impulsor es una parte vital de la bomba centrifuga y está expuesto al desgaste debido al flujo constante de agua. Con el tiempo, el impulsor puede desgastarse lo que resulta en una disminución del caudal y la presión estregada por la bomba.

Los sellos son componentes cruciales que evitan fugas de agua en las bombas centrifugas. Si los sellos están desgastados o dañados pueden producirse

fugas que comprometan el rendimiento de la bomba y puedan afectar la calidad del agua.

El motor eléctrico que impulsa la bomba puede experimentar fallas como sobrecalentamiento, problemas de conexión eléctrica o desgaste de los devanados del motor. Estas fallas pueden llevar a una disminución de la potencia de la bomba o incluso a un fallo completo del sistema.

La existencia de aire en el sistema de bombeo podría de bombeo puede afectar negativamente el rendimiento de la bomba centrifuga. El aire puede provocar cavitación, lo que reduce la eficiencia de la bomba y puede dañar los componentes internos.

Una alineación incorrecta entre el motor y la bomba puede genera vibraciones excesivas, desgaste prematuro de los rodamientos y una disminución en la vida útil de la bomba.

La bomba de agua usada en este trabajo corresponde a una bomba centrifuga vertical de múltiples etapas, estas bombas son comunes en aplicaciones de bombeo de agua potable ya que son eficientes y confiables, usadas para manejar grandes caudales de agua.

Figura 25

Bombas centrifugas verticales de múltiples escalonamientos.

Nota: Fotografía de bombas centrífugas verticales de mpultiples escalonamientos (https://www.sintechpumps.com/bombas/que-son-las-bombas-de-turbina-vertical-y-como-funcionan/?lang=es).

CAPÍTULO III

MATERIALES Y MÉTODOS

3.1. IMPORTACIÓN DE LIBRERIA

Para llevar a cabo el análisis predictivo del mantenimiento de las bombas de agua, se hizo uso extensivo de diversas librerías de Python que ofrecen una amplia gama de herramientas para la manipulación de datos, el preprocesamiento, la construcción de modelos predictivos y la evaluación de los resultados. A continuación, se detallan las principales librerías utilizadas:

- Numpy: Esta librería fue empleada para realizar operaciones matemáticas de alto rendimiento sobre grandes conjuntos de datos numéricos. A través de sus arreglos multidimensionales (arrays) se facilitó la manipulación eficiente de los datos, permitiendo la creación de funciones vectorizadas y la ejecución de cálculos matemáticos avanzados necesarios para los algoritmos de machine learning. En este caso, se utilizó principalmente para manejar los datos en la ventana deslizante, facilitando el análisis temporal de los datos en el contexto de series temporales y machine learning.
- Pandas: Se utilizó ampliamente para la manipulación de estructuras de datos tabulares a través de DataFrames y Series. Esta librería permitió la carga y limpieza de los datos en formato CSV, así como su transformación en estructuras adecuadas para su análisis. Con pandas se gestionaron operaciones como la imputación de valores faltantes, la transformación de variables categóricas a numéricas y la creación de subconjuntos de datos. También permitió realizar

análisis estadísticos descriptivos como la media, mediana, y desviación estándar, proporcionando una comprensión inicial de la distribución de los datos.

- Matplotlib y Seaborn: Ambas librerías fueron fundamentales para la visualización de datos. Matplotlib permitió la creación de gráficos detallados como líneas de tendencia, gráficos de dispersión y gráficos de barras, facilitando la interpretación visual de los datos. Por otro lado, Seaborn se utilizó para generar visualizaciones más avanzadas y estilizadas, como los mapas de calor (heatmaps), los cuales fueron esenciales para visualizar la matriz de correlación de las variables y entender las interacciones entre ellas. Estas herramientas gráficas ayudaron a identificar patrones importantes, tendencias y posibles valores atípicos antes de aplicar los modelos predictivos.
- Statsmodels: Se utilizó el módulo Holt-Winters SimpleExpSmoothing para llevar a cabo análisis de series temporales, específicamente para aplicar suavizamiento exponencial simple, una técnica que ayudó a modelar patrones estacionales y de tendencia en los datos de funcionamiento de las bombas. Asimismo, se implementó la prueba de Dickey-Fuller Aumentada (ADF), que permitió evaluar la estacionariedad de las series temporales. La estacionariedad es un requisito clave en muchos modelos predictivos basados en series temporales ya que asegura que las propiedades estadísticas del proceso generador de datos no cambian con el tiempo.
- Scikit-learn: Esta librería fue una de las más relevantes en el desarrollo del modelo de machine learning. Se utilizaron varios módulos de scikit-learn para distintas etapas del proceso:

- Preprocesamiento: El preprocesamiento de los datos incluyó la codificación de variables categóricas utilizando LabelEncoder, que transformó las etiquetas no numéricas en valores enteros.
- Modelado: Se entrenaron modelos de regresión utilizando LinearRegression, una técnica de regresión lineal simple que permitió generar modelos de predicción para estimar valores numéricos continuos. Además, se empleó RandomForestRegressor, algoritmo basado en bosques aleatorios que son altamente efectivos para tareas de regresión. Estos modelos se usaron para realizar predicciones sobre los datos de mantenimiento de las bombas y evaluar su capacidad para predecir fallas o anomalías.
- Evaluación: Para medir el rendimiento de los modelos, se utilizó el cálculo del mean_squared_error (RMSE) y del mean_absolute_error (MAE), el cual proporcionó una métrica clave para evaluar la precisión de los modelos en términos de la diferencia entre los valores predichos y los valores reales. Esta métrica fue fundamental para seleccionar el modelo con mejor desempeño en las predicciones.

3.2. ESTADISTICA DESCRIPTIVA Y ANALISIS EXPLORATORIO DE DATOS

En el presente trabajo, la base de datos utilizada fue cargada y manipulada en forma de DataFrame mediante la librería Pandas. A lo largo del análisis, se asignó la variable df para referirse a esta base de datos. El DataFrame contiene un total de 220,320 filas y 55 columnas, lo que representa una gran cantidad de datos que necesitó ser procesada de manera eficiente. Para realizar una inspección inicial, se visualizan las

primeras cinco filas del DataFrame utilizando la función df.head(), lo cual proporciona una vista preliminar de su estructura y contenido. Sin embargo, debido a la extensión y complejidad de los datos, la visualización completa de las filas y columnas no es factible directamente.

Tabla 1

Estructura de los datos

N°	Marca de tiempo	Sensor 00	Sensor 01	Sensor 02	sensor 03		Sensor 48	Sensor 49	Sensor 50	Sensor 51	Condición
0	1/04/2018 00:00	2.46539	47.092	53.212	46.31076	…	157.986	67.7083	243.056	201.389	NORMAL
1	1/04/2018 00:01	2.46539	47.092	53.212	46.31076	…	157.986	67.7083	243.056	201.389	NORMAL
2	1/04/2018 00:02	2.44473	47.3524	53.212	46.39757	…	155.961	67.1296	241.319	203.704	NORMAL
3	1/04/2018 00:03	2.46047	47.092	53.168	46.39757	…	155.961	66.8403	240.451	203.125	NORMAL
4	1/04/2018 00:04	2.44572	47.1354	53.212	46.39757	…	158.276	66.5509	242.188	201.389	NORMAL

Nota: Vista de una parte de los datos, ya que por sus dimensiones no es posible ver todos los datos.

La estructura de los datos incluye columnas con tipos de datos variados. Los tipos de datos principales en este DataFrame son:

- int64: Representa valores numéricos enteros, tanto positivos como negativos.
- float64: Se utiliza para números reales con decimales, lo que incluye tanto valores positivos como negativos.
- object: Este tipo se utiliza para representar datos textuales o combinaciones de texto y números.

A continuación, se muestra la distribución de tipos de datos en el DataFrame:

- 52 columnas contienen datos de tipo float64, lo que representa la mayoría de los valores numéricos con decimales asociados a las lecturas de sensores y mediciones continuas.
- 1 columna contiene datos de tipo int64, correspondiente a valores enteros, que en este caso se refiere al identificador único o la numeración secuencial de los registros.
- 2 columnas contienen datos de tipo object. La primera columna almacena las fechas de los eventos, registradas como combinaciones de números y caracteres. La última columna almacena el estado de la máquina, que puede representar distintos estados categóricos, como "operativo", "en mantenimiento", o "fallo", permitiendo el análisis cualitativo del rendimiento del equipo.

Este análisis preliminar es crucial para entender la estructura de los datos y para identificar posibles problemas, como valores ausentes o anomalías. Por ejemplo, se observó que el sensor 15 no registró ningún valor durante el periodo de recolección de datos, lo que requiere un tratamiento posterior, ya sea mediante imputación o exclusión de la columna en análisis posteriores.

Tabla 2

Tipos de datos presentes en la base de datos

#	Columna	No nulo	Cuenta	Tipo de dato
1	Marca de tiempo	220320	non-null	object
2	sensor_00	210112	non-null	float64
3	sensor_01	219951	non-null	float64
4	sensor_02	220301	non-null	float64
5	sensor_03	220301	non-null	float64
6	sensor_04	220301	non-null	float64
7	sensor_05	220301	non-null	float64

#	Columna	No nulo	Cuenta	Tipo de dato
8	sensor_06	215522	non-null	float64
9	sensor_07	214869	non-null	float64
10	sensor_08	215213	non-null	float64
11	sensor_09	215725	non-null	float64
12	sensor_10	220301	non-null	float64
13	sensor_11	220301	non-null	float64
14	sensor_12	220301	non-null	float64
15	sensor_13	220301	non-null	float64
16	sensor_14	220299	non-null	float64
17	sensor_15	0	non-null	float64
18	sensor_16	220289	non-null	float64
19	sensor_17	220274	non-null	float64
20	sensor_18	220274	non-null	float64
21	sensor_19	220304	non-null	float64
22	sensor_20	220304	non-null	float64
23	sensor_21	220304	non-null	float64
24	sensor_22	220279	non-null	float64
25	sensor_23	220304	non-null	float64
26	sensor_24	220304	non-null	float64
27	sensor_25	220284	non-null	float64
28	sensor_26	220300	non-null	float64
29	sensor_27	220304	non-null	float64
30	sensor_28	220304	non-null	float64
31	sensor_29	220248	non-null	float64
32	sensor_30	220059	non-null	float64
33	sensor_31	220304	non-null	float64
34	sensor_32	220252	non-null	float64
35	sensor_33	220304	non-null	float64
36	sensor_34	220304	non-null	float64
37	sensor_35	220304	non-null	float64
38	sensor_36	220304	non-null	float64
39	sensor_37	220304	non-null	float64
40	sensor_38	220293	non-null	float64
41	sensor_39	220293	non-null	float64
42	sensor_40	220293	non-null	float64
43	sensor_41	220293	non-null	float64
44	sensor_42	220293	non-null	float64
45	sensor_43	220293	non-null	float64
46	sensor_44	220293	non-null	float64
#	**Columna**	**No nulo**	**Cuenta**	**Tipo de dato**

47	sensor_45	220293	non-null	float64
48	sensor_46	220293	non-null	float64
49	sensor_47	220293	non-null	float64
50	sensor_48	220293	non-null	float64
51	sensor_49	220293	non-null	float64
52	sensor_50	143303	non-null	float64
53	sensor_51	204937	non-null	float64
54	Condición	220320	non-null	object
Tipo de datos: float64(52), int64(1), object(2)				

Nota: Cantidad y tipos de datos presentes en la base de datos.

Este preanálisis de los datos proporciona una base sólida para el análisis estadístico ya que permite conocer con precisión las características de las variables involucradas, esto facilitará la correcta aplicación de los modelos de machine learning más adelante.

3.2.1. Estadística aplicada a los datos

El uso de la estadística en el contexto de esta investigación fue fundamental para entender el comportamiento histórico de los datos operacionales de las bombas de agua. La aplicación de técnicas estadísticas descriptivas permitió realizar una evaluación detallada de los datos, con el fin de identificar patrones, tendencias y posibles anomalías antes de aplicar los algoritmos de Machine Learning. Este análisis preliminar es clave para asegurar que los modelos predictivos reciban datos coherentes y útiles, lo que mejora la precisión de las predicciones sobre el estado de los equipos y su mantenimiento.

Figura 26

Estadística aplicada a los datos

	count	mean	std	min	25%	50%	75%	max
Unnamed: 0	220320.0	110159.500000	63601.049991	0.000000	55079.750000	110159.500000	165239.250000	220319.000000
sensor_00	210112.0	2.372221	0.412227	0.000000	2.438831	2.456539	2.499826	2.549016
sensor_01	219951.0	47.591611	3.296666	0.000000	46.310760	48.133678	49.479160	56.727430
sensor_02	220301.0	50.867392	3.666820	33.159720	50.390620	51.649300	52.777770	56.032990
sensor_03	220301.0	43.752481	2.418887	31.640620	42.838539	44.227428	45.312500	48.220490
sensor_04	220301.0	590.673936	144.023912	2.798032	626.620400	632.638916	637.615723	800.000000
sensor_05	220301.0	73.396414	17.298247	0.000000	69.976260	75.576790	80.912150	99.999880
sensor_06	215522.0	13.501537	2.163736	0.014468	13.346350	13.642940	14.539930	22.251160
sensor_07	214869.0	15.843152	2.201155	0.000000	15.907120	16.167530	16.427950	23.596640
sensor_08	215213.0	15.200721	2.037390	0.028935	15.183740	15.494790	15.697340	24.348960
sensor_09	215725.0	14.799210	2.091963	0.000000	15.053530	15.082470	15.118630	25.000000
sensor_10	220301.0	41.470339	12.093519	0.000000	40.705260	44.291340	47.463760	76.106860
sensor_11	220301.0	41.918319	13.056425	0.000000	38.856420	45.363140	49.656540	60.000000
sensor_12	220301.0	29.136975	10.113935	0.000000	28.686810	32.515830	34.939730	45.000000

Nota: Estadística de los 12 primeros sensores de la máquina.

En esta investigación se calcularon los siguientes estadísticos para los datos recolectados:

- Cuenta: Se utilizó para determinar el número total de mediciones disponibles en el conjunto de datos, lo cual es esencial para verificar la cantidad de datos con los que se cuenta para el entrenamiento de los modelos de Machine Learning. Un número adecuado de datos es crucial para la estabilidad y precisión de los algoritmos predictivos.
- Media: La media o promedio de las variables recolectadas sirvió como referencia central para identificar el comportamiento típico de las bombas. Este valor es útil en los procesos de mantenimiento predictivo ya que permite establecer un umbral base sobre el cual se pueden identificar desviaciones importantes que puedan indicar un fallo inminente.
- Desviación estándar: Este parámetro se calculó para cada variable con el fin de medir la dispersión de los datos alrededor de la media. Una alta desviación estándar podría indicar una variabilidad considerable en las

mediciones, lo que puede ser indicativo de inestabilidad en el funcionamiento del equipo, mientras que una baja desviación señala un comportamiento más estable. En el contexto de mantenimiento predictivo una desviación alta puede ser una señal de alerta.

- Mínimo y máximo: Los valores extremos del conjunto de datos son útiles para definir los límites dentro de los cuales han operado las bombas. Estos valores ayudan a detectar comportamientos atípicos o fuera de los rangos habituales que podrían ser indicativos de condiciones anómalas o potenciales fallos. Este análisis es esencial para los algoritmos de mantenimiento predictivo, ya que los eventos fuera de los valores normales deben ser considerados en el modelado.
- Percentiles 25%, 50% y 75%: El cálculo de los percentiles permitió una evaluación más detallada de la distribución de los datos. El percentil 50% o mediana, por ejemplo, proporcionó una medida central alternativa menos afectada por valores extremos o atípicos, mientras que los percentiles 25% y 75% ayudaron a definir los cuartiles y estudiar cómo se distribuyen los datos. Esto fue útil para detectar si una proporción significativa de los datos se encuentra por encima o por debajo de ciertos umbrales, lo que facilita la identificación de patrones de desgaste o fallos en las bombas.

Estos análisis estadísticos fueron aplicados a los datos recolectados para realizar una descripción precisa y completa de las características del sistema. Esta comprensión inicial de los datos permitió optimizar la preparación de los mismos para la fase siguiente en la cual se entrenaron algoritmos de regresión lineal y random forest para predecir la necesidad de mantenimiento. El uso de la estadística descriptiva también permitió una limpieza y tratamiento de los datos

eliminando valores atípicos. De esta forma, se garantizó que los algoritmos de Machine Learning recibieran datos de alta calidad, lo cual es fundamental para mejorar la precisión de los pronósticos.

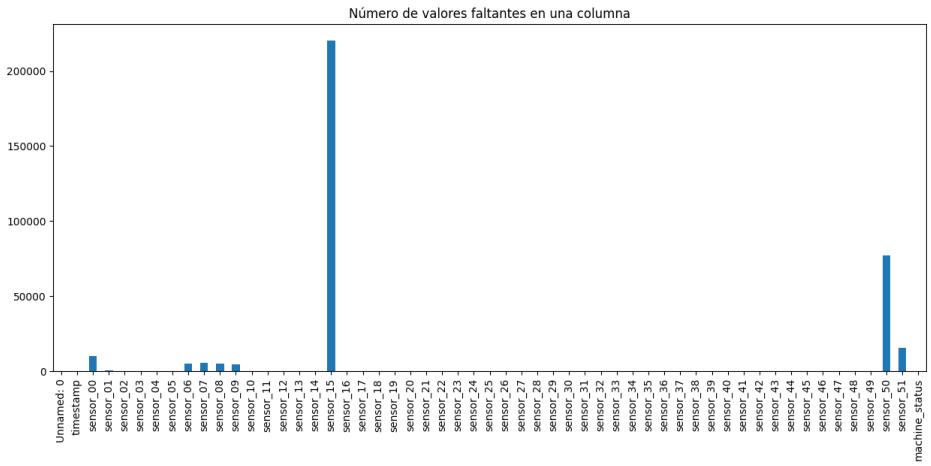
Finalmente, esta fase de análisis estadístico sentó las bases para las siguientes etapas del proyecto donde se implementaron los modelos de Machine Learning con el objetivo de mejorar la eficiencia y eficacia del mantenimiento predictivo en las bombas de agua, permitiendo un enfoque más proactivo y menos reactivo en la gestión de los activos de la planta.

3.2.2. Manejo de valores faltantes (NaN)

En el análisis de los datos recolectados para esta investigación, se identificó la presencia de valores faltantes, específicamente en las columnas correspondiente a los sensores 00, 15, 50 y 51. Tras revisar el conjunto de datos, se determinó que estas columnas no contenían mediciones válidas.

Figura 27

Valores Faltantes (NaN) en los sensores



Nota: Cantidad de NaN en los datos.

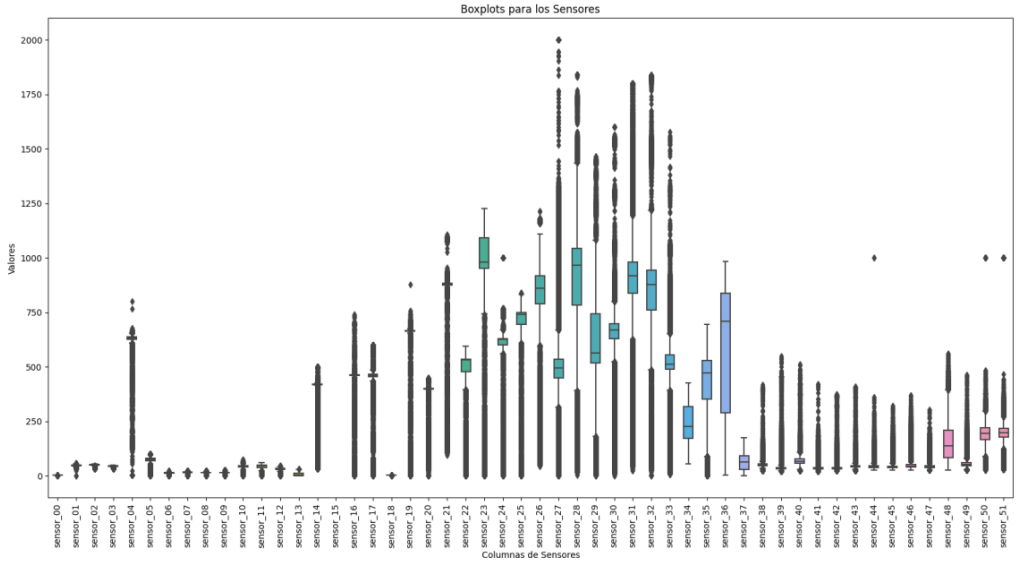
Ya que los demás sensores proporcionaron datos consistentes y trabajables, la única decisión viable fue eliminar las columnas de los sensores 00, 15, 50 y 51. Esta eliminación no afectó negativamente al análisis, ya que los datos restantes eran suficientes para el desarrollo y entrenamiento de los modelos de Machine Learning. El manejo adecuado de los valores faltantes es esencial para asegurar la calidad del conjunto de datos y la precisión de las predicciones en el mantenimiento predictivo. Al eliminar solo los datos irrelevantes, se logró mantener la integridad de la información utilizada en el estudio.

3.2.3. Manejo de valores atípicos (Outliers)

En el análisis de datos, los valores atípicos pueden tener un impacto significativo en los resultados de los modelos de Machine Learning, ya que pueden distorsionar la media y afectar la precisión de las predicciones. Por ello, se realizó una revisión exhaustiva de los datos recolectados para identificar cualquier valor que pudiera considerarse atípico. Se aplicaron métodos estadísticos como la identificación mediante el cálculo de la media y la desviación estándar, así como el uso de boxplots para visualizar la distribución de los datos y detectar posibles outliers.

Figura 28

Boxplot de los sensores



Nota: Cantidad de NaN en los datos.

Se decidió conservar aquellos outliers que eran justificados y que representaban situaciones reales de operación de las bombas de agua. Por ejemplo, ciertos picos en las mediciones podrían reflejar eventos extraordinarios o condiciones operativas específicas que, aunque atípicas, eran relevantes para el análisis. Este enfoque para el manejo de valores atípicos permitió mantener la integridad de los datos utilizados en los modelos de Machine Learning.

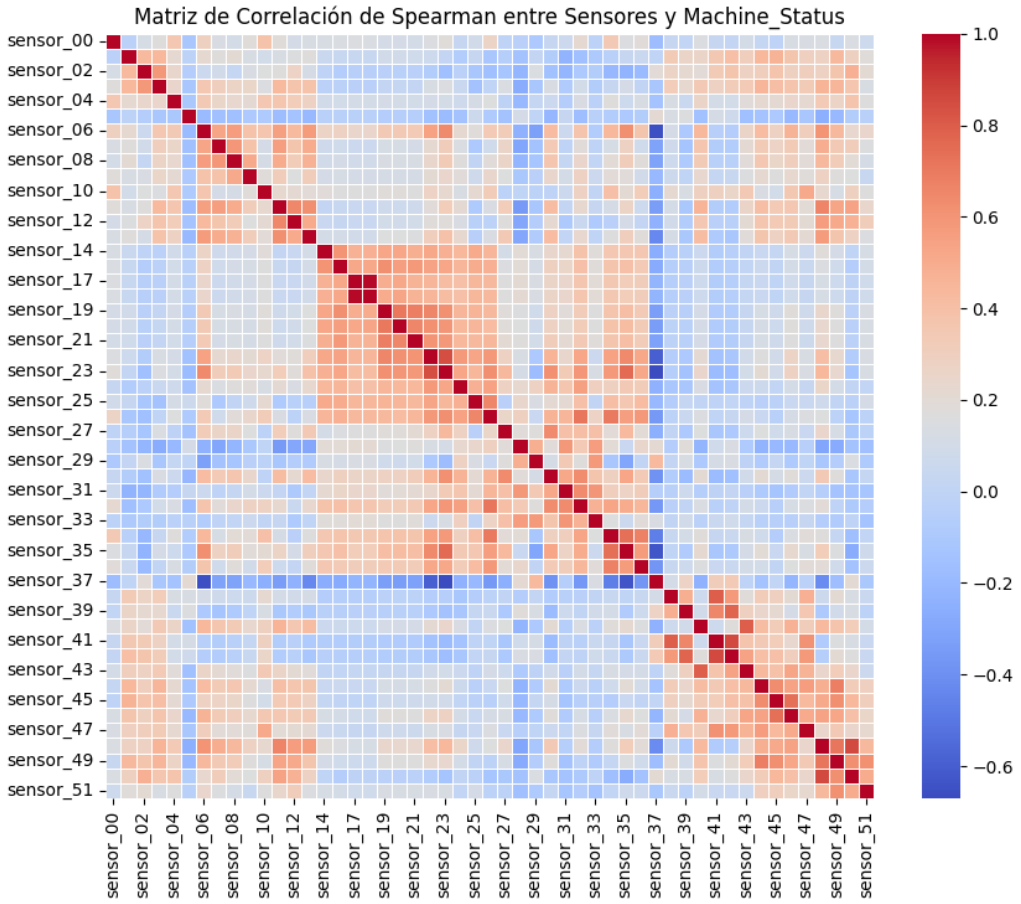
3.2.4. Matriz de correlación

Se realizó la matriz de correlación utilizando la correlación de Spearman para analizar las relaciones entre las variables operacionales de las bombas de agua basadas en las mediciones de los sensores. Este análisis, aunque no era estrictamente necesario para la predicción permitió explorar cómo se interrelacionan diferentes variables y contribuyó a una comprensión más integral de los datos. La elección de la correlación de Spearman fue adecuada ya que se

trabajó con una combinación de variables numéricas y categóricas. Este método evalúa la relación entre las variables a través de sus rangos, evitando suposiciones sobre la normalidad de los datos. La matriz resultante facilitó la identificación de patrones y tendencias, ayudando a visualizar la fuerza y dirección de las correlaciones. A través de este análisis se pudieron identificar relaciones significativas que podrían influir en el comportamiento y el rendimiento de las bombas. Aunque el enfoque principal del estudio es la predicción del mantenimiento la matriz de correlación aportó información valiosa sobre cómo ciertas variables pueden estar relacionadas con los modos de falla.

Figura 29

Matriz de correlación de Spearman aplicado a los datos



Nota: Se uso un mapa de calor para representar las magnitudes.

3.3. SEPARACIÓN DE DATOS DE ENTRENAMIENTO Y DE PRUEBA

La división de los datos en conjuntos de entrenamiento y prueba es un paso crucial en el desarrollo de modelos de machine learning. Como se mencionó en la revisión de la literatura, la razón principal de esta división es evaluar la capacidad del modelo para generalizar correctamente a datos que no ha visto antes, lo que es esencial para asegurar su rendimiento en entornos reales. Si bien no existe una proporción fija, las distribuciones más comunes en la práctica son 50%-50%, 70%-30%, y 80%-20%, dependiendo de la cantidad de datos disponibles y la complejidad del problema.

En este estudio, se optó por utilizar una división del 50%-50%, lo cual significa que la mitad de los datos se utilizó para entrenar los modelos de machine learning, mientras que la otra mitad se destinó para probar su capacidad predictiva. Esta elección se basó en la necesidad de evaluar el desempeño de los algoritmos de manera rigurosa y garantizar que los modelos no se sobreajustaran a los datos de entrenamiento. Además, dado el tamaño del conjunto de datos, con más de 220,000 filas, la proporción de 50%-50% proporcionó una muestra representativa tanto para el entrenamiento como para la evaluación, permitiendo así obtener resultados más robustos.

Tabla 3

Rangos de los datos de entrenamiento y prueba.

	División de datos
Entrenamiento (train dataframe)	Desde el 2019-04-01 00:00:00 hasta 2018-06-09 10:39:00
Prueba (test dataframe)	Desde el 2018-06-09 10:40:00 hasta el 2018-08-17 21:20:00

Nota: En base a las fechas del registro de datos vemos como se dividió los datos para trainig y testing.

3.4. COMPORTAMIENTO DE LA MÁQUINA

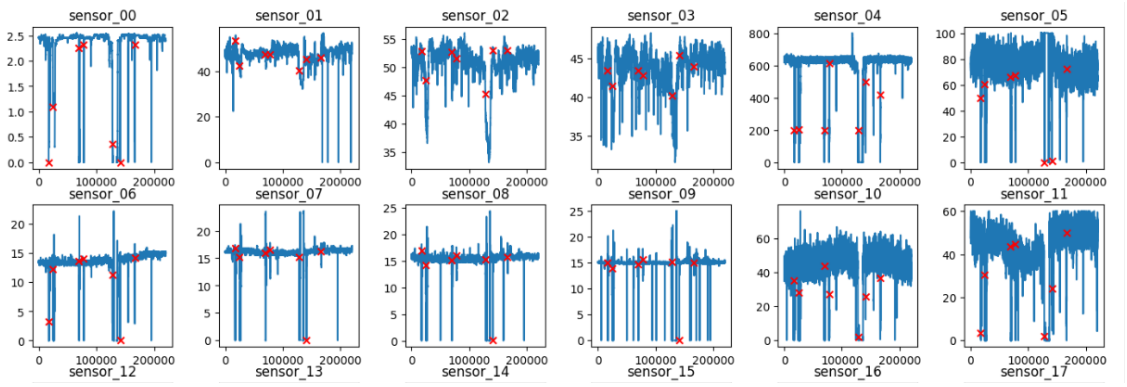
El análisis del comportamiento de la máquina se realizó en base a los registros obtenidos de los sensores ubicados en puntos críticos de la misma. Estos sensores, que son capaces de medir la distancia entre un componente rotatorio y una superficie fija, permiten un monitoreo continuo de la posición relativa y posibles desviaciones. Aunque este análisis no se centrará en espectros de vibración detallados, sí se consideró una evaluación general del comportamiento de la máquina a lo largo del tiempo, observando las tendencias registradas por los sensores.

En este caso, los datos recolectados mostraron la evolución del estado de la máquina en diversas condiciones operativas, registrando parámetros clave como el desplazamiento del eje y otros indicadores de vibración que pueden señalar un posible desbalanceo o desalineación. Según las normas ISO aplicables para turbomáquinas, como la ISO 10816-1 y la ISO 7919-1, el monitoreo de vibraciones es esencial para determinar el estado de salud de la máquina. Aunque no se realizó un análisis espectral, las variaciones significativas en los datos registrados sugieren posibles anomalías que podrían estar asociadas a un desgaste de los componentes.

Además, se evaluó el estado final de cada sensor y su comportamiento en conjunto, utilizando gráficos de tendencia para visualizar las variaciones a lo largo del tiempo. Estas gráficas permitieron identificar patrones y picos en los datos que, aunque no se tradujeron en fallas inmediatas, podrían alertar sobre la necesidad de un mantenimiento preventivo en futuras inspecciones. De esta manera, se logró tener una visión clara de cómo la máquina ha venido operando y las condiciones en las que podría estar desarrollando fallas incipientes, ajustándose a los lineamientos de las normas mencionadas.

Figura 30

Registro de los sensores



Nota: Se marco con una x roja al estado broken.

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

4.1. RESULTADOS

4.1.1. Entrenar algoritmos de machine Learning (regresión lineal y random forest) para las labores de mantenimiento predictivo.

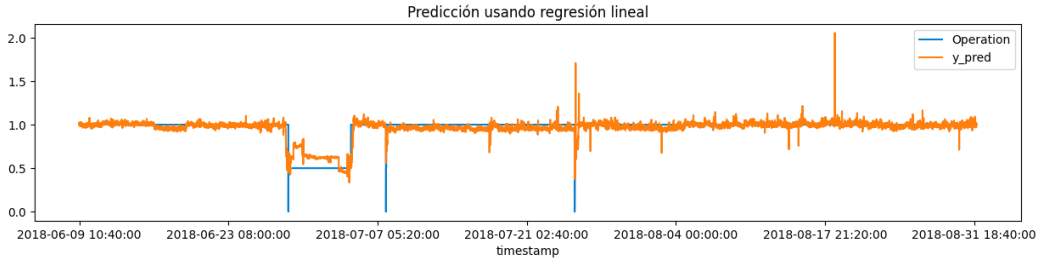
En base a la separación de datos para prueba (Testing) y entrenamiento (Training) se entrenó los algoritmos de Regresión lineal múltiple y Random Forest, cabe aclarar algo, los datos no fueron normalizados ya que a simple vista la tendencia de los datos no es lineal, además que el algoritmo de regresión lineal no es el más óptimo a aplicar en entornos reales ya que, como se dijo al inicio, los datos generalmente siguen otros patrones y es necesario ahorrar recursos computacionales ya que se tiene una gran cantidad de datos. Luego del entrenamiento de datos, otro punto a explicar sería el uso de la validación Walk – Forward en Random forest para el uso en series temporales, la estructura de los datos amerita un análisis de series temporales, Random Forest es capaz de tratar esta estructura de datos, la regresión lineal puede no ser la mejor opción para modelar series temporales, ya que no captura patrones temporales complejos ni la dependencia temporal entre observaciones. Las series temporales a menudo exhiben autocorrelación y tendencias que no son bien modeladas por un enfoque lineal

4.1.2. Analizar los resultados y hacer comparaciones entre algunos algoritmos.

En la primera prueba tomando los datos tal como se mencionó, la regresión lineal nos da el siguiente resultado:

Figura 31

Predicción usando regresión lineal.

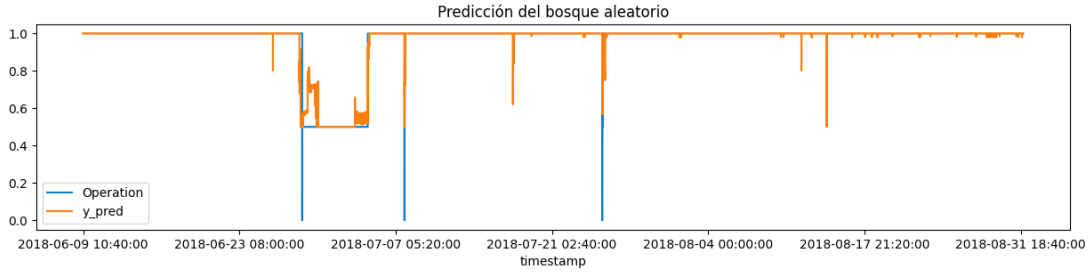


Nota: Gráfica que muestra la comparación entre la predicción y la operación usando Regresión lineal múltiple.

Vemos que y_pred (valor predicho) se asemeja a los registros de operación reales, para el caso de Random Forest se obtuvo lo siguiente:

Figura 32

Predicción del bosque aleatorio.



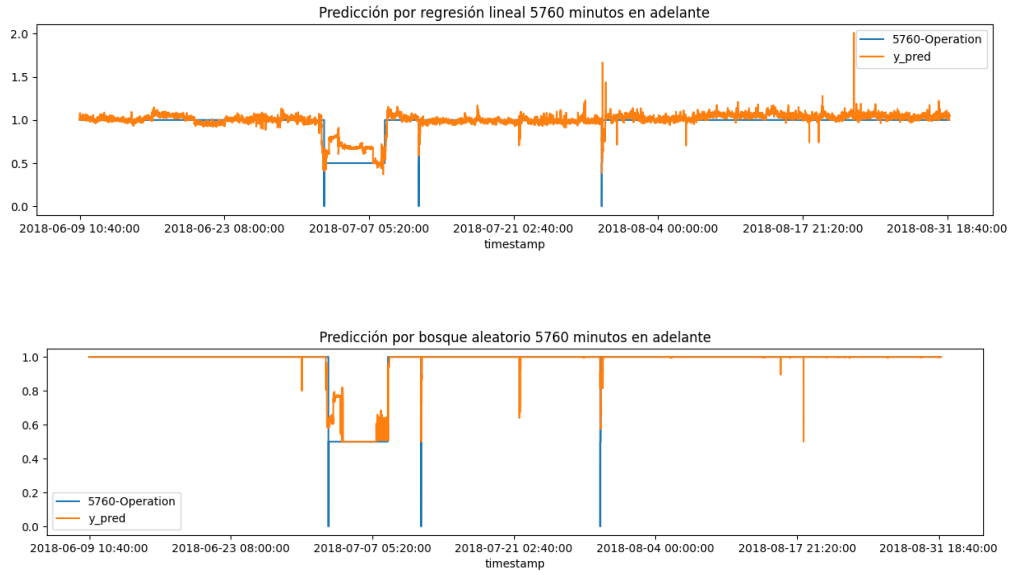
Nota: Gráfica que muestra la comparación entre la predicción y la operación usando Random Forest.

Para Random Forest vemos que y_predición se asemeja más a los registros de operación, como se trata de un problema de series temporales se crearon columnas nuevas que vendrían a representar los valores pasados de los valores en diferentes intervalos de tiempo, se hizo una prueba para 5760 minutos (4 días), estas columnas nuevas están concatenadas con los datos originales, como una

ventana deslizante, las gráficas luego de este tratamiento de datos para regresión lineal y random forest son:

Figura 33

Predicción por regresión lineal y Random Forest 5760 minutos en adelante.



Nota: En las graficas vemos que con el algoritmo de Random Forest se logra un mejor ajuste a los datos.

La diferencia es mínima, pero son importantes para mejorar la precisión del modelo.

Tabla 4

Comparación de métodos de machine learning.

Método	Exactitud	Precisión	Sensibilidad
Random Forest	98,95%	98,33%	99,57%
SVM	99,37%	98,95%	99,78%
Gradient Boosting	99,05%	98,74%	99,36%
XGBoost	99,26%	98,95%	99,57%
KNN	99,05%	98,53%	99,57%
Naive Bayes	98,21%	96,91%	99,57%
Árbol de decisión	98,01%	99,13%	96,83%

Nota: Cuadro comparativo con otros algoritmos del autor Moraga, (2020).

Con la finalidad de determinar si existe algún método de machine learning que proporcione mejor rendimiento en el modelo, se probó 6 métodos los cuales son: máquina de soporte vectorial (SVM), Gradient Boosting, XGBoost, k vecinos más cercanos (KNN), Naive Bayes y árbol de decisión. Cabe acotar que la comparación de los métodos se la realizará con hiperparámetros predeterminados, haciendo uso del conjunto seleccionado de características y solamente se tomará en cuenta las métricas de evaluación.

Realizando un análisis muy minucioso sobre la aplicación del aprendizaje de máquina para la detección de fallos en mantenimiento predictivo, lo que más importancia tiene es que se logren predecir de forma correcta la mayor cantidad de fallos posibles de una máquina y al mismo tiempo disminuir el valor de Falsos Positivos Y Falsos Negativos. La métrica de evaluación que determina la cantidad de predicciones positivas reales que fueron clasificadas correctamente es la sensibilidad, por lo que para determinar que método de aprendizaje de máquina muestra un mejor rendimiento para el mantenimiento predictivo se utilizará esta métrica de evaluación, a su vez se debe tener mucho cuidado en los falsos positivos ya que estos representan un sobre mantenimiento.

4.1.3. Revisar cuál de los dos algoritmos tiene mejor desempeño.

Para saber cuál de los algoritmos es mejor para nuestro caso usamos las métricas de evaluación MAE y RMSE, tanto para la predicción a 4 días como para los datos originales. La diferencia lo vemos en la siguiente tabla:

Tabla 5

Métricas aplicadas a la Regresión lineal.

Regresión lineal	RMSE	MAE
Datos originales	0.059	0.036
Para 4 días	0.070	0.044

Nota: Métricas de evaluacion RMSE y MAE aplicado a la Regresión lineal para saber la precisión del modelo.

Tabla 6

Métricas aplicadas a Randon Forest.

Random Forest	RMSE	MAE
Datos originales	0.036	0.005
Para 4 días	0.037	0.006

Nota: Métricas de evaluación RMSE y MAE aplicado a Random Forest para saber la precisión del modelo

RMSE (Error cuadrático medio) mide la magnitud promedio de los errores al cuadrado, lo que significa que penaliza más los errores grandes. Un valor más bajo de RMSE indica que el modelo predice con mayor precisión. Random Forest tiene un RMSE más bajo (0.036 vs 0.059), lo que sugiere que, en promedio, comete menos errores grandes comparado con la regresión lineal múltiple.

MAE (Error absoluto medio) mide el error promedio de las predicciones, sin dar más peso a los errores grandes. Nuevamente, Random Forest tiene un MAE más bajo (0.005 vs 0.036), lo que indica que tiene un desempeño más consistente en cuanto a precisión general.

Lo mismo ocurre con los datos RMSE y MAE para 4 días, Random Forest es el algoritmo más preciso para este caso.

4.1.4. Lograr el aprendizaje automático según los datos del equipo para lograr predecir el estado de la maquina en el futuro inmediato.

Se logró el aprendizaje mediante Regresión lineal múltiple y Random Forest según los datos de la bomba, este nos muestra el estado de la bomba, pero no nos muestra el modo de falla, cabe aclarar esto, para esto es necesario un análisis de espectro y de envolvente de cada punto monitoreado por los sensores automáticos, además que, para esto es necesario contar con otro tipo de sensor, un acelerómetro que nos muestre más tipos de ondas, tanto del desplazamiento, velocidad y aceleración. En este trabajo se cumplió con el objetivo de aplicar algoritmos de Machine Learning para que aprenda de manera general cuando el equipo va a fallar, siendo una herramienta útil para tomar decisiones al momento de estimar paradas de equipos para mantenimiento preventivo.

4.1.5. Contrastación de hipótesis

La aplicación de los algoritmos de regresión lineal y random forest al mantenimiento predictivo permitirá una mejora significativa en la precisión y eficiencia en la detección temprana de fallas en equipos industriales en comparación con los métodos tradicionales empleados en la industria. Además, que al usar los dos métodos sabremos cuál de los dos es el más óptimo según la estructura de datos que se tiene.

Tabla 7

Contrastación de la hipótesis.

Prueba de ANOVA		
		variable1media (Agrupada)
N		22
Parámetros normales[a,b]	Media	2,32
	Desv. Desviación	,477
Máximas diferencias extremas	Absoluto	,430
	Positivo	,430
	Negativo	-,252
Estadístico de prueba		,430
Sig. asintótica(bilateral)		,000[c]

a. La distribución de prueba es normal.

b. Se calcula a partir de datos.

c. Corrección de significación de Lilliefors.

Nota: Prueba ANOVA aplicada al trabajo.

En la tabla 7 de la contrastación de la hipótesis general se acepta la hipótesis que la prueba de la muestra fue: Sig. Bilateral = 0.000 por ende, incidió significativamente en la precisión y eficiencia en la detección temprana de fallas en equipos industriales en comparación con los métodos tradicionales empleados en la industria.

4.2. DISCUSIÓN

El objetivo de este trabajo fue aplicar algoritmos de Machine Learning para predecir de manera general el momento en que un equipo podría fallar proporcionando así una herramienta útil para planificar paradas y optimizar el mantenimiento preventivo. En su investigación, Soto (2021) determinó que el tango de funcionamiento normal de los equipos se encuentra entre 1780 y 6200 RPM mientras que el rango de 6201 a 9500 RPM fue utilizado para la simulación de fallas. El autor presentó dos modelos predictivos uno basado en la regresión lineal de Bayes y otro en el modelo KNN, diseñados para predecir nuevos datos provenientes del funcionamiento de las bombas. Aunque su trabajo se centra en simulaciones, Soto destaca que su propuesta constituye una estrategia de mantenimiento predictivo aplicable en entornos reales. Vilema (2022) concluyó en su estudio que utilizó el 75% de los datos totales para entrenamiento y el 25% restante para pruebas. Tras seguir 6 pasos en su metodología demostró que el modelo mejora su rendimiento al emplear un conjunto de características seleccionadas y ajustar los hiperparámetros de manera óptima. En su conclusión destaco que el modelo presentó un buen desempeño en la detección de fallas. Finalmente recomendó que una adecuada preparación de los datos es fundamental para obtener resultados satisfactorios.

Reveco (2019) en su investigación desarrollo un modelo para los motores diésel Cummins, modelo QSK60 HPI, utilizados en la flota de camiones Komatsu 930E. El modelo se basó en datos provenientes del análisis de muestras de aceite que fueron empleados para entrenar algoritmos de Machine Learning. En la conclusión de su trabajo destacó que los algoritmos de clasificación utilizados Multiclass Decision Jungle y Forest lograron buenos resultados.

Huamán (2021) demuestra que el algoritmo más efectivo para el diagnóstico del índice de salud de los interruptores de potencia fue Random Forest Classifier con una

precision del 99.27 % mientras que la lógica Fuzzy obtuvo 2.61%. En ese sentido machine learning demuestra ser una herramienta muy importante en el ámbito industrial, además que según su análisis económico de costos concluye que usar modelos predictivos es más rentable que usar el modelo fuzzy.

En el trabajo de Sanchez y Rivera (2024) obtiene un resultado del 85.4% para los modelos de Machine Learning, siendo un valor aceptable, aunque menciona que la lógica para calcular esta métrica o indicador cambia de acuerdo a la operación del equipo, esto demuestra la correlación que existe entre una buena operación, buena recolección de datos y un entrenamiento más acertado, más preciso. En la investigación de Sánchez (2021) menciona que selecciono el algoritmo de Random Forest debido a su destacado rendimiento. Al final de su trabajo realizo una comparación de los resultados obtenidos con los de otros investigadores que abordaron el mismo problema, destacando la eficacia de su enfoque. Bartolomé (2018) concluyó que al igual que en otras investigaciones, uno de los principales objetivos del mantenimiento predictivo es reducir los costos asociados a mantenimientos inesperados, aprovechando las herramientas de la industria 4.0. Por su parte, Contreras (2020) propuso implementar un método de mantenimiento predictivo basado en IA para monitorear el estado de los motores y predecir el momento óptimo para el reemplazo de sus componentes, evitando paradas inesperadas de planta. Este enfoque busca aumentar la competitividad de las industrias que adopten estos sistemas de gestión de mantenimiento. En su trabajo, Contreras uso el aprendizaje supervisado de Machine Learning para predecir el estado de los rodamientos en motores de inducción.

V. CONCLUSIONES

El uso de regresión lineal múltiple al ser un algoritmo para regresión o para predecir el comportamiento de las variables de la base de datos es útil siempre que la distribución de las variables sea de forma lineal o se asemeje a esta, aunque también podemos adaptar un algoritmo de regresión lineal a uno polinomial la naturaleza de una serie temporal obliga a considerar la correlación que tienen los datos a través del tiempo, según la estructura de los datos se escoge el algoritmo a usar, ya que algunos permiten ciertas libertades y otros tienen más restricciones al momento de trabajar. Para el caso de random forest el algoritmo permite trabajar con datos no normalizados y además con outliers, en esa parte es conveniente y facilita a un preprocesamiento de datos, pero no es un algoritmo para series temporales, en ese caso random forest usa una técnica llamada ventana deslizante que de alguna manera permite trabajar con series temporales.

Vemos que es posible predecir el comportamiento de una maquina según sus parámetros como alternativa a los análisis de mantenimiento predictivo normalmente usados, obviamente esto debe de ser comprobado técnicamente en ambientes laborales reales para ver si las predicciones aciertan y en qué porcentaje de precisión lo hacen a través del tiempo en que sean probados. Actualmente en las pequeñas industrias y en algunas grandes se subestima el hecho de mantener un buen registro de datos, no solo datos técnicos como la temperatura, presión, etc. Sino que debería de registrarse la mayor cantidad de datos como el tiempo medio entre fallas, tiempo medio para fallar, tiempo medio para reparar, etc. Vimos en este trabajo que Random Forest es el algoritmo que mejores resultados nos da.

Como vimos en los gráficos de los sensores las fallas a veces ocurren en tiempos donde algunos sensores registran datos normales y otros no, con la cantidad de datos que

se tiene esta acción sería muy difícil de realizar para una persona, con la implementación de un análisis de datos tradicional o con IA reducimos el tiempo de análisis mejorando la eficiencia del encargado de mantenimiento, estas técnicas deben de ser tomadas con criterio ya que algunas veces una precisión buena no significa que todo este mejorando, sobre todo en el caso de machine learning esto algunas veces indica un sobreajuste y como se mencionó, hay algoritmos que trabajan mejor según la estructura de los datos.

Se concluye que en la tendencia actual, las industrias son cada vez más automáticas y por lo tanto la disponibilidad de información relevante es mayor, con el internet de las cosas y la industria 4.0 las empresas cuentan con la posibilidad de tener un historial de funcionamiento, datos que aunque parezca que solo ocupan espacio en las memorias pueden ser usados a favor del crecimiento empresarial y la mejora continua, para mejorar el rendimiento de este análisis es mejor tener una base de datos grande para que los algoritmos o el algoritmo usado sea más robusto al momento de hacer las predicciones y no tenga un sobreajuste en sus respuestas

VI. RECOMENDACIONES

Tanto para hacer un análisis con machine learning como para tener buenos registros del área de mantenimiento dentro de una empresa es necesario tener y registrar la mayor cantidad de datos posible, las industrias en la región de Puno y en todo el Perú deberían de dar más importancia al análisis de datos sobre todo si el objetivo es hacer crecer una empresa, estas técnicas tienen la posibilidad de mejorar el área de mantenimiento haciendo que se tomen mejores decisiones antes de realizar una parada de planta o de un equipo aumentando la disponibilidad de estos y la confiabilidad.

Otra opción para este caso es usar el modelo ARIMA para análisis de series temporales, que es lo que se usa comúnmente, este modelo estadístico trabaja y está enfocada en estructuras de datos como el que vimos en este trabajo, en caso que una empresa busque la solución más practica al análisis de datos regresivo para una serie temporal sin tener que acudir a técnicas nuevas o no que no estén bien implementadas sería la mejor opción.

Para automatizar aún más este proceso de análisis de datos y enfocándonos en el algoritmo más adecuado para una base de datos con una estructura de serie temporal debemos de entrar a Deep learning, específicamente a las redes neuronales recurrentes RNN que es un modelo que trabaja mejor con series temporales, aunque el desarrollo de un modelo de Deep learning no está considerado dentro de este trabajo en mi opinión este sería la mejor opción si se quiere usar técnicas de IA.

VII. REFERENCIA BIBLIOGRÁFICA

Albornoz Cabello, G. A. (2021). *Aplicación del aprendizaje automático supervisado en el mantenimiento predictivo de los motores eléctricos de inducción en las empresas mineras del Perú [Tesis de licenciatura, Universidad Nacional del Centro del Perú]*. Repositorio intitucional. Obtenido de https://repositorio.uncp.edu.pe/handle/20.500.12894/7567

Bartolomé Aramburu, R. (2018). *Mantenimiento predictivo de los equipos industriales mediante el uso de la inteligencia artificial [Tesis de licenciatura, Universidad Politécnica de Cataluña]*. Repositorio Intitucional. Obtenido de https://upcommons.upc.edu/handle/2117/167239

Bock, T. (s.f.). *www.displayr.com*. Obtenido de https://www.displayr.com/what-is-a-correlation-matrix/

Condori Arias, E. F. (2010). *Roconocimiento y clasificación de objetos usando inteligencia artificial basada en SVM y visión estereoscópica [Tesis de licenciatura, Universidad Nacional de Ingenieria]*. Repositorio institucional. Obtenido de https://cybertesis.uni.edu.pe/handle/20.500.14076/2093

ConMan. (26 de Octubre de 2017). *Mathematics Stack Exchange*. Obtenido de https://math.stackexchange.com/q/2491589

Contreras Alvarez, J. L. (2020). *Diseño de un modelo para mantenimiento predictivo en motores de inducción utilizando técnicas de la industria 4.0 [Tesis de licenciatura, Universidad Tecnologica del Perú]*. Repositorio Institucional. Obtenido de https://repositorio.utp.edu.pe/handle/20.500.12867/4275

Géron, A. (2019). *Hands-on Machine Learing with Scikit-learn, Keras & TensorFlow.* (Second edition ed.). Sebastopol, United States: O'Reilly Media, Inc.

IBM Cluod education. (15 de Julio de 2020). *ibm.com*. Obtenido de https://www.ibm.com/ar-es/cloud/learn/machine-learning

Mataix, C. (1975). *Turbomaquinas Hidraulicas.* Madrid: ICAI.

Medrano Márquez, J. Á., González Ajuech, V. L., & Díaz de León Santiago, V. M. (2017). *Mantenimiento Técnicas y aplicaciones industriales.* México: Grupo editorial patria S.A.

Mobley, R. K. (2002). *An Introduction to predictive maintenance* (Second edition ed.). United States of America: Butterworth-Heinemann.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). *Foundations of Machine Learning* (second edition ed.). Cambridge: The MIT Press.

ORACLE. (s.f.). *docs.oracle.com*. Obtenido de https://docs.oracle.com/cloud/help/es/pbcs_common/PFUSU/insights_metrics_RMSE.htm#PFUSU-GUID-FD9381A1-81E1-4F6D-8EC4-82A6CE2A6E74

Ponce Cruz, P. (2010). *Inteligencia Artificial con aplicaciones a la ingenieria.* Mexico: Alfaomega.

Raschka , S. (2018). *pages.stat.wisc.edu*. Obtenido de https://sebastianraschka.com/pdf/lecture-notes/stat479fs18/02_knn_notes.pdf

Raschka, S., & Mirjalili, V. (2019). *Python Machine Learning* (2° edición ed.). (S. Llena, Trad.) España: Marcombo S.A.

Reveco Diaz, M. I. (2019). *Analisis predictivo de activos mineros para la obtención de intervalo de falla mediante algoritmos de machine learning [Tesis de licenciatura, Universidad de Chile]*. Repositorio Institucional. Obtenido de https://repositorio.uchile.cl/handle/2250/173707

Rulemanes Alvear S.A. (s.f.). *rulemanesalvear.com.ar*. Obtenido de https://rulemanesalvear.com.ar/termografia/

Sánchez Martín, D. J. (2021). *Mantenimiento predictivo, Machine learning para la detección automatizada de fallas [Tesis de licenciatura, Universidad Complutense de Madrid]*. Repositorio Institucional. Obtenido de https://eprints.ucm.es/id/eprint/68126/

Shai Shalev, S., & Shai Ben, D. (2014). *Understanding Machine Learning from theory to algorithms.* New York: Cambridge University Press.

SINTECH PUMPS. (17 de Junio de 2021). *sintechpumps.com*. Obtenido de https://www.sintechpumps.com/bombas/que-son-las-bombas-de-turbina-vertical-y-como-funcionan/?lang=es

Soto Zabala, D. F. (2021). *Implementacion de mantenimiento predictivo para una bomba centrifuga utilizando Machine Learning [Tesis de licenciatura, Universidad Antonio Nariño]*. Repositorio Intitucional. Obtenido de http://repositorio.uan.edu.co/handle/123456789/5154

Vapnik N, V. (2000). *The Nature of Statistical learning theory*. New York: Springer.

Vilema Lara, P. H. (2022). *Detección de fallos en mantenimiento predictivo usando el método de aprendizaje de maquina random forest [tesis de licenciatura, Escuela Superior Politécnica de Chimborazo]*. Repositorio institucional. Obtenido de http://dspace.espoch.edu.ec/handle/123456789/16950

ANEXOS

ANEXO 1: Código realizado en google colab para la elaboración de este trabajo

Machine Learning aplicado en mantenimiento predictivo

Para empezar con este trabajo se importará las librerías necesarias como numpy, pandas, matplotlib, seaborn y sklearn. Para esto se hace la siguiente operación.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from statsmodels.tsa.holtwinters import SimpleExpSmoothing

from sklearn.preprocessing import LabelEncoder, MinMaxScaler
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestClassifier, RandomForestRegressor
from sklearn.metrics import mean_squared_error

from statsmodels.tsa.stattools import adfuller
```

Cargamos los datos

```
df = pd.read_csv('sensor.csv')
```

Exploramos los datos, los visualizamos y sacamos las características estadísticas más importantes

```
df.head()
```

	Unnamed: 0	timestamp	sensor_00	sensor_01	sensor_02	sensor_03	sensor_04	sensor_05	sensor_06	sensor_07	...	sensor_43	sensc
0	0	2018-04-01 00:00:00	2.465394	47.09201	53.2118	46.310760	634.3750	76.45975	13.41146	16.13136	...	41.92708	39.64
1	1	2018-04-01 00:01:00	2.465394	47.09201	53.2118	46.310760	634.3750	76.45975	13.41146	16.13136	...	41.92708	39.64
2	2	2018-04-01 00:02:00	2.444734	47.35243	53.2118	46.397570	638.8889	73.54598	13.32465	16.03733	...	41.66666	39.3
3	3	2018-04-01 00:03:00	2.460474	47.09201	53.1684	46.397568	628.1250	76.98898	13.31742	16.24711	...	40.88541	39.0
4	4	2018-04-01 00:04:00	2.445718	47.13541	53.2118	46.397568	636.4583	76.58897	13.35359	16.21094	...	41.40625	38.77

5 rows × 55 columns

```
df.shape
```

(220320, 55)

```
df.info()
```

```
---  ------          --------------   -----
 0   Unnamed: 0      220320 non-null  int64
 1   timestamp       220320 non-null  object
 2   sensor_00       210112 non-null  float64
 3   sensor_01       219951 non-null  float64
 4   sensor_02       220301 non-null  float64
 5   sensor_03       220301 non-null  float64
 6   sensor_04       220301 non-null  float64
 7   sensor_05       220301 non-null  float64
 8   sensor_06       215522 non-null  float64
 9   sensor_07       214869 non-null  float64
 10  sensor_08       215213 non-null  float64
 11  sensor_09       215725 non-null  float64
 12  sensor_10       220301 non-null  float64
 13  sensor_11       220301 non-null  float64
 14  sensor_12       220301 non-null  float64
 15  sensor_13       220301 non-null  float64
 16  sensor_14       220299 non-null  float64
 21  sensor_19       220304 non-null  float64
 22  sensor_20       220304 non-null  float64
 23  sensor_21       220304 non-null  float64
 24  sensor_22       220279 non-null  float64
 25  sensor_23       220304 non-null  float64
 26  sensor_24       220304 non-null  float64
 27  sensor_25       220284 non-null  float64
 28  sensor_26       220300 non-null  float64
 29  sensor_27       220304 non-null  float64
 30  sensor_28       220304 non-null  float64
 31  sensor_29       220248 non-null  float64
 32  sensor_30       220059 non-null  float64
 33  sensor_31       220304 non-null  float64
 34  sensor_32       220252 non-null  float64
 35  sensor_33       220304 non-null  float64
 36  sensor_34       220304 non-null  float64
 37  sensor_35       220304 non-null  float64
 38  sensor_36       220304 non-null  float64
 39  sensor_37       220304 non-null  float64
 40  sensor_38       220293 non-null  float64
 41  sensor_39       220293 non-null  float64
 42  sensor_40       220293 non-null  float64
 43  sensor_41       220293 non-null  float64
 44  sensor_42       220293 non-null  float64
 45  sensor_43       220293 non-null  float64
 46  sensor_44       220293 non-null  float64
 47  sensor_45       220293 non-null  float64
 48  sensor_46       220293 non-null  float64
 49  sensor_47       220293 non-null  float64
 50  sensor_48       220293 non-null  float64
 51  sensor_49       220293 non-null  float64
 52  sensor_50       143303 non-null  float64
 53  sensor_51       204937 non-null  float64
 54  machine_status  220320 non-null  object
dtypes: float64(52), int64(1), object(2)
memory usage: 92.5+ MB
```

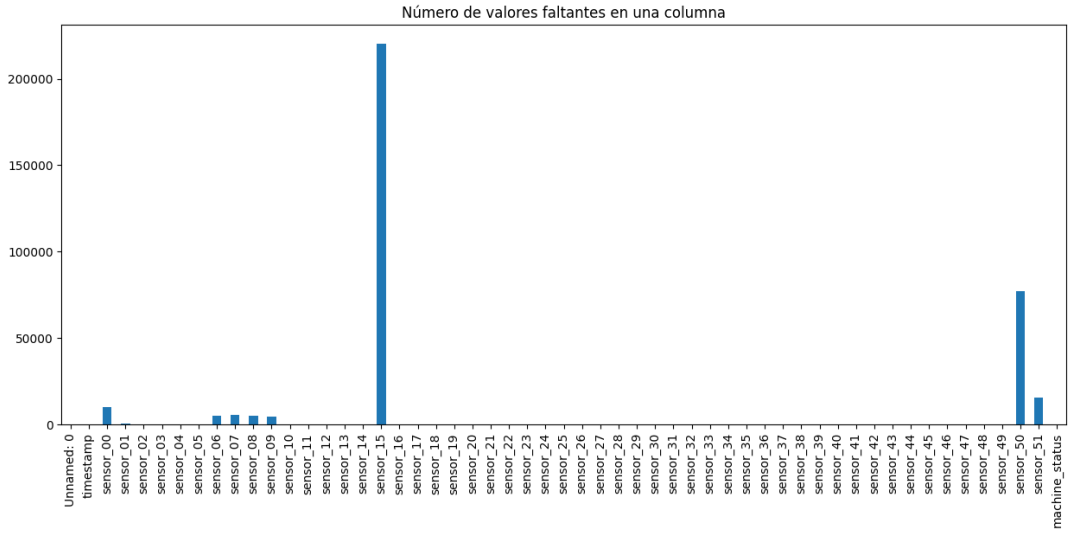
Con el comando .info podemos saber las características de nuestros datos, como vemos en el resumen para el sensor 15 se tienen datos vacíos.

	count	mean	std	min	25%	
Unnamed: 0	220320.0	110159.500000	63601.049991	0.000000	55079.750000	110159.500
sensor_00	210112.0	2.372221	0.412227	0.000000	2.438831	2.456
sensor_01	219951.0	47.591611	3.296666	0.000000	46.310760	48.133
sensor_02	220301.0	50.867392	3.666820	33.159720	50.390620	51.649
sensor_03	220301.0	43.752481	2.418887	31.640620	42.838539	44.227
sensor_04	220301.0	590.673936	144.023912	2.798032	626.620400	632.638
sensor_05	220301.0	73.396414	17.298247	0.000000	69.976260	75.576
sensor_06	215522.0	13.501537	2.163736	0.014468	13.346350	13.642
sensor_07	214869.0	15.843152	2.201155	0.000000	15.907120	16.167
sensor_08	215213.0	15.200721	2.037390	0.028935	15.183740	15.494
sensor_09	215725.0	14.799210	2.091963	0.000000	15.053530	15.082
sensor_10	220301.0	41.470339	12.093519	0.000000	40.705260	44.291
sensor_11	220301.0	41.918319	13.056425	0.000000	38.856420	45.363
sensor_12	220301.0	29.136975	10.113935	0.000000	28.686810	32.515
sensor_13	220301.0	7.078858	6.901755	0.000000	1.538516	2.929
sensor_14	220299.0	376.860041	113.206382	32.409550	418.103250	420.106
sensor_15	0.0	NaN	NaN	NaN	NaN	N
sensor_16	220289.0	416.472892	126.072642	0.000000	459.453400	462.856
sensor_17	220274.0	421.127517	129.156175	0.000000	454.138825	462.020
sensor_18	220274.0	2.303785	0.765883	0.000000	2.447542	2.533
sensor_19	220304.0	590.829775	199.345820	0.000000	662.768975	665.672
sensor_20	220304.0	360.805165	101.974118	0.000000	398.021500	399.367
sensor_21	220304.0	796.225942	226.679317	95.527660	875.464400	879.697
sensor_22	220279.0	459.792815	154.528337	0.000000	478.962600	531.855
sensor_23	220304.0	922.609264	291.835280	0.000000	950.922400	981.925
sensor_24	220304.0	556.235397	182.297979	0.000000	601.151050	625.873
sensor_25	220284.0	649.144799	220.865166	0.000000	693.957800	740.203
sensor_26	220300.0	786.411781	246.663608	43.154790	790.489575	861.869
sensor_27	220304.0	501.506589	169.823173	0.000000	448.297950	494.468
sensor_28	220304.0	851.690339	313.074032	4.319347	782.682625	967.279
sensor_29	220248.0	576.195305	225.764091	0.636574	518.947225	564.872
sensor_30	220059.0	614.596442	195.726872	0.000000	627.777800	668.981
sensor_31	220304.0	863.323100	283.544760	23.958330	839.062400	917.708
sensor_32	220252.0	804.283915	260.602361	0.240716	760.607475	878.850
sensor_33	220304.0	486.405980	150.751836	6.460602	489.761075	512.271
sensor_34	220304.0	234.971776	88.376065	54.882370	172.486300	226.356
sensor_35	220304.0	427.129817	141.772519	0.000000	353.176625	473.349
sensor_36	220304.0	593.033876	289.385511	2.260970	288.547575	709.668
sensor_37	220304.0	60.787360	37.604883	0.000000	28.799220	64.295
sensor_38	220293.0	49.655946	10.540397	24.479166	45.572910	49.479
sensor_39	220293.0	36.610444	15.613723	19.270830	32.552080	35.416
sensor_40	220293.0	68.844530	21.371139	23.437500	57.812500	66.406
sensor_41	220293.0	35.365126	7.898665	20.833330	32.552080	34.895
sensor_42	220293.0	35.453455	10.259521	22.135416	32.812500	35.156
sensor_43	220293.0	43.879591	11.044404	24.479166	39.583330	42.968
sensor_44	220293.0	42.656877	11.576355	25.752316	36.747684	40.509
sensor_45	220293.0	43.094984	12.837520	26.331018	36.747684	40.219
sensor_46	220293.0	48.018585	15.641284	26.331018	40.509258	44.849

sensor_47	220293.0	44.340903	10.442437	27.199070	39.062500	42.534
sensor_48	220293.0	150.889044	82.244957	26.331018	83.912030	138.020
sensor_49	220293.0	57.119968	19.143598	26.620370	47.743060	52.662
sensor_50	143303.0	183.049260	65.258650	27.488426	167.534700	193.865

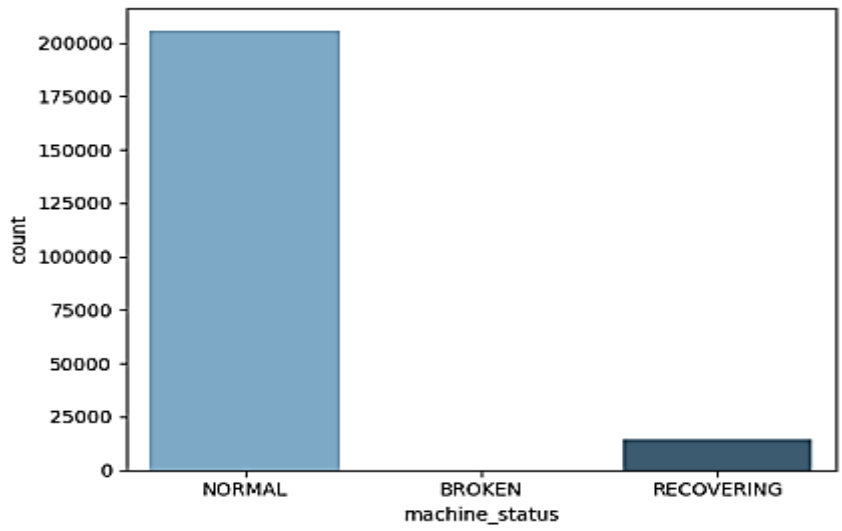
Para poder ver mejor los valores faltantes se hará lo siguiente:

```
df.isnull().sum().plot(kind='bar', figsize=(15,6));
plt.title('Número de valores faltantes en una columna');
```



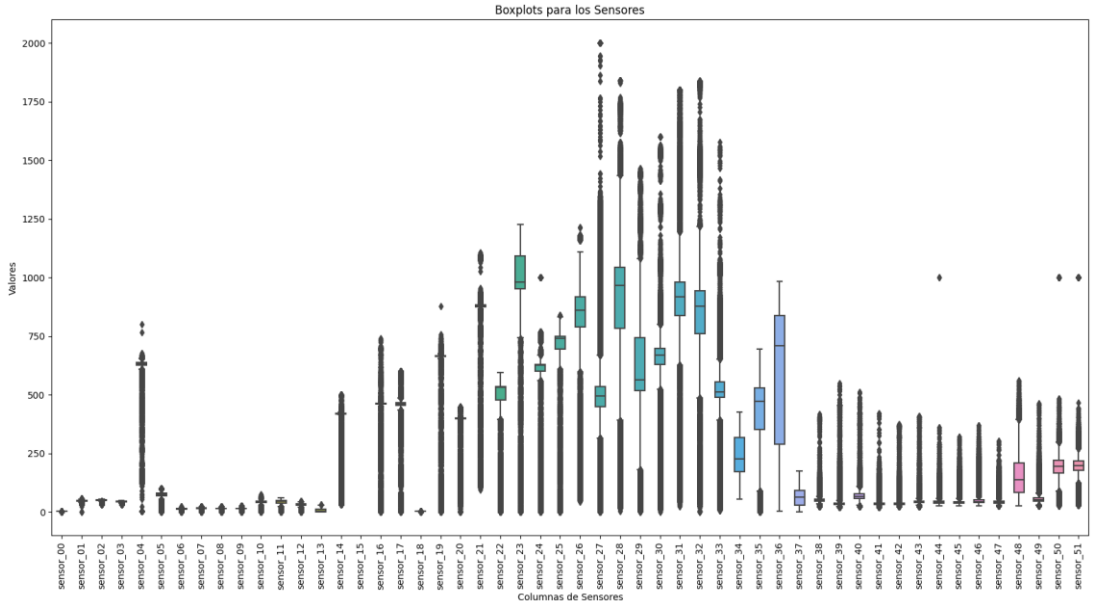

```
print(df.machine_status.value_counts())
sns.countplot(x='machine_status',data=df,palette='Blues_d');
```

```
NORMAL        205836
RECOVERING     14477
BROKEN             7
Name: machine_status, dtype: int64
```



Para ver la presencia de outliers se usa el siguiente código.

```
columnas_sensor = df.filter(like='sensor_').columns

plt.figure(figsize=(20, 10))

sns.boxplot(data=df[columnas_sensor], orient='v', width=0.5)

plt.xticks(rotation=90)
plt.xlabel('Columnas de Sensores')
plt.ylabel('Valores')
plt.title('Boxplots para los Sensores')

plt.show()
```

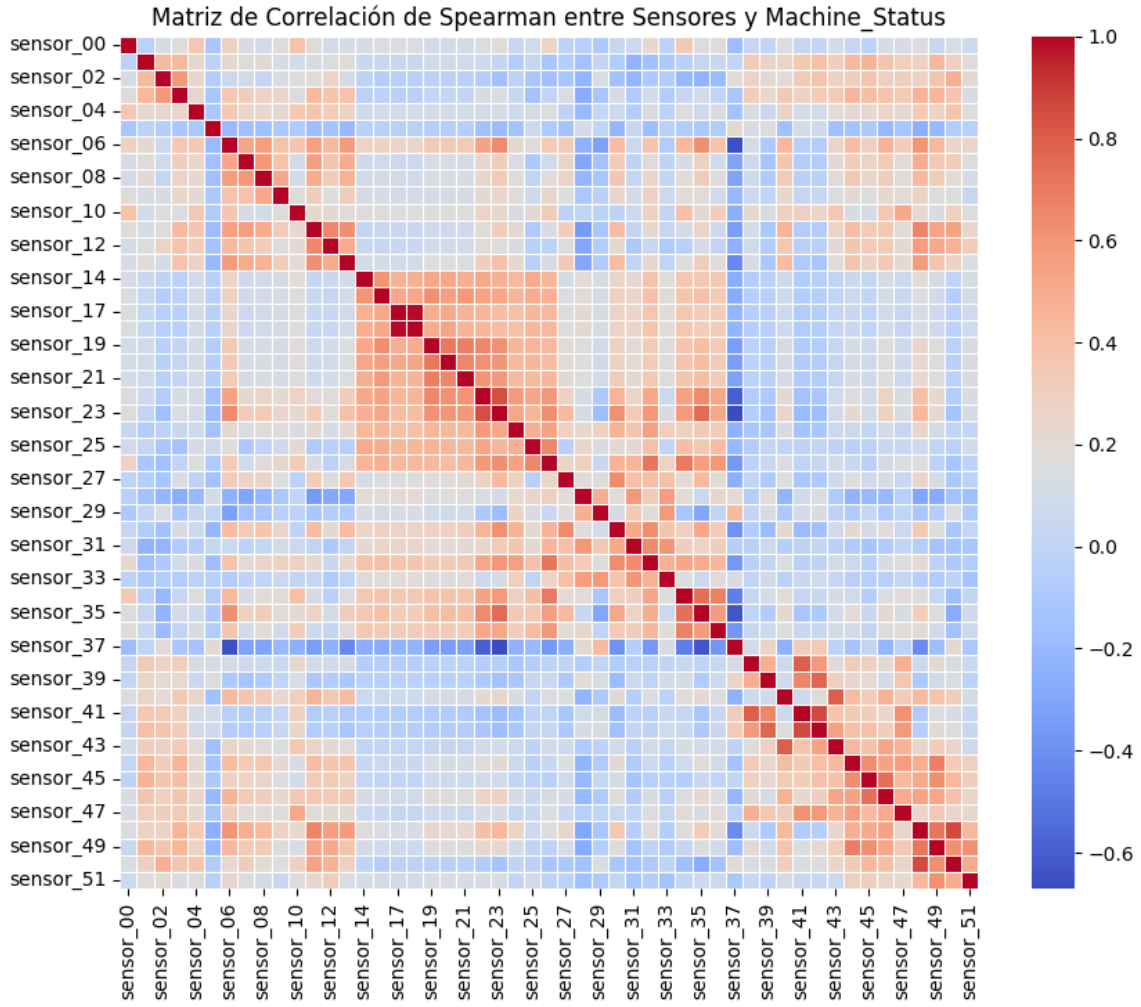



```
broken= df[df['machine_status']=='BROKEN']

df_2 = df.drop(['machine_status'],axis=1)
names= df_2.columns
```

Ahora veremos la correlación

```
columnas_sensores = [col for col in df.columns if 'sensor_' in col]

if 'sensor_15' in df.columns:
    df = df.drop('sensor_15', axis=1)


df_encoded = pd.get_dummies(df, columns=['machine_status'], drop_first=True)

# Obtener las columnas específicas para 'recovering', 'broken' y 'normal'
columnas_interes = [col for col in df_encoded.columns if 'machine_status' in col]

correlation_matrix_spearman = df_encoded[columnas_sensores +
columnas_interes].corr(method='spearman')
```

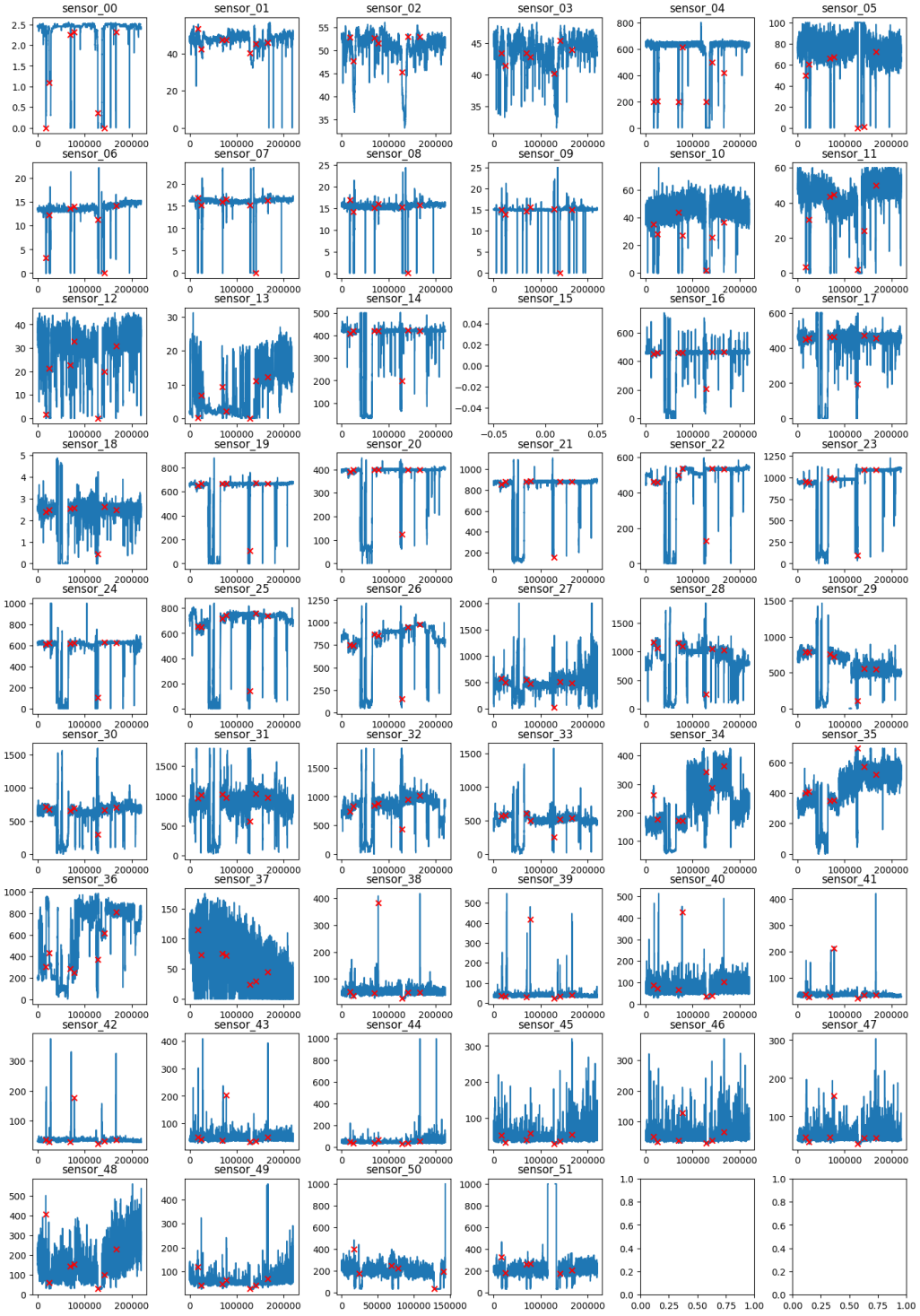
```
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix_spearman, annot=False, cmap='coolwarm', fmt='.2f',
linewidths=0.5)
plt.title('Matriz de Correlación de Spearman entre Sensores y Machine_Status')
plt.show()
```



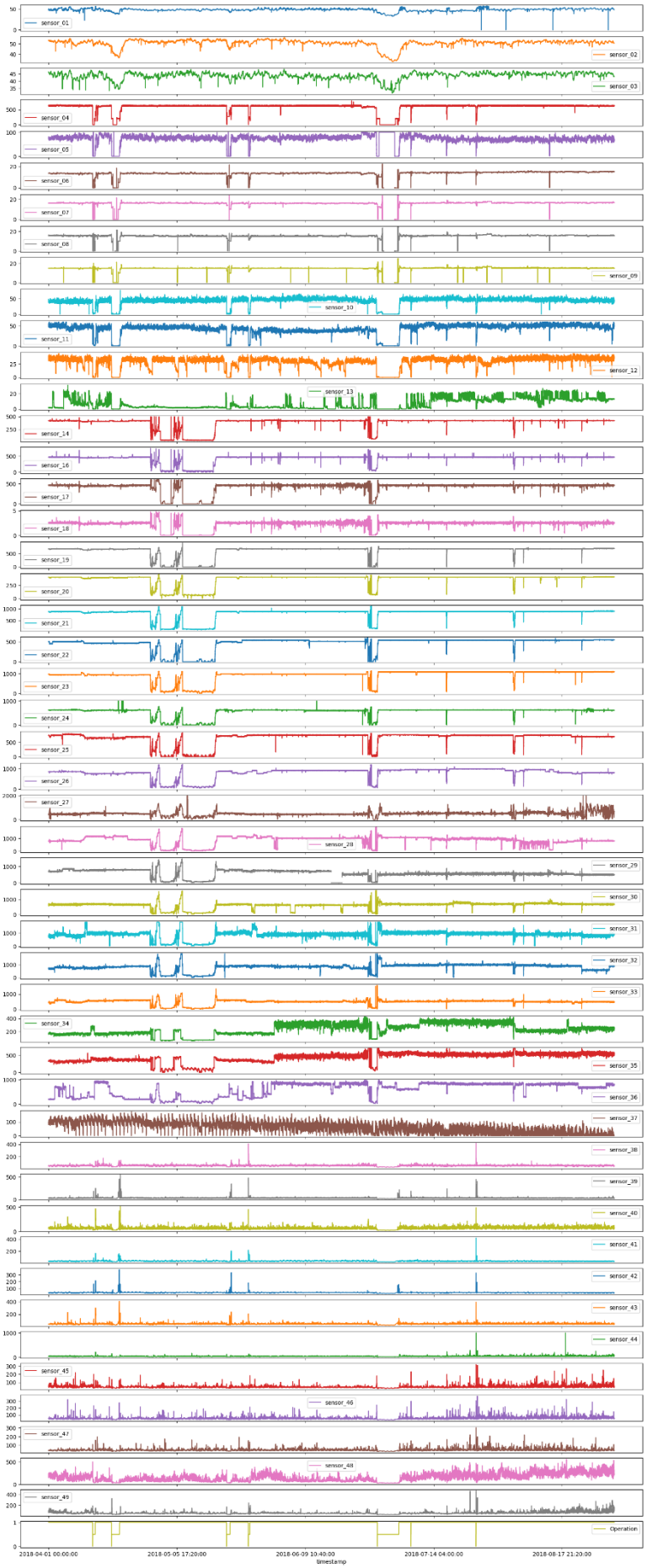
Ahora veremos una inspección visual de los registros de cada sensor.

```
conditions = [(df['machine_status'] =='NORMAL'), (df['machine_status'] =='BROKEN'),
(df['machine_status'] =='RECOVERING')]
choices = [1, 0, 0.5]
df['Operation'] = np.select(conditions, choices, default=0)

ymin = 0
i = 0
fig, axs = plt.subplots(9, 6, figsize=(14, 20))
fig.tight_layout()

for x0 in range(0, 9):
    for y0 in range(0, 6):
```

```
if i < 10:
    sensor_number = 'sensor_0{}'.format(i)
    ymax = df[sensor_number].max()
elif i > 51:
    break
else:
    sensor_number = 'sensor_{}'.format(i)
    ymax = df[sensor_number].max()

axs[x0, y0].plot(df[sensor_number])
axs[x0, y0].set_title(sensor_number)

broken_indices = df[df['machine_status'] == 'BROKEN'].index
broken_values = df[df['machine_status'] == 'BROKEN'][sensor_number]
axs[x0, y0].scatter(broken_indices, broken_values, c='red', marker='x', zorder=10)
i += 1
```

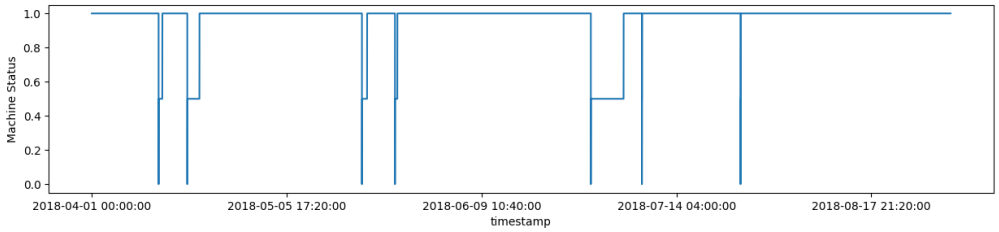


df.drop(['Unnamed: 0','sensor_00','sensor_15','sensor_50','sensor_51'], axis=1, inplace=True)
df.set_index('timestamp').plot(subplots =True, sharex = True, figsize = (20,50));



En esta parte se está cambiando el estado de la maquina a valores de 1, 0 y 0.5 para poder trabajar mejor.

```
# Estado de la máquina. 1 = operando (normal), 0.5 mantenimiento (recovering) and 0 = falla (broken)
df.set_index('timestamp').Operation.plot(figsize=(15,3));
plt.ylabel('Machine Status');
```



En machine learning no se suele trabajar con series de tiempo, en lo siguiente veremos cómo es el comportamiento de los algoritmos sin cambio de tiempo.

```
to_convert = ['sensor_01', 'sensor_02', 'sensor_03', 'sensor_04',
      'sensor_05', 'sensor_06', 'sensor_07', 'sensor_08', 'sensor_09',
      'sensor_10', 'sensor_11', 'sensor_12', 'sensor_13', 'sensor_14',
      'sensor_16', 'sensor_17', 'sensor_18', 'sensor_19', 'sensor_20',
      'sensor_21', 'sensor_22', 'sensor_23', 'sensor_24', 'sensor_25',
      'sensor_26', 'sensor_27', 'sensor_28', 'sensor_29', 'sensor_30',
      'sensor_31', 'sensor_32', 'sensor_33', 'sensor_34', 'sensor_35',
      'sensor_36', 'sensor_37', 'sensor_38', 'sensor_39', 'sensor_40',
      'sensor_41', 'sensor_42', 'sensor_43', 'sensor_44', 'sensor_45',
      'sensor_46', 'sensor_47', 'sensor_48', 'sensor_49']
```

El siguiente código se utiliza para propagar hacia adelante los últimos valores no nulos en el DataFrame

```
df = df.backfill()
```

Con el siguiente código veremos un resumen estadístico de los valores faltantes NaN del Dataframe.

```
df.isna().describe()
```

	timestamp	sensor_01	sensor_02	sensor_03	sensor_04	sensor_05	sensor_06	:
count	220320	220320	220320	220320	220320	220320	220320	
unique	1	1	1	1	1	1	1	
top	False	False	False	False	False	False	False	
freq	220320	220320	220320	220320	220320	220320	220320	

4 rows × 51 columns

Con el siguiente código establecemos como índice del Dataframe la columna timestamp, esto para hacer trabajos en series de tiempo.
Estableciendo también la frecuencia en minutos

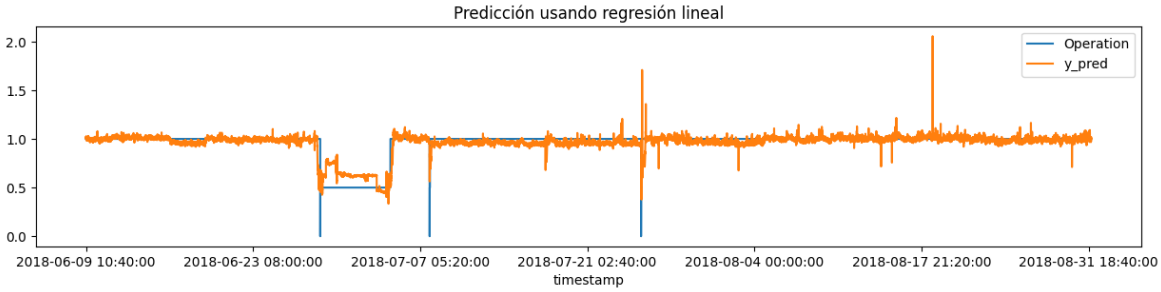
```
df.set_index('timestamp', inplace=True)
df.index.freq = 'min'
```

Dividimos los datos en datos de entrenamiento (train) y en datos de prueba (test).

```
train_df = df.loc[df.index < "2018-06-09 10:40:00"]
test_df = df.loc[df.index >= "2018-06-09 10:40:00"]
X_train = train_df.drop(['machine_status', 'Operation'], axis = 1)
y_train = train_df.Operation
X_test = test_df.drop(['machine_status', 'Operation'], axis = 1)
y_test = test_df.Operation
```

Aplicando la regresión lineal

```
reg = LinearRegression()
reg.fit(X_train, y_train)
y_pred = reg.predict(X_test)

y_test_plot = y_test.copy()
y_test_plot = pd.DataFrame(y_test_plot)
y_test_plot['y_pred'] = y_pred.tolist()
y_test_plot.plot.line(figsize=(15,3));
plt.title('Predicción usando regresión lineal');
```



Ahora usaremos la métrica de evaluación del modelo

```
import statsmodels.api as sm
from sklearn.metrics import mean_squared_error, mean_absolute_error
import numpy as np

# Crear el modelo de regresión lineal OLS
mod = sm.OLS(y_train, sm.add_constant(X_train))

res = mod.fit()
```

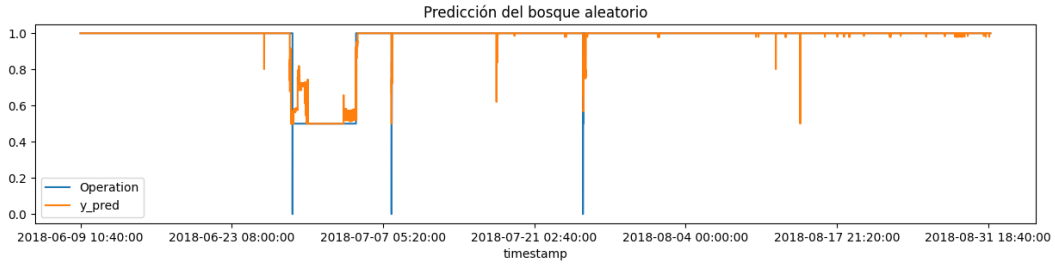
```
y_pred = res.predict(sm.add_constant(X_test))

rmse = np.sqrt(mean_squared_error(y_test, y_pred))
print('RMSE: %.3f' % rmse)

mae = mean_absolute_error(y_test, y_pred)
print('MAE: %.3f' % mae)
```

```
RMSE: 0.059
MAE: 0.036
```

Aplicando el Bosque aleatorio

```
rf = RandomForestRegressor(n_estimators=200, max_depth=10, min_samples_split=5)
rf.fit(X_train, y_train)
y_pred = rf.predict(X_test)
```

```
y_test_plot = pd.DataFrame(y_test_plot)
y_test_plot['y_pred'] = y_pred.tolist()
y_test_plot.plot.line(figsize=(15,3));
plt.title('Predicción del bosque aleatorio');
```



Usando las métricas de evaluación

```
from sklearn.metrics import mean_squared_error, mean_absolute_error

y_pred_rf = rf.predict(X_test)

rmse_rf = np.sqrt(mean_squared_error(y_test, y_pred_rf))
print('RMSE for Random Forest: %.3f' % rmse_rf)

mae_rf = mean_absolute_error(y_test, y_pred_rf)
print('MAE for Random Forest: %.3f' % mae_rf)
```

```
RMSE for Random Forest: 0.036
MAE for Random Forest: 0.005
```

Los algoritmos usados en este trabajo no son los más adecuados para ser aplicados a series temporales, los métodos más comunes usados para estos casos son las ARIMAS, vectores autorregresivos, etc. En el caso de deep learning se tiene a las Redes Recurrentes (RNN´s) pero no se usará estos algoritmos para este trabajo ya que no es el objetivo del mismo. Según lo visto anteriormente el error cuadrático medio es menor en el bosque aleatorio lo que lo convierte en la mejor opción para predecir estos datos. Para poder aplicar random forest a series temporales usamos la validación walk-forward en comparación con la validación que se usa por defecto que es k-fold.

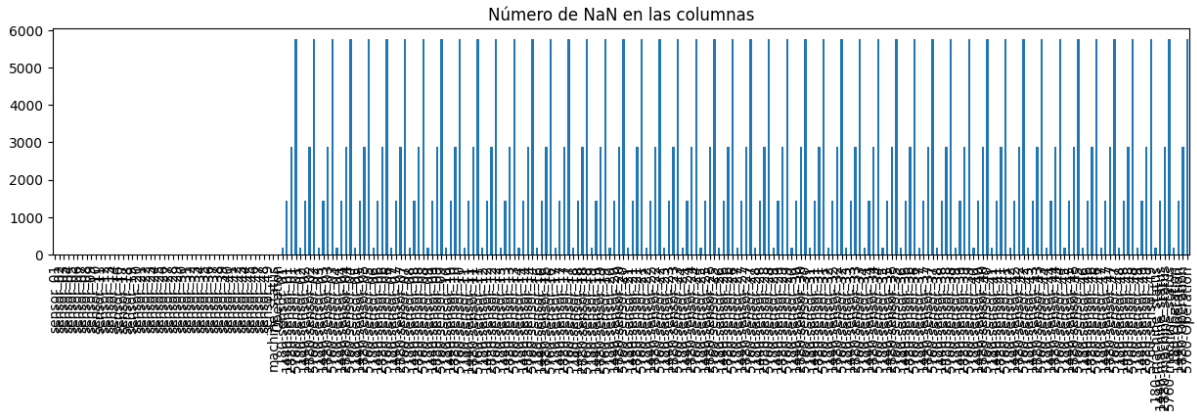
Con el siguiente código se crea columnas nuevas que vendrían a representar los valores pasados de los valores en diferentes intervalos de tiempo, la escala del tiempo se da en minutos por lo que 5760 minutos corresponde a 4 días.

```
for i in df.columns:
    if i == 'timestamp':
        continue
    else:
        for t in [180, 1440, 2880, 5760]:
            df[f'{t}-{i}'] = df[i].shift(t)
```

En el código anterior se hace la predicción para 180, 1440, 2880 y 5760 minutos.

La cantidad de Nan para cada columna nueva es:

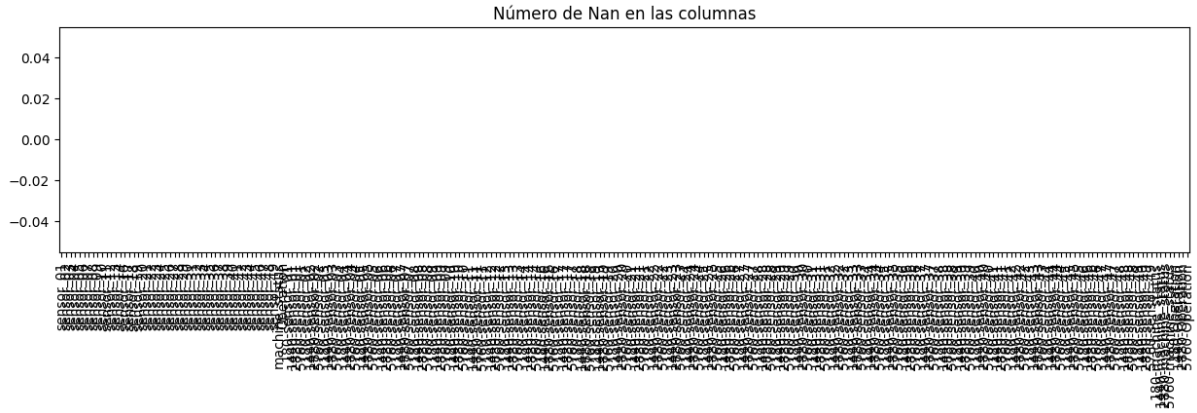
```
df.isna().sum().plot(kind='bar', figsize = (15, 3));
plt.title('Número de NaN en las columnas');
```



Se remueven las columnas que contienen Nan con el siguiente código

```
df = df.backfill()
df.dropna(inplace = True)
```

```
df.isna().sum().plot(kind='bar', figsize = (15, 3));
plt.title('Número de Nan en las columnas');
```



Con el codigo .index establecemos la periodicidad (frecuencia) de los datos en minutos.

En el contexto de análisis de series temporales, establecer la frecuencia es importante para indicar cómo están espaciados en el tiempo los puntos de datos en el índice. Esto es fundamental para realizar análisis y cálculos precisos en series temporales.

```
df.index.freq = 'min'
```

Ahora se crean columnas nuevas concatenadas con los datos originales en función de diferentes grupos de características basados en los intervalos de tiempo.

```
df_180 = pd.concat([df[df.columns[50::4]], df[df.columns[1:48:1]]], axis = 1)
df_1440 = pd.concat([df[df.columns[51::4]], df[df.columns[1:48:1]]], axis= 1)
df_2880 = pd.concat([df[df.columns[52::4]], df[df.columns[1:48:1]]], axis =1)
df_5760 = pd.concat([df[df.columns[53::4]], df[df.columns[1:48:1]]], axis = 1)
```

5760 minutos de predicción

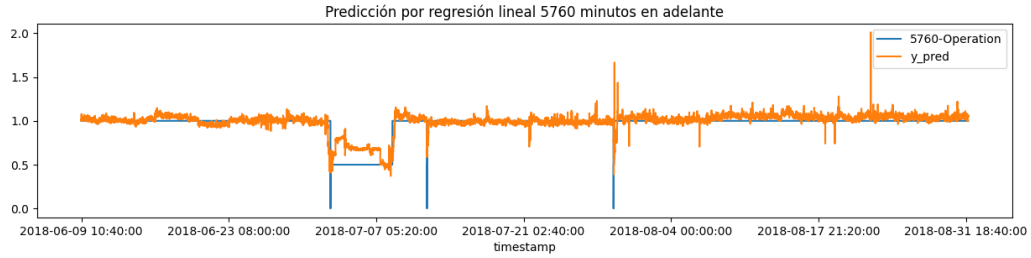
El siguiente código realiza la división de un conjunto de datos en intervalos de tiempo de 4 días (5760 minutos), generando conjuntos de entrenamiento y prueba. Alrededor del 50% de los datos se asignan para el entrenamiento y la otra mitad para la prueba. Luego, se preparan las características y etiquetas respectivas para ambos conjuntos, las columnas relacionadas con "machine_status" y "Operation" se eliminan para la entrada, formando así conjuntos listos para ser utilizados en el entrenamiento y evaluación de un modelo de aprendizaje automático.

```
number = 5760
df_number = df_5760

train_df = df_number.loc[df.index < "2018-06-09 10:40:00"]
test_df = df_number.loc[df.index >= "2018-06-09 10:40:00"]

# X_train = train_df.drop(['machine_status', 'Operation'], axis = 1)
X_train = train_df.drop([f'{number}-machine_status', f'{number}-Operation'], axis = 1)
y_train = train_df[f'{number}-Operation']

# X_test = test_df.drop(['machine_status', 'Operation'], axis = 1)
X_test = test_df.drop([f'{number}-machine_status', f'{number}-Operation'], axis = 1)
```

```
X_test_plot = test_df.drop([f'{number}-machine_status'], axis = 1)
y_test = test_df[f'{number}-Operation']
```

Volvemos a aplicar la regresión lineal múltiple con los datos predichos para 4 días.

```
# Regresión lineal para 4 dias
reg = LinearRegression()
reg.fit(X_train, y_train)
y_pred = reg.predict(X_test)

y_test_plot = y_test.copy()
y_test_plot = pd.DataFrame(y_test_plot)
y_test_plot['y_pred'] = y_pred.tolist()
y_test_plot.plot.line(figsize=(15,3));
plt.title(f'Predicción por regresión lineal {number} minutos en adelante');
```



Aplicando las métricas de evaluación.

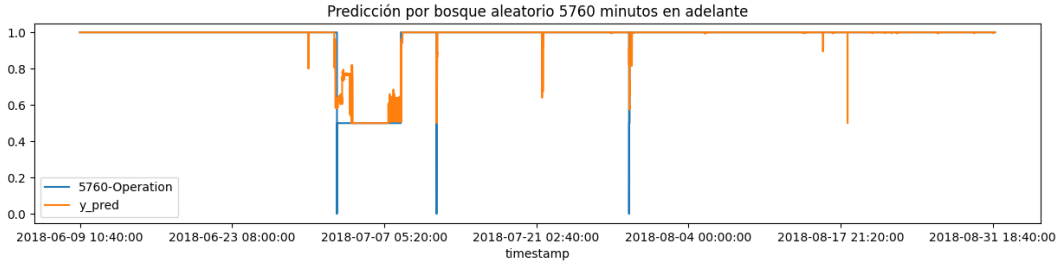
```
# Crear el modelo de regresión lineal OLS
mod = sm.OLS(y_train, sm.add_constant(X_train))
res = mod.fit()
y_pred = res.predict(sm.add_constant(X_test))
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
print('RMSE: %.3f' % rmse)
mae = mean_absolute_error(y_test, y_pred)
print('MAE: %.3f' % mae)
```

RMSE: 0.070
MAE: 0.044

Aplicando random forest para los datos predichos en 4 días

```
#random forest para 4 dias
rf = RandomForestRegressor(n_estimators=200, max_depth=10, min_samples_split=5)
rf.fit(X_train, y_train)
y_pred = rf.predict(X_test)

rf = RandomForestRegressor()
rf.fit(X_train, y_train)
y_pred = rf.predict(X_test)
```

```
y_test_plot = y_test.copy()
y_test_plot = pd.DataFrame(y_test_plot)
y_test_plot['y_pred'] = y_pred.tolist()
y_test_plot.plot.line(figsize=(15,3));
plt.title(f'Predicción por bosque aleatorio {number} minutos en adelante');
```

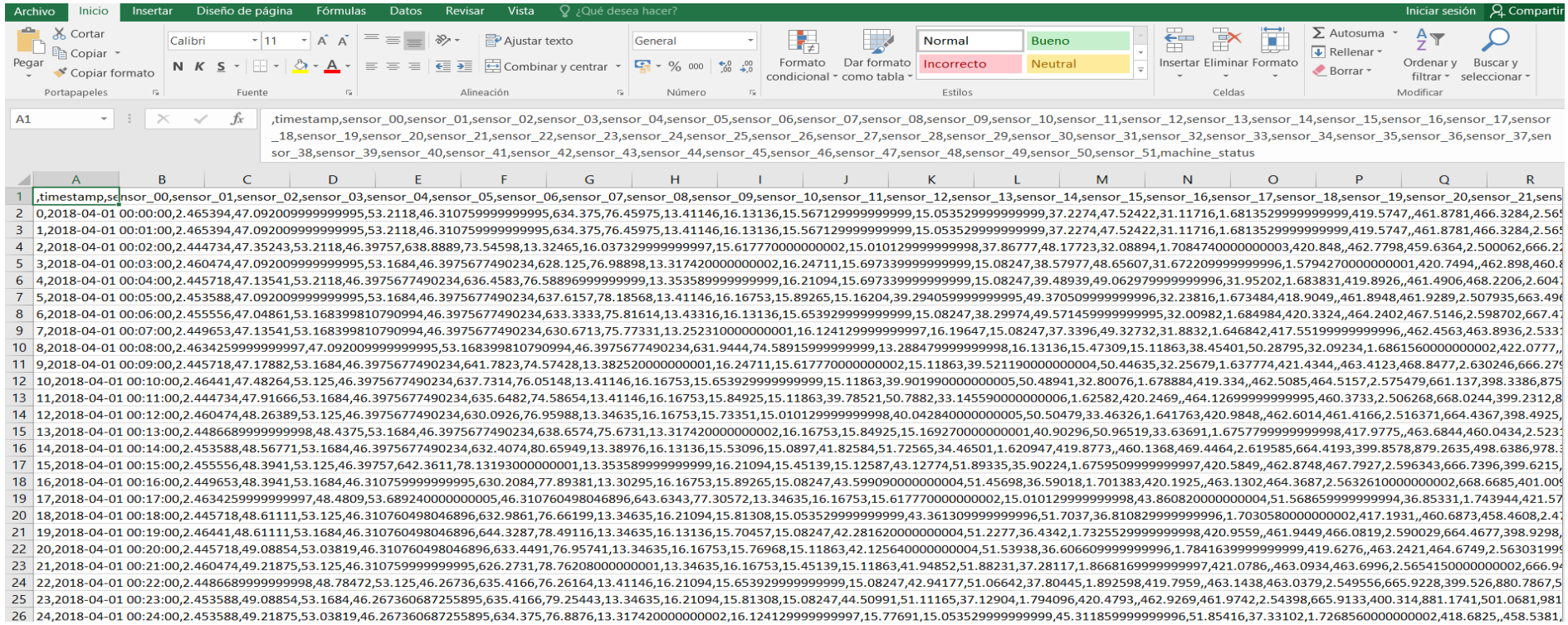


Aplicando métricas de evaluación

```
y_pred_rf = rf.predict(X_test)

rmse_rf = np.sqrt(mean_squared_error(y_test, y_pred_rf))
print('RMSE for Random Forest: %.3f' % rmse_rf)

mae_rf = mean_absolute_error(y_test, y_pred_rf)
print('MAE for Random Forest: %.3f' % mae_rf)
```

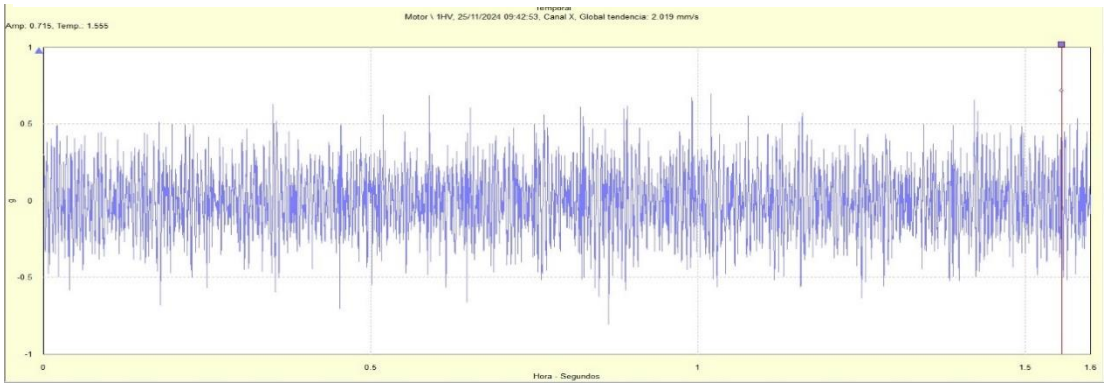
RMSE for Random Forest: 0.037
MAE for Random Forest: 0.006

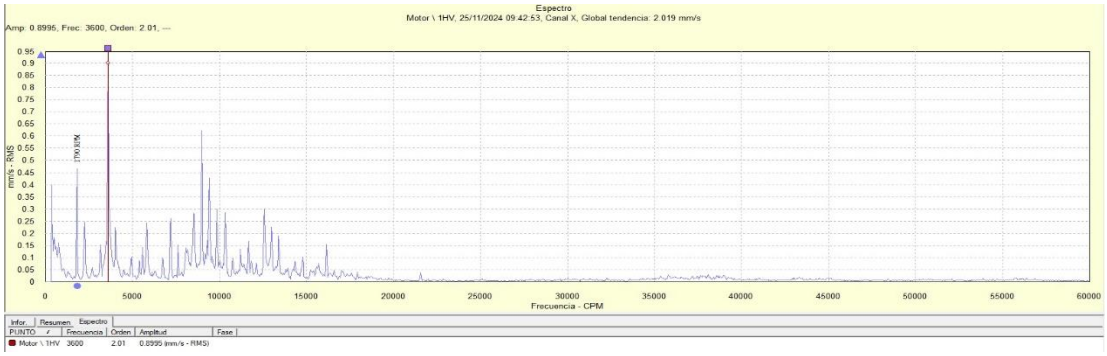
ANEXO 2: Base de datos

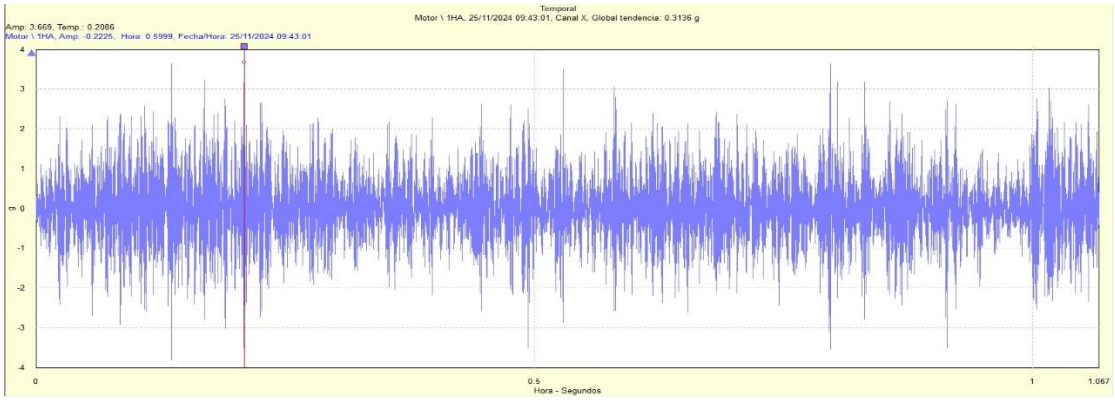


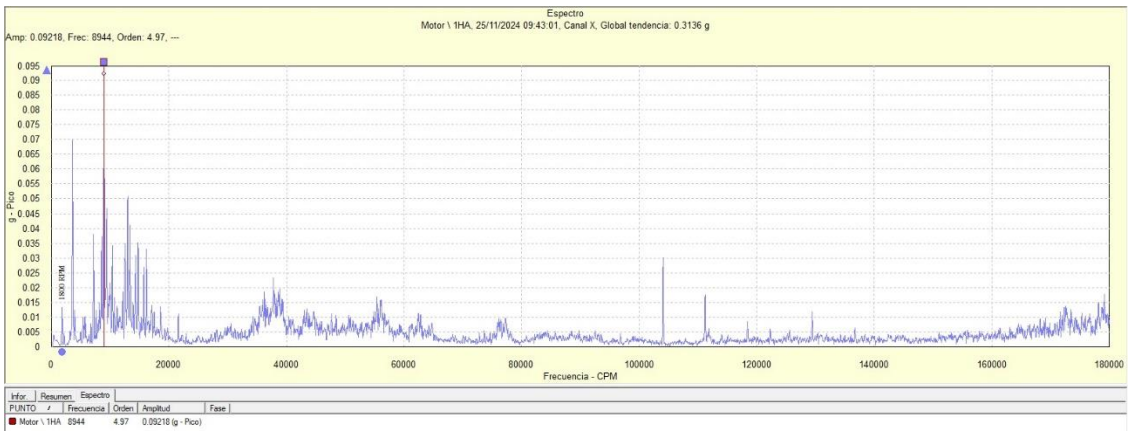
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	,timestamp,sensor_00,sensor_01,sensor_02,sensor_03,sensor_04,sensor_05,sensor_06,sensor_07,sensor_08,sensor_09,sensor_10,sensor_11,sensor_12,sensor_13,sensor_14,sensor_15,sensor_16,sensor_17,sensor_18,sensor_19,sensor_20,sensor_21,sens																	
2	0,2018-04-01 00:00:00,2.465394,47.092009999999995,53.2118,46.310759999999995,634.375,76.45975,13.41146,16.13136,15.567129999999999,15.053529999999999,37.2274,47.52422,31.11716,1.6813529999999999,419.5747,,461.8781,466.3284,2.565																	
3	1,2018-04-01 00:01:00,2.465394,47.092009999999995,53.2118,46.310759999999995,634.375,76.45975,13.41146,16.13136,15.567129999999999,15.053529999999999,37.2274,47.52422,31.11716,1.6813529999999999,419.5747,,461.8781,466.3284,2.565																	
4	2,2018-04-01 00:02:00,2.444734,47.35243,53.2118,46.39757,638.8889,73.54598,13.32465,16.037329999999997,15.617770000000002,15.010129999999998,37.86777,48.17723,32.08894,1.7084740000000003,420.848,,462.7798,459.6364,2.500062,666.22																	
5	3,2018-04-01 00:03:00,2.460474,47.092009999999995,53.1684,46.3975677490234,628.125,76.98898,13.317420000000002,16.24711,15.697339999999999,15.08247,38.57977,48.65607,31.672209999999996,1.5794270000000001,420.7494,,462.898,460.8																	
6	4,2018-04-01 00:04:00,2.445718,47.13541,53.2118,46.3975677490234,636.4583,76.58896999999999,13.353589999999999,16.21094,15.697339999999999,15.08247,39.48939,49.062979999999996,31.95202,1.683831,419.8926,,461.4906,468.2206,2.6047																	
7	5,2018-04-01 00:05:00,2.453588,47.092009999999995,53.1684,46.3975677490234,637.6157,78.18568,13.41146,16.16753,15.89265,15.16204,39.294059999999995,49.370509999999996,32.23816,1.673484,418.9049,,461.8948,461.9289,2.507935,663.496																	
8	6,2018-04-01 00:06:00,2.455556,47.04861,53.168399810790994,46.3975677490234,633.3333,75.81614,13.43316,16.13136,15.653929999999999,15.08247,38.29974,49.571459999999995,32.00982,1.684984,420.3324,,464.2402,467.5146,2.598702,667.47																	
9	7,2018-04-01 00:07:00,2.449653,47.13541,53.168399810790994,46.3975677490234,630.6713,75.77331,13.252310000000001,16.124129999999997,16.19647,15.08247,37.3396,49.32732,31.8832,1.646842,417.55199999999996,,462.4563,463.8936,2.5331																	
10	8,2018-04-01 00:08:00,2.4634259999999997,47.092009999999995,53.168399810790994,46.3975677490234,631.9444,74.58915999999999,13.288479999999998,16.13136,15.47309,15.11863,38.45401,50.28795,32.09234,1.6861560000000002,422.0777,,																	
11	9,2018-04-01 00:09:00,2.445718,47.17882,53.1684,46.3975677490234,641.7823,74.57428,13.382520000000001,16.24711,15.617770000000002,15.11863,39.521190000000004,50.44635,32.25679,1.637774,421.4344,,463.4123,468.8477,2.630246,666.279																	
12	10,2018-04-01 00:10:00,2.46441,47.48264,53.125,46.3975677490234,637.7314,76.05148,13.41146,16.16753,15.653929999999999,15.11863,39.901990000000005,50.48941,32.80076,1.678884,419.334,,462.5085,464.5157,2.575479,661.137,398.3386,875																	
13	11,2018-04-01 00:11:00,2.444734,47.91666,53.1684,46.3975677490234,635.6482,74.58654,13.41146,16.16753,15.84925,15.11863,39.78521,50.7882,33.145590000000006,1.62582,420.2469,,464.12699999999995,460.3733,2.506268,668.0244,399.2312,8																	
14	12,2018-04-01 00:12:00,2.460474,48.26389,53.125,46.3975677490234,630.0926,76.95988,13.34635,16.16753,15.73351,15.010129999999998,40.042840000000005,50.50479,33.46326,1.641763,420.9848,,462.6014,461.4166,2.516371,664.4367,398.4925,																	
15	13,2018-04-01 00:13:00,2.4486689999999998,48.4375,53.1684,46.3975677490234,638.6574,75.6731,13.317420000000002,16.16753,15.84925,15.169270000000001,40.90296,50.96519,33.63691,1.6757799999999998,417.9775,,463.6844,460.0434,2.523																	
16	14,2018-04-01 00:14:00,2.453588,48.56771,53.1684,46.3975677490234,632.4074,80.65949,13.38976,16.13136,15.53096,15.0897,41.82584,51.72565,34.46501,1.620947,419.8773,,460.1368,469.4464,2.619585,664.4193,399.8578,879.2635,498.6386,978.3																	
17	15,2018-04-01 00:15:00,2.455556,48.3941,53.125,46.39757,642.3611,78.13193000000001,13.353589999999999,16.21094,15.45139,15.12587,43.12774,51.89335,35.90224,1.6759509999999997,420.5849,,462.8748,467.7927,2.596343,666.7396,399.6215,																	
18	16,2018-04-01 00:16:00,2.449653,48.3941,53.1684,46.310759999999995,630.2084,77.89381,13.30295,16.16753,15.89265,15.08247,43.599090000000004,51.45698,36.59018,1.701383,420.1925,,463.1302,464.3687,2.5632610000000002,668.6685,401.009																	
19	17,2018-04-01 00:17:00,2.4634259999999997,48.4809,53.689240000000005,46.310760498046896,643.6343,77.30572,13.34635,16.16753,15.617770000000002,15.010129999999998,43.860820000000004,51.568659999999994,36.85331,1.743944,421.57																	
20	18,2018-04-01 00:18:00,2.445718,48.61111,53.125,46.310760498046896,632.9861,76.66199,13.34635,16.21094,15.81308,15.053529999999999,43.361309999999996,51.7037,36.810829999999996,1.7030580000000002,417.1931,,460.6873,458.4608,2.47																	
21	19,2018-04-01 00:19:00,2.46441,48.61111,53.1684,46.310760498046896,644.3287,78.49116,13.34635,16.13136,15.70457,15.08247,42.281620000000004,51.2277,36.4342,1.7325529999999998,420.9559,,461.9449,466.0819,2.590029,664.4677,398.9298,																	
22	20,2018-04-01 00:20:00,2.445718,49.08854,53.03819,46.310760498046896,633.4491,76.95741,13.34635,16.16753,15.76968,15.11863,42.125640000000004,51.53938,36.606609999999996,1.7841639999999999,419.6276,,463.2421,464.6749,2.563031999																	
23	21,2018-04-01 00:21:00,2.460474,49.21875,53.125,46.310759999999995,626.2731,78.76208000000001,13.34635,16.16753,15.45139,15.11863,41.94852,51.88231,37.28117,1.8668169999999997,421.0786,,463.0934,463.6996,2.5654150000000002,666.94																	
24	22,2018-04-01 00:22:00,2.4486689999999998,48.78472,53.125,46.26736,635.4166,76.26164,13.41146,16.21094,15.653929999999999,15.08247,42.94177,51.06642,37.80445,1.892598,419.7959,,463.1438,463.0379,2.549556,665.9228,399.526,880.7867,5																	
25	23,2018-04-01 00:23:00,2.453588,49.08854,53.1684,46.267360687255895,635.4166,79.25443,13.34635,16.21094,15.81308,15.08247,44.50991,51.11165,37.12904,1.794096,420.4793,,462.9269,461.9742,2.54398,665.9133,400.314,881.1741,501.0681,981																	
26	24,2018-04-01 00:24:00,2.453588,49.21875,53.03819,46.267360687255895,634.375,76.8876,13.317420000000002,16.124129999999997,15.77691,15.053529999999999,45.311859999999996,51.85416,37.33102,1.7268560000000002,418.6825,,458.5381,																	

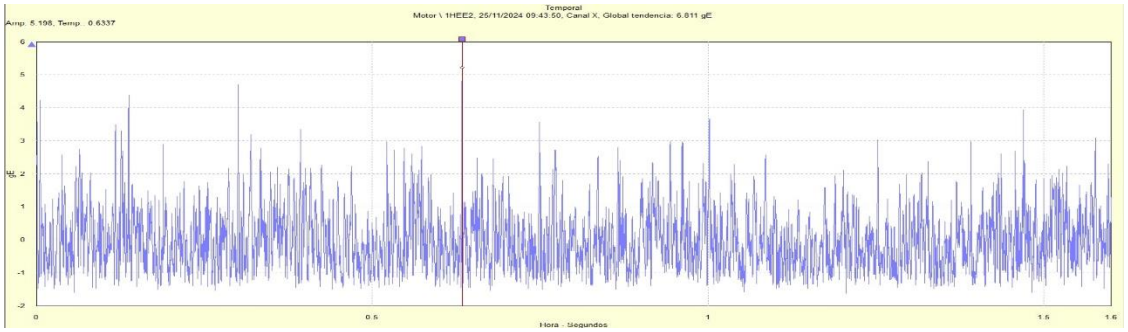
ANEXO 3: Medidas de vibración hechas por un acelerómetro.

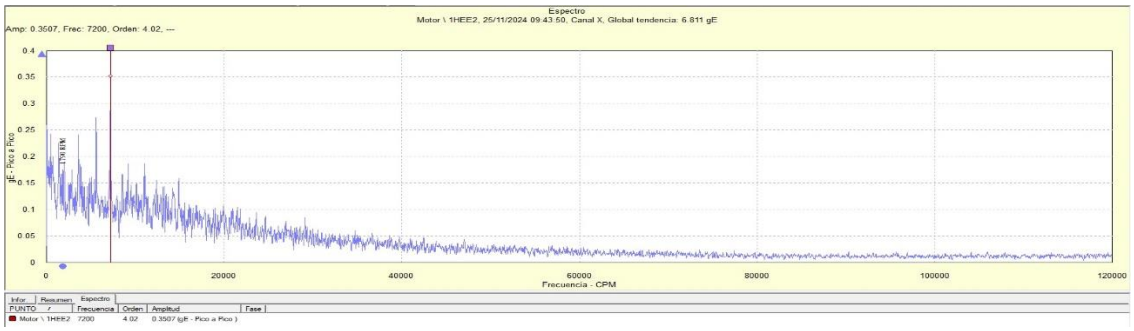


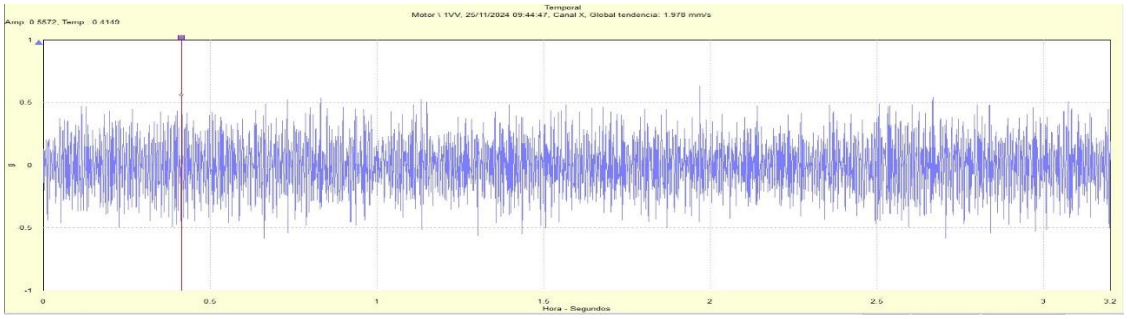


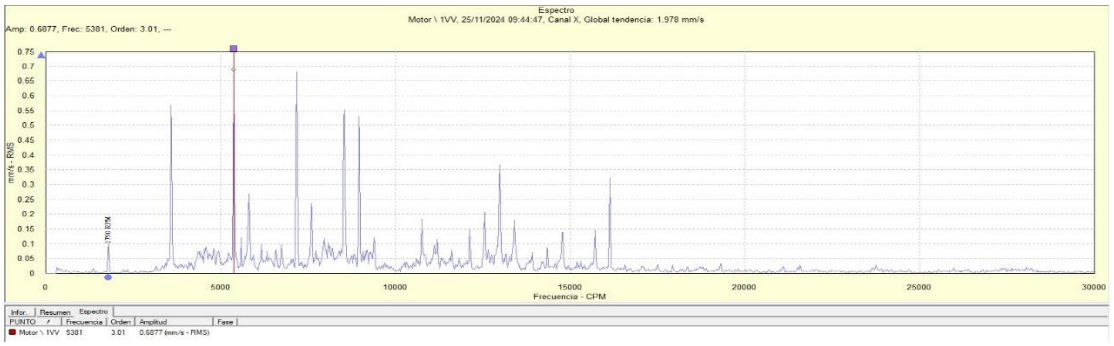


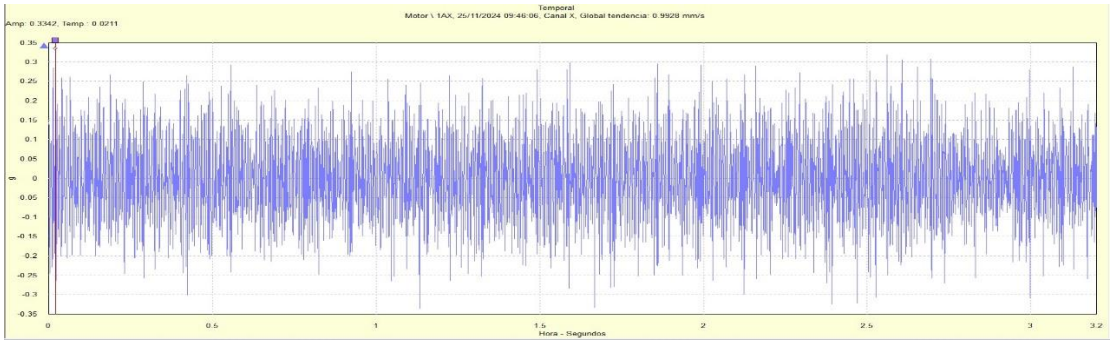


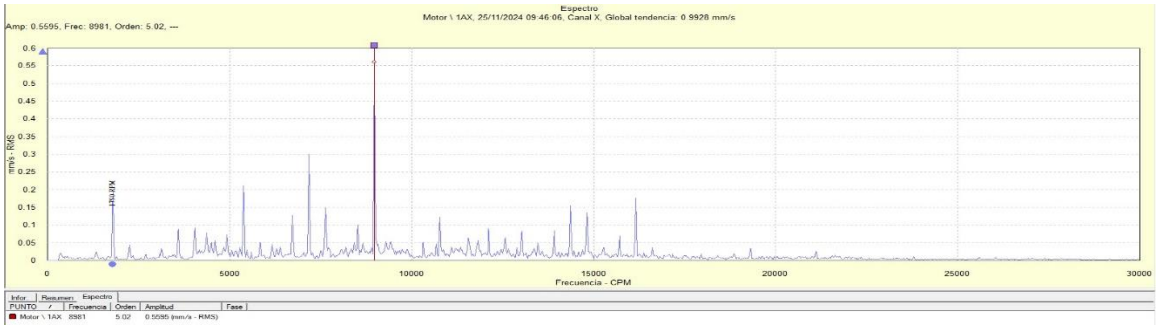


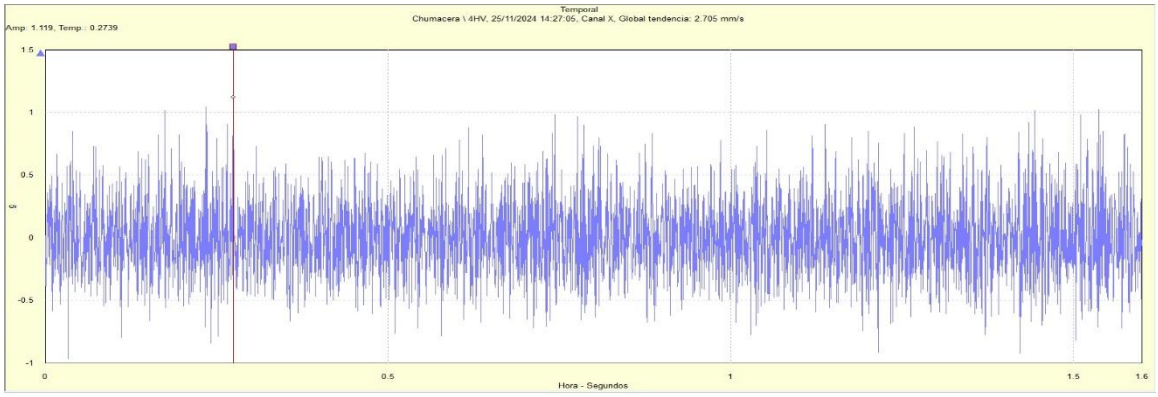


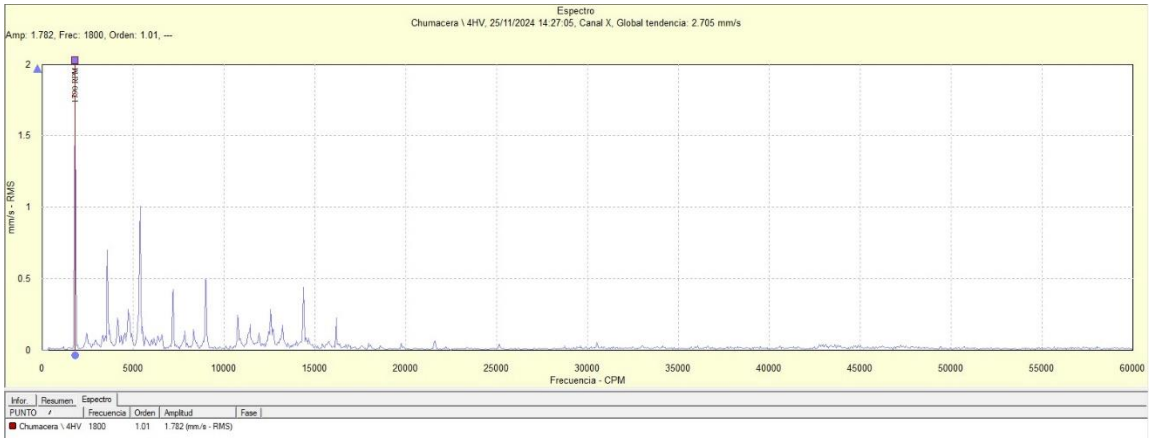


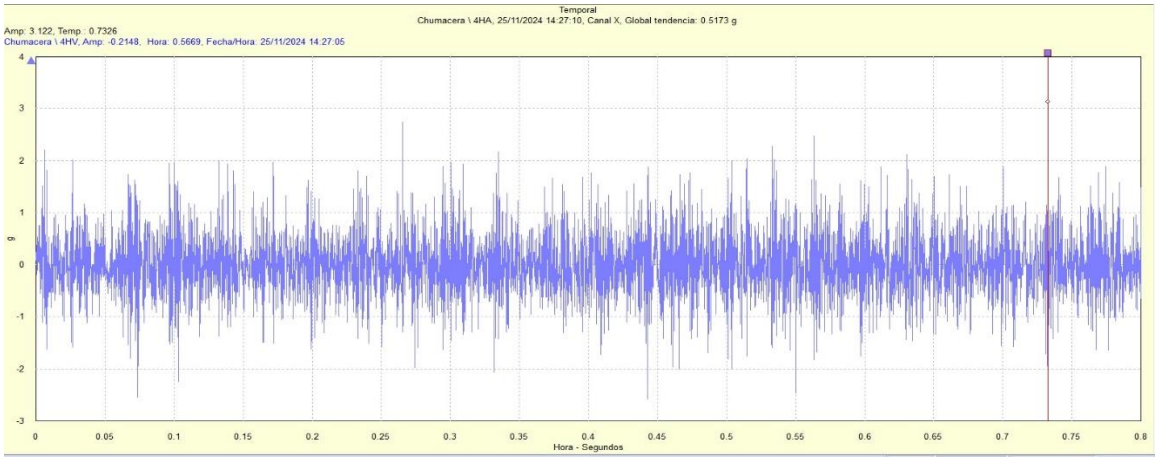


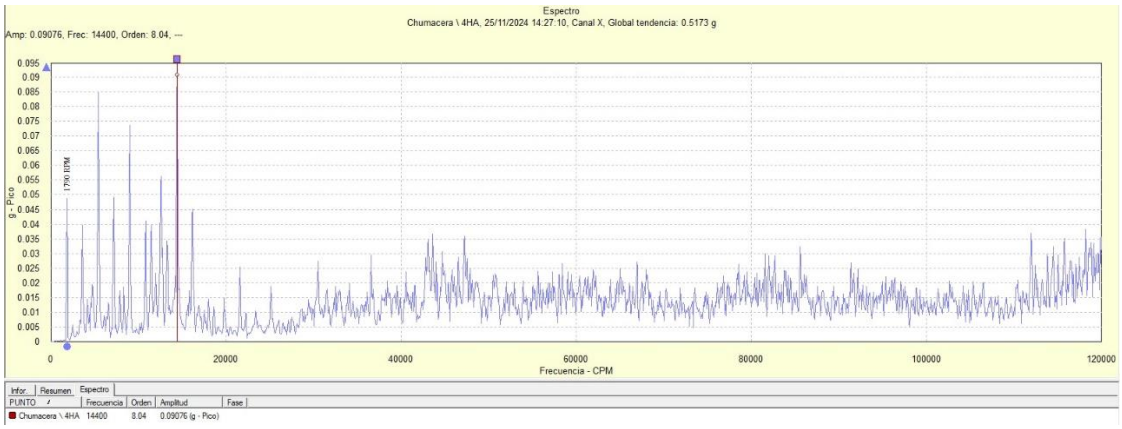


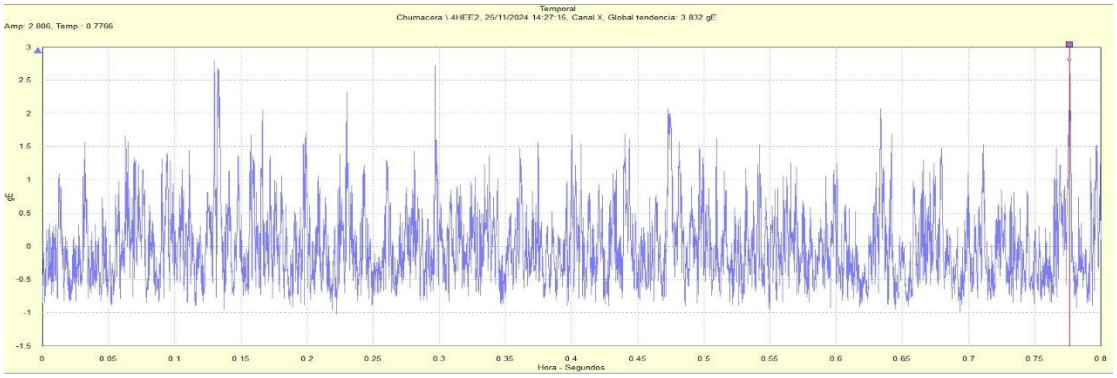


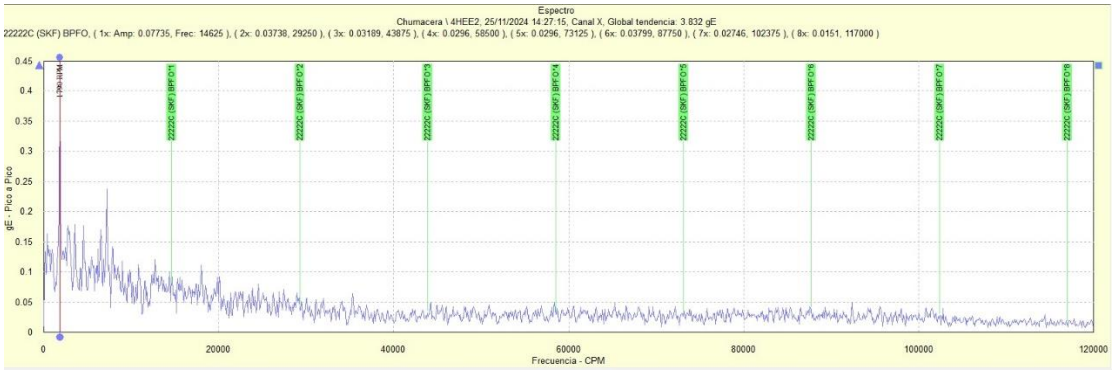


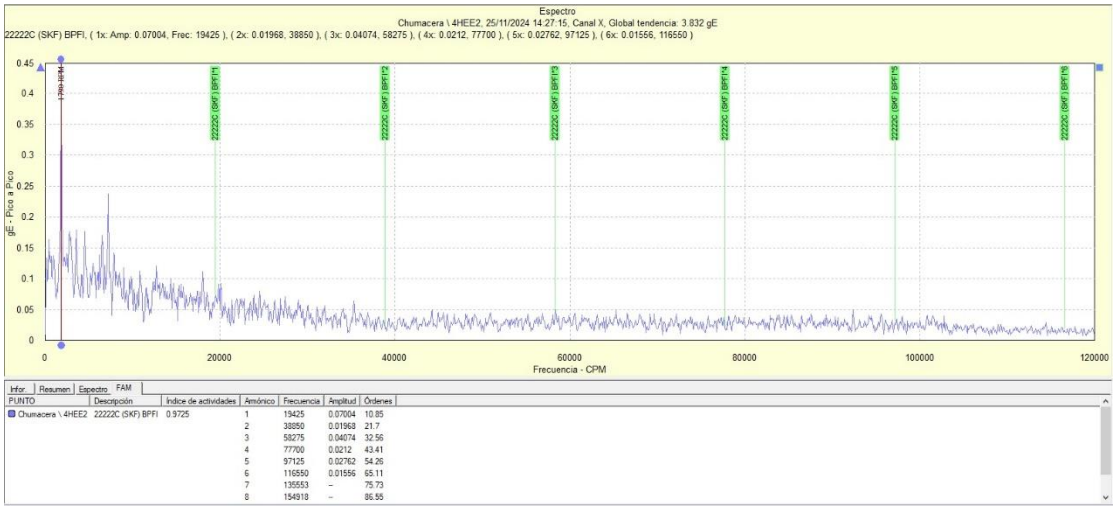












ANEXO 4: Autorización para el depósito de tesis o trabajo de investigación en el repositorio institucional

Universidad Nacional del Altiplano Puno | Vicerrectorado de Investigación | Repositorio Institucional

AUTORIZACIÓN PARA EL DEPÓSITO DE TESIS O TRABAJO DE INVESTIGACIÓN EN EL REPOSITORIO INSTITUCIONAL

Por el presente documento, Yo Aldair Rodrigo Bemavembe,
identificado con DNI 73939235 en mi condición de egresado de:

* ☒ **Escuela Profesional,** ☐ **Programa de Segunda Especialidad,** ☐ **Programa de Maestría o Doctorado**

Ingeniería Mecánica Eléctrica,
informo que he elaborado el/la ☒ **Tesis** o ☐ **Trabajo de Investigación** denominada:

"MANTENIMIENTO PREDICTIVO USANDO ALGORITMOS DE MACHINE LEARNING APLICADO A BOMBAS DE AGUA"

para la obtención de ☐ **Grado,** ☒ **Título Profesional** o ☐ **Segunda Especialidad.**

Por medio del presente documento, afirmo y garantizo ser el legítimo, único y exclusivo titular de todos los derechos de propiedad intelectual sobre los documentos arriba mencionados, las obras, los contenidos, los productos y/o las creaciones en general (en adelante, los "Contenidos") que serán incluidos en el repositorio institucional de la Universidad Nacional del Altiplano de Puno.

También, doy seguridad de que los contenidos entregados se encuentran libres de toda contraseña, restricción o medida tecnológica de protección, con la finalidad de permitir que se puedan leer, descargar, reproducir, distribuir, imprimir, buscar y enlazar los textos completos, sin limitación alguna.

Autorizo a la Universidad Nacional del Altiplano de Puno a publicar los Contenidos en el Repositorio Institucional y, en consecuencia, en el Repositorio Nacional Digital de Ciencia, Tecnología e Innovación de Acceso Abierto, sobre la base de lo establecido en la Ley N° 30035, sus normas reglamentarias, modificatorias, sustitutorias y conexas, y de acuerdo con las políticas de acceso abierto que la Universidad aplique en relación con sus Repositorios Institucionales. Autorizo expresamente toda consulta y uso de los Contenidos, por parte de cualquier persona, por el tiempo de duración de los derechos patrimoniales de autor y derechos conexos, a título gratuito y a nivel mundial.

En consecuencia, la Universidad tendrá la posibilidad de divulgar y difundir los Contenidos, de manera total o parcial, sin limitación alguna y sin derecho a pago de contraprestación, remuneración ni regalía alguna a favor mío; en los medios, canales y plataformas que la Universidad y/o el Estado de la República del Perú determinen, a nivel mundial, sin restricción geográfica alguna y de manera indefinida, pudiendo crear y/o extraer los metadatos sobre los Contenidos, e incluir los Contenidos en los índices y buscadores que estimen necesarios para promover su difusión.

Autorizo que los Contenidos sean puestos a disposición del público a través de la siguiente licencia:

Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional. Para ver una copia de esta licencia, visita: https://creativecommons.org/licenses/by-nc-sa/4.0/

En señal de conformidad, suscribo el presente documento.

Puno 12 de Diciembre del 2024

FIRMA (obligatoria) Huella

ANEXO 5: Declaración jurada de autenticidad de tesis

Universidad Nacional del Altiplano Puno | Vicerrectorado de Investigación | Repositorio Institucional

DECLARACIÓN JURADA DE AUTENTICIDAD DE TESIS

Por el presente documento, Yo Aldair Rodrigo Benavente,
identificado con DNI 73939935 en mi condición de egresado de:

☒ **Escuela Profesional,** ☐ **Programa de Segunda Especialidad,** ☐ **Programa de Maestría o Doctorado**
Ingeniería Mecánica Electrica,
informo que he elaborado el/la ☒ **Tesis** o ☐ **Trabajo de Investigación** denominada:
"MANTENIMIENTO PREDICTIVO USANDO ALGORITMOS DE MACHINE LEARNING APLICADO A BOMBAS DE AGUA"

Es un tema original.

Declaro que el presente trabajo de tesis es elaborado por mi persona y **no existe plagio/copia** de ninguna naturaleza, en especial de otro documento de investigación (tesis, revista, texto, congreso, o similar) presentado por persona natural o jurídica alguna ante instituciones académicas, profesionales, de investigación o similares, en el país o en el extranjero.

Dejo constancia que las citas de otros autores han sido debidamente identificadas en el trabajo de investigación, por lo que no asumiré como suyas las opiniones vertidas por terceros, ya sea de fuentes encontradas en medios escritos, digitales o Internet.

Asimismo, ratifico que soy plenamente consciente de todo el contenido de la tesis y asumo la responsabilidad de cualquier error u omisión en el documento, así como de las connotaciones éticas y legales involucradas.

En caso de incumplimiento de esta declaración, me someto a las disposiciones legales vigentes y a las sanciones correspondientes de igual forma me someto a las sanciones establecidas en las Directivas y otras normas internas, así como las que me alcancen del Código Civil y Normas Legales conexas por el incumplimiento del presente compromiso

Puno 12 de Diciembre del 2024

FIRMA (obligatoria)

Huella