



Facultad de Ingeniería
Carrera de Ingeniería Mecatrónica

Tesis:

**“DISEÑO DE UN MODELO PARA
MANTENIMIENTO PREDICTIVO EN
MOTORES DE INDUCCIÓN UTILIZANDO
TÉCNICAS DE LA INDUSTRIA 4.0”**

Autor: Jose Luis Contreras Alvarez

**Para optar el Título Profesional de
Ingeniero Mecatrónico**

Asesor: Ing. Robert Isaías Quispe Romero

Lima – Perú

2020

DEDICATORIA

Al Señor mi Dios, por haberme enseñado
todo lo que se hasta ahora y lo que me
seguirá enseñando.

A mis hermanas por alentarme siempre
A mis familiares por haberme apoyado en la
medida de sus posibilidades.

AGRADECIMIENTO

Agradezco desde lo más profundo de mi corazón a mi Señor mi Dios porque que me ayudo, me animo, me alentó y me fortaleció en los momentos más difíciles de mi vida y sin Él no hubiese sido posible estar donde estoy.

Agradezco a mis hermanas que fueron el instrumento que el Señor utilizo para alentarme y animarme cuando las cosas se ponían difíciles

Agradezco a la universidad y a los profesores y a mi asesor por compartir sus experiencias y conocimiento con el único fin de formarme académica y profesionalmente.

Agradezco a mis compañeros y amigos que tuve durante mi preparación, sí que fueron muchos, la incondicional ayuda que recibí de ellos, y que de alguna forma contribuyeron en la elaboración de esta Tesis de fin de grado.

RESUMEN

La industria en el mundo está pasando por un constante cambio tecnológico. El Perú no es ajeno a tales cambios, actualmente se está fomentando lentamente, pero a pasos firmes el desarrollo de nuevas tecnologías, tales como la Inteligencia artificial, el trabajo colaborativo, el Big data, el internet de las cosas (IoT), que nos están introduciendo a la cuarta revolución industrial o industria 4.0.

La industria en general tiene como uno de sus activos principales al motor de inducción, y en ese sentido es necesario implementar un método de mantenimiento predictivo basado en la inteligencia artificial, el cual monitorizara el estado del motor, con el fin de predecir el momento adecuado de cambio de sus elementos, y así evitar las paradas inesperadas que generan altos costos, haciendo a la industria local más competitiva.

Es así que, el presente trabajo utilizara el aprendizaje supervisado de Machine Learning para predecir el estado de los rodamientos del motor de inducción. Se trabajará con datos experimentales de la base de datos de acceso público de la Case Western Reserve University (Case Western & University Reserve (CWRU), n.d.), los cuales obtuvieron mediante diversos ensayos controlados en laboratorio.

PALABRAS CLAVE

Mantenimiento predictivo, Industria 4.0, Modelo, Sensor, Machine Learning, Big Data

ABSTRACT

The industry in the world is going through constant technological change. Perú is no stranger to such changes, currently the development of new technologies is being promoted slowly, but steadily, such as artificial intelligence, collaborative work, Big data, the internet of things (IoT), which we They are ushering in the fourth industrial revolution or Industry 4.0.

The industry in general has the induction motor as one of its main assets, and in that sense, it is necessary to implement a predictive maintenance method based on artificial intelligence, which will monitor the condition of the motor, in order to predict the right moment of operation. change of its elements, and thus avoid unexpected stops that generate high costs, making the local industry more competitive.

Thus, the present work will use supervised Machine Learning to predict the state of the induction motor bearings. We will work with experimental data from the Case Western Reserve University public access database (Case Western & University Reserve (CWRU), n.d.), which were obtained through various controlled laboratory tests.

KEYWORDS

Predictive maintenance, Industry 4.0, Model, Sensor, Machine Learning, Big Data

ÍNDICE GENERAL

DEDICATORIA.....	i
AGRADECIMIENTO.....	ii
RESUMEN.....	iii
ABSTRACT	iv
ÍNDICE GENERAL	v
INTRODUCCIÓN.....	ix
CAPITULO 1.....	1
ASPECTOS GENERALES.....	1
1.1. Definición del problema.....	1
1.1.1. Descripción del problema.....	1
1.1.2. Formulación del problema.....	2
1.2. Definición de objetivos	2
1.2.1. Objetivo general	2
1.2.2. Objetivos específicos.....	3
1.3. Alcances y Limitaciones	3
1.3.1. Alcance	3
1.3.2. Limitación	3
1.4. Justificación	4
1.5. Estado del arte	4
1.5.1. Soluciones Tecnológicas.....	7
CAPITULO 2.....	12
MARCO TEÓRICO.....	12
2.1. Teoría del funcionamiento del motor de inducción	12
2.2. Tipos de motores de inducción	14
2.3. Fallos más comunes en los motores de inducción	16
2.3.1. Fallas en el aislamiento en las bobinas del estator.....	17
2.3.2. Fallas en los rodamientos	18
2.3.3. Fallas por causa de excentricidad.....	21
2.4. Técnicas de mantenimiento	22
2.4.1. Mantenimiento Correctivo	22
2.4.2. Mantenimiento Preventivo.....	23
2.4.3. Mantenimiento Predictivo	23
2.5. Machine Learning	29
2.6. Aprendizaje Supervisado	30
2.7. Aprendizaje No supervisado	32
2.8. Técnicas utilizadas en Machine Learning	32
2.8.1. K vecinos más cercanos (KNN)	32

2.8.2.	Árbol de decisión	34
2.8.3.	Random Forest.....	37
2.8.4.	Máquina de vector de soporte (SVM).....	39
2.8.5.	Regularización.....	40
2.9.	Métricas de evaluación de los algoritmos de clasificación.....	43
2.9.1.	Matriz de Confusión.....	43
2.9.2.	Curva ROC y Área bajo la Curva (AUC)	44
2.9.3.	F1 Score	45
CAPÍTULO 3.....		47
DESARROLLO DE LA SOLUCIÓN.....		47
3.1.	Introducción a la solución.....	47
3.2.	Adquisición de Datos.....	48
3.2.1.	Banco de Ensayos.....	49
3.3.	Preparación de los datos.....	52
3.4.	Selección del lenguaje de programación	53
3.5.	Método de entrenamiento	55
3.6.	Entrenamiento y Simulación.....	56
3.7.	Entrenando el modelo para el caso del rodamiento extremo del eje (_DE)	56
3.7.1.	Entrenamiento con el modelo KNN	56
3.7.1.1.	Entrenamiento con el modelo Support Vector Machine (SVM).....	59
3.7.2.	Entrenamiento con el modelo Random Forest.....	63
CAPITULO 4.....		67
ANÁLISIS DE RESULTADOS.....		67
4.1.	Resultados.....	67
4.1.1.	Comparativo de eficiencia por modelo predictivo	68
4.1.2.	Comparativo de eficiencia de fallo por categoría	69
CONCLUSIONES.....		71
RECOMENDACIONES		73
BIBLIOGRAFÍA.....		74
ANEXOS.....		79
GLOSARIO.....		92

ÍNDICE DE FIGURAS

<i>Figura 1.</i> Proceso de adquisición de datos	7
<i>Figura 2.</i> Maquina Llenadora Cronos.....	9
<i>Figura 3.</i> Captura de imagen de placa de rayos x	10
<i>Figura 4.</i> Personal utilizando el aplicativo “eRx”	11
<i>Figura 5.</i> Motor de inducción de jaula de ardilla	14
<i>Figura 6.</i> Motor de inducción monofásico	15
<i>Figura 7.</i> Rotor de doble jaula de ardilla	16
<i>Figura 8.</i> Fallas comunes en el motor de inducción.....	17
<i>Figura 9.</i> Factores más comunes que deterioran el aislamiento del motor	18
<i>Figura 10.</i> Componentes del rodamiento.....	20
<i>Figura 11.</i> Falla de rodamientos.....	21
<i>Figura 12.</i> Tipos de Excentricidad.....	22
<i>Figura 13.</i> Cámara Termográfica	24
<i>Figura 14.</i> Medidor de Ultrasonido	26
<i>Figura 15.</i> Analizador de vibraciones	29
<i>Figura 16.</i> Sub campo de la inteligencia artificial.....	30
<i>Figura 17.</i> Captura de patrones	31
<i>Figura 18.</i> Reconocimiento de patrones (KNN)	34
<i>Figura 19.</i> Partes de un árbol de decisión	37
<i>Figura 21.</i> Arboles Aleatorios	38
<i>Figura 22.</i> Hiperplanos de separación (SVM)	39
<i>Figura 23.</i> Sobreajuste, bajo ajuste, ajuste correcto.....	41
<i>Figura 24.</i> Tipos de curva ROC.....	45
<i>Figura 25.</i> Diagrama de bloques del sistema de falla	48
<i>Figura 26.</i> Ordenador Portátil.....	50
<i>Figura 27.</i> Banco de ensayos.....	50
<i>Figura 28.</i> Señal tipo Drive End Bearing.....	53
<i>Figura 29.</i> Señal tipo Fan End Bearing	53
<i>Figura 30.</i> Librerías de Python	55
<i>Figura 31.</i> Método de entrenamiento	55
<i>Figura 32.</i> Características de las muestras para entrenar KNN	58
<i>Figura 33.</i> Características de los datos SVM.....	60
<i>Figura 34.</i> Características de los datos Random Forest.....	64
<i>Figura 35.</i> Grafica de importancia de variable	66
<i>Figura 36.</i> Tiempo estimado de análisis por etapa	67

ÍNDICE DE TABLAS

Tabla 1. <i>Aplicación en el sector Automotriz</i>	5
Tabla 2. <i>Aplicación en la Industria 4.0</i>	6
Tabla 3. <i>Aplicación en el Cuidado de la Salud</i>	6
Tabla 4. <i>Registros de valores de temperatura</i>	8
Tabla 5. <i>Valores mínimos seguros recomendadas para maquinas rotativas</i>	18
Tabla 6. <i>Matriz de Confusión</i>	43
Tabla 7. <i>Características del rodamiento</i>	51
Tabla 8. <i>Data de fallo de rodamiento</i>	51
Tabla 9. <i>Matriz de Confusión de KNN_DE</i>	58
Tabla 10. <i>Índices de Calidad del Modelo KNN_DE</i>	59
Tabla 11. <i>Matriz de Confusión SVM_Lineal_DE</i>	60
Tabla 12. <i>Índices de Calidad del Modelo SVM_Lineal_DE</i>	61
Tabla 13. <i>Matriz de Confusión SVM_Poly_DE</i>	61
Tabla 14. <i>Índices de Calidad del Modelo SVM_Poly_DE</i>	62
Tabla 15. <i>Matriz de Confusión SVM_Rbf_DE</i>	62
Tabla 16. <i>Índices de Calidad del Modelo SVM_Rbf_DE</i>	63
Tabla 17. <i>Cuadro Comparativo</i>	63
Tabla 18. <i>Matriz de Confusión Random Forest_DE</i>	65
Tabla 19. <i>Índices de Calidad del Modelo Random Forest_DE</i>	65
Tabla 20. <i>Importancia de la variable del Modelo Random Forest_DE</i>	66
Tabla 21. <i>Comparativo de Matrices de Confusión</i>	68
Tabla 22. <i>Comparativo de modelos</i>	69
Tabla 23. <i>Comparativo de categoría por modelos</i>	69

INTRODUCCIÓN

La industria en el presente está tomando un papel importante en la sociedad, se considera un motor de innovación y crecimiento. Sin embargo, la competencia es cada vez más intensa y solo aquellas empresas e industrias que logren conseguir productos de mayor calidad disminuyendo los tiempos de producción con la máxima eficiencia y eficacia en sus plantas, serán más competitivas. Una de las soluciones acorde a estos tiempos que se plantea, es utilizar técnicas que envuelven a la industria 4.0, esto supone un cambio de mentalidad importante en el mantenimiento industrial, a la vez un cambio al modelo conocido hasta el momento y reemplazarlo por el nuevo modelo que integra, la industria del internet de las cosas (IIoT), las redes de sensores, el Big Data, Cloud computing, etc.

El motor de inducción es uno de los activos más importantes que tiene la industria en la actualidad, aun siendo muy fiable necesita un mantenimiento para alargar su vida útil. Las técnicas más usuales han sido el mantenimiento preventivo, correctivo y predictivo, estas técnicas no ayudan a aprovechar totalmente la vida útil del componente o elemento que compone al motor de inducción. Es por ello que el desarrollo del presente trabajo tiene como objetivo, utilizar las técnicas predictivas más modernas para el diseño y desarrollo de un modelo de mantenimiento predictivo, que sea eficiente y competitivo para el motor de inducción.

CAPITULO 1

ASPECTOS GENERALES

1.1. Definición del problema

1.1.1.Descripción del problema

En el Perú, el mantenimiento industrial enfrenta un cambio y desarrollo tanto en lo económico como en lo tecnológico, el mantenimiento bien diseñado es importante en las industrias, porque esto generara una mayor eficiencia en su producción, evitara paros de equipos no deseados, mayor eficiencia en el consumo de energía, etc. Pero “una instalación, que no cuenta con un adecuado plan de mantenimiento dirigido a disminuir la carga correctiva y las paradas inesperadas, puede suponer al menos una pérdida del 10% de la producción anual, atribuible exclusivamente a la reparación de averías que se pueden evitar” (IRIM, 2018).

En la actualidad, una gran parte de los motores eléctricos en el sector industrial son de inducción.El motor de inducción, a pesar que es muy seguro y fiable, necesita de un adecuado mantenimiento preventivo, para su buen funcionamiento y alargar la vida útil de sus elementos que la componen. Sin embargo, hay que conocer los problemas que se genera al no realizar un mantenimiento adecuado cuando toca, los principales problemas según Del Pozo Gallego, 2018 son: fallo del bobinado del motor con un 37%, fallo del rotor con un 10%, fallo del rodamiento con un 42%,

otros fallos mecánicos con 11%. Adicionalmente en la publicación de SFK, 2017 nos menciona que un porcentaje del 9.5% se reemplazan los rodamientos antes de que fallen por razones de seguridad.

Las técnicas más usadas de mantenimiento hasta el momento han sido las siguientes: mantenimiento preventivo, efectuando la sustitución de componentes cada número de horas prefijadas. Mantenimiento predictivo basado en una serie de ensayos no invasivos con el fin de monitorear los equipos con el fin de detectar signos de advertencia que indiquen que alguno de sus componentes no esté funcionando correctamente; estas técnicas no ayudan a optimizar costes ni aprovechar totalmente la vida útil del componente o elemento que compone al motor de inducción.

1.1.2. Formulación del problema

Problema General: ¿De qué manera el diseño de un modelo de mantenimiento predictivo basado en las técnicas de la industria 4,0 evitara los paros inesperados de los motores de inducción y alargara la vida útil de sus componentes?

Problema Específico: ¿De qué manera la gestión de datos recabados de los rodamientos de un motor de inducción en funcionamiento, durante y después de que se averíe nos ayudara a predecir el grado de falla?, ¿De qué manera el diseño del algoritmo de Machine Learning del modelo predictivo nos ayudara a la predicción de futuros fallos en los motores de inducción?

1.2. Definición de objetivos

1.2.1. Objetivo general

Diseñar un modelo de mantenimiento predictivo en motores de inducción utilizando técnicas de la industria 4.0.

1.2.2.Objetivos específicos

Obtener una adecuada base de datos referido a la falla de los rodamientos del motor de inducción.

Diseñar la programación más adecuada para el entrenamiento del modelo predictivo.

Obtener los algoritmos de clasificación más adecuados para el entrenamiento del modelo predictivo.

Diseñar el índice de calidad para cada categoría y clase de fallo y seguidamente seleccionar el modelo predictivo más eficiente para la solución del problema planteado.

1.3. Alcances y Limitaciones

1.3.1.Alcance

En esta tesis abarcaremos el principio de funcionamiento y fallas del motor de inducción, diversas técnicas de mantenimiento, estudio de los algoritmos de Machine Learning para la predicción de la gravedad de falla del rodamiento, No se contempla el diseño de los diferentes sensores, actuadores y controladores necesarios para el correcto funcionamiento del modelo, solo a la selección del software de programación y su parametrización de productos ya existentes que tienen gran confiabilidad en su funcionamiento.

1.3.2.Limitación

Teórica. - Durante el desarrollo de la presente investigación se utilizará las teorías, enfoques teóricos y vocabulario propios del mantenimiento predictivo basado en la Inteligencia Artificial y Machine Learning, así como los relacionados en el proceso operacional de los motores de inducción, para tal efecto emplearemos las tesis, revistas especializadas, artículos científicos actuales referentes al campo que se

estudiara y se tomara diferentes fuentes documentales para resolver de la mejor manera el problema.

Temporal. - El estudio de la presente investigación es de tipo longitudinal esto quiere decir que el desarrollo de la investigación es relativamente corto y durante la cual se observara las variaciones desde el inicio hasta el final.

Espacial. - Nuestra unidad de análisis son los motores de inducción y sus rodamientos, con lo cual los resultados de nuestro análisis se limitan a las características propias de este tipo de motores.

1.4. Justificación

En el aspecto del mantenimiento. – Se optimizará el tiempo de vida útil de los elementos, en este caso en particular, se tomará los rodamientos que están comprendidos en el motor de inducción. Optimizar los planes de mantenimiento correctivo. Llegar a eliminar las paradas inesperadas. Difundir el mantenimiento predictivo automatizado basado en la inteligencia artificial y Machine Learning como filosofía y herramienta de mantenimiento en la industria nacional.

En el aspecto de costos. - Se optimizará los costos del mantenimiento programado.

Se eliminará los costos de mantenimiento no programado. Se mejora de gran manera la administración de contratos destinados a los servicios de mantenimiento. Se reducirá las pérdidas de producción causadas por las paradas inesperadas.

1.5. Estado del arte

La información descrita a continuación nos muestra un panorama de cómo se encuentra la tecnología que se pretende utilizar para resolver la problemática expuesta. Asimismo, se explica algunas soluciones planteadas por otros autores.

El concepto de Industria 4.0, surge en Alemania en el año 2011, cuando el gobierno alemán y el sector empresarial, encabezado por la firma Bosch, conforman un grupo de investigación para encontrar un marco común que permitiera la aplicación de las nuevas tecnologías emergentes,

entregando su primer informe en el 2012, que luego fue presentado en público durante la Feria de Hannover en 2013.(Candanedo, González, & Muñoz, 2018)

En la actualidad la inteligencia artificial (Machine Learning y Deep Learning), robótica, nanotecnología, biotecnología, vehículos autónomos son algunas de las tecnologías disruptivas que están revolucionando la economía mundial, redefiniendo la forma de trabajo de industrias existentes y creando otras nuevas. A esta ola creciente de innovación el Foro Económico Mundial (WEF por sus siglas en inglés) la bautizó con el nombre de la Cuarta Revolución Industrial (4RI).(Peñaranda Castañeda, 2018).

A continuación, presentamos algunas aplicaciones en los diferentes sectores

Tabla 1. *Aplicación en el sector Automotriz*

Aplicación	Tipo	Impacto	Riqueza de datos
Identificar y navegar carreteras y obstáculos en tiempo real para una conducción autónoma	Procesar datos no estructurados	Alto	Alto
Predecir fallas y mantenimiento proactivo recomendado en el vehículo	Mantenimiento predictivo	Medio	Alto
Optimizar las opciones de carriles y el enrutamiento de rutas en función de datos multimodales para reducir la duración del viaje	Optimización en tiempo real	Medio	Alto

Fuente: Elaboración propia

Tabla 2. Aplicación en la Industria 4.0

Aplicación	Tipo	Impacto	Riqueza de datos
Predecir fallas y recomendar mantenimiento proactivo para equipos de producción y movimiento	Mantenimiento predictivo	Alto	Medio
Predecir el rendimiento para la agricultura aprovechando los datos del sensor IoT y otros datos relevantes	Predicción	Medio	Bajo
Predecir las tendencias futuras de la demanda y las limitaciones potenciales en la cadena de suministro	Predicción	Medio	Medio

Fuente: Elaboración propia

Tabla 3. Aplicación en el Cuidado de la Salud

Aplicación	Tipo	Impacto	Riqueza de datos
Diagnostica enfermedades conocidas a partir de escaneos, biopsias, audio y otros datos	Analítica Predictiva	Alto	Muy bajo
Optimiza la dotación de personal y la asignación de recursos para reducir los cuellos de botella	Asignación de Recursos	Medio	Bajo
Optimiza el diseño de ensayos clínicos, incluida la redacción de etiquetas y la selección de pacientes	Mejoramiento	Alto	Bajo

Fuente: Elaboración propia

1.5.1. Soluciones Tecnológicas

Aplicación de técnicas de Machine Learning con regularización al diagnóstico de fallos en motores de inducción (Del Pozo Gallego, 2016)

En la tesis desarrollada por el Ing. del Pozo propone utilizar el aprendizaje automático o más conocido por su nombre en inglés como Machine Learning. Él desarrollo un modelo de mantenimiento predictivo a los componentes del motor de inducción, en este caso particular a los rodamientos, para esto utilizó la técnica de clasificación, entre ellas la regresión logística con regularización de LASSO. Esto permite aplicar un mantenimiento predictivo eficaz. Utilizando como ayuda el software MATLAB, aprovechando la herramienta GUIDO de MATLAB donde crea una interfase gráfica donde se puede clasificar los estados del motor en una forma más sencilla. En la figura 1. ilustra el esquema utilizado para la toma de datos del estator del motor.

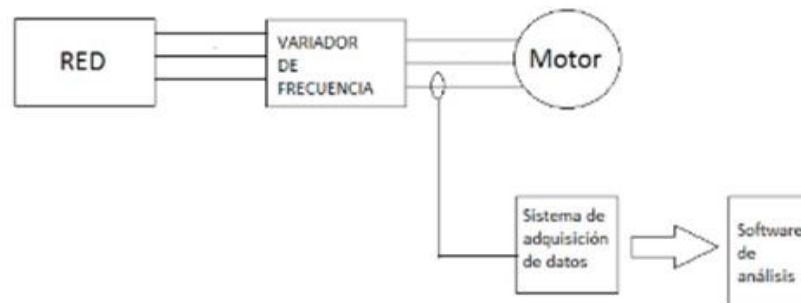


Figura 1. Proceso de adquisición de datos
Fuente: Del Pozo Gallego, 2016)

Machine Learning Predictive Model for Industry 4.0 (Sitton, Hernández, Rodríguez, Santos, & Gonzales, 2018)

Este trabajo se presenta una aplicación del mantenimiento predictivo en la industria 4.0 direccionado a los sistemas de calefacción, ventilación y aire acondicionado

(HACE). Estos equipos son importantes en el funcionamiento de una fábrica en el contexto de la industria 4.0.

En este caso de estudio se utilizó un conjunto de datos organizados por columna, los cuales contienen los registros de los valores de temperatura óptima y valores reales medios capturada por una red de sensores, instaladas en un conjunto de edificios que tenían entre 0 a 30 años, en 20 edificios. La Tabla 4. muestra la estructura del conjunto de datos y sus variables.

Los resultados presentados al utilizar los algoritmos de supervisión: regresión logística binaria y bosque aleatorio (RF) fueron del 65% y 64% respectivamente los cuales no son muy óptimos en comparación de otros métodos como por ejemplo las redes neuronales que nos dan un mayor porcentaje de predicción, como menciona el artículo en sus conclusiones el modelo aún se encuentra en una etapa inicial de desarrollo.

Tabla 4. *Registros de valores de temperatura*

Column	Description
Date	Date of measurement
Time	Measurement time
Target temp	Temperature measured by the sensor
Actualtemp	Optimal temperature for the system
System	System model
System age	Age of the HVAC system
Building id	Building identifier

Fuente: Sitton et al., 2018

Mantenimiento Predictivo Mediante Técnicas de Machine Learning (Cano et al., 2019).

En este artículo los autores pretenden implementar un plan piloto utilizando el Machine Learning o también conocido como aprendizaje automático en la planta de embotellado de Aguas de Suelan de Cabras ubicada en Beteta (Cuenca, Ecuador).

En la cual toma como objeto de estudio una máquina de llenado de chorro libre Modulillo del fabricante alemán Crones que se muestra en la Figura 2. implementando a la máquina un conjunto de sensores que puedan recabar todos los datos de los elementos que la componen, lo que plantean es utilizar el aprendizaje no supervisado esto quiere decir que utilizaron un conjunto grande de datos no clasificados, a partir de esto se busca propiedades, características particulares en los datos, segmentándolos en grupos de datos similares llamados también clúster. Es una técnica que está desarrollándose con gran rapidez y la denominan Deep Learning que es un sub conjunto del Machine Learning.



Figura 2. Máquina Llenadora Crones
Fuente: Cano et al., 2019

Inteligencia Artificial e Innovación para Optimizar el Proceso de Diagnóstico de la Tuberculosis (Curioso & Brunette, 2020)

En la presente revista se presenta la aplicación de la inteligencia artificial y Machine Learning en el campo de la medicina, específicamente en el diagnóstico de la tuberculosis en el Perú.

El artículo sirve de referencia para poder entender el comportamiento del modelo que contempla el aprendizaje supervisado que comprende entre una de ellas a las redes neuronales convolucionales.

El proyecto desarrollado en este artículo se denomina “eRx”; en lo cual los profesionales de enfermería o personal técnico capacitado utilizan teléfonos inteligentes como dispositivos para capturar imágenes de las radiografías de tórax de los pacientes en los establecimientos de salud, esto se muestra en la Figura 3. Posteriormente, las imágenes de rayos X se transmiten a través de una aplicación (App) previamente instalada en el sistema operativo del dispositivo móvil, esto se puede observar en la Figura 4. Este modelo está diseñado en base a métodos de inteligencia artificial y Machine Learning, utilizando redes neuronales convolucionales. Dicha aplicación permite detectar anomalías pulmonares y sugiere manifestaciones clínicas preliminares de la tuberculosis.

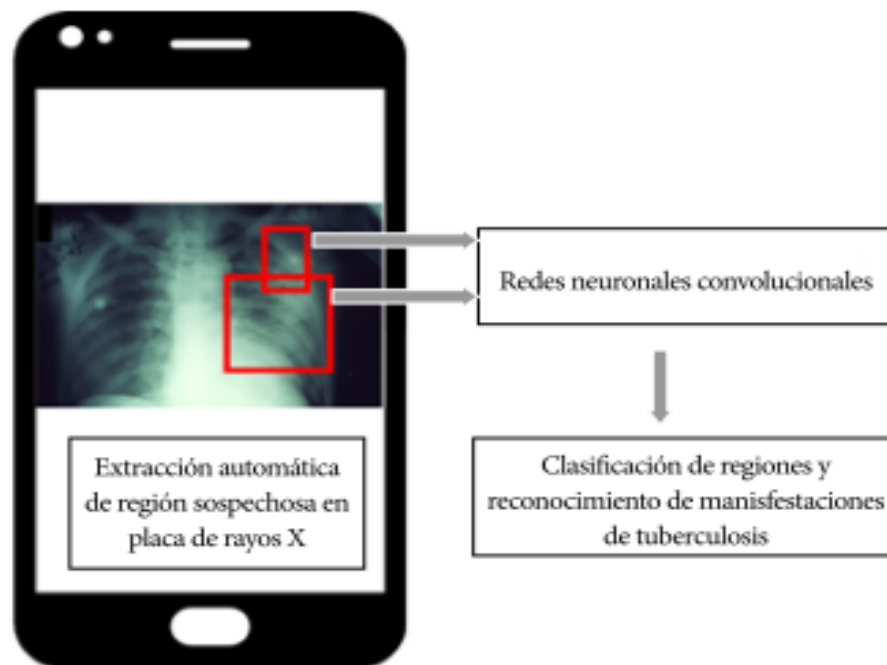


Figura 3. Captura de imagen de placa de rayos x
Fuente: Curioso & Brunete, 2020



Figura 4. Personal utilizando el aplicativo “eRx”
Fuente: Curioso & Brunette, 2020

CAPITULO 2

MARCO TEÓRICO

2.1. Teoría del funcionamiento del motor de inducción

El motor eléctrico de inducción funciona basándose en la teoría de campos magnéticos giratorios. El principio comprende en alimentar con tensión trifásica de corriente alterna a las tres bobinas desfasadas en un tercio de vuelta entre ellas, y equidistantes al rotor de jaula de ardilla, esto generaría un campo magnético giratorio, que podría llegar a tener una frecuencia de giro de 50 o 60 hercios, esto dependerá la frecuencia que utiliza el país (Del Pozo Gallego, 2018).

Según Faraday, la fuerza electromotriz inducida existente en el rotor es directamente proporcional al campo magnético en función del tiempo, y al número de espiras de cada bobina.

E. (1)

$$e = -N \frac{d\phi}{dt} \quad \text{E. (1)}$$

Una vez inducida la tensión en el rotor, comenzará a circular corriente en las barras de la jaula de ardilla del motor de inducción, que conjuntamente con el campo magnético, producirá un par de fuerzas como establece la ley de Faraday. Su expresión matemática se verá en la Ec. (2) (Enriquez Harper, 2001)

$$F = i \int i(dIxB) = i(LxB) \quad \text{Ec. (2)}$$

La velocidad de sincronismo también conocido como velocidad de giro es una parte muy importante en el funcionamiento de motor de inducción. Su expresión matemática se verá reflejada en la Ec. (3)

$$n_s = \frac{60f_1}{p} \quad \text{Ec. (3)}$$

Donde:

n_s velocidad de sincronización.

f_1 frecuencia de corriente.

p número de pares de polos.

Otro factor que debemos tener en cuenta para que pueda funcionar adecuadamente el motor es, el “deslizamiento” el cual es fundamental para determinar las corrientes que circulan por el rotor, estas corrientes nos indicaran perdidas en el cobre, calor fuera de lo normal entre otros (Enríquez Harper, 2001)(Martin & Garcia, 2016). Su expresión matemática será veré en la Ec. (4):

$$S = \frac{n_s - n}{n} \quad \text{Ec. (4)}$$

Donde:

S Deslizamiento

n_s velocidad de sincronismo

n velocidad de funcionamiento

2.2. Tipos de motores de inducción

Motor de inducción de jaula de ardilla. - es el motor más utilizado en la industria actualmente, las características que lo hacen singular van desde su simplicidad, robustez, mantenimiento sencillo, fácil conexionado, larga vida útil. Su constitución es la siguiente: el estator donde se ubican las bobinas, un rotor de la forma de jaula de ardilla, eje, rodamientos, borneras, carcasa y placa de características, (José & Rosales Fernández, 2016). Figura 5.

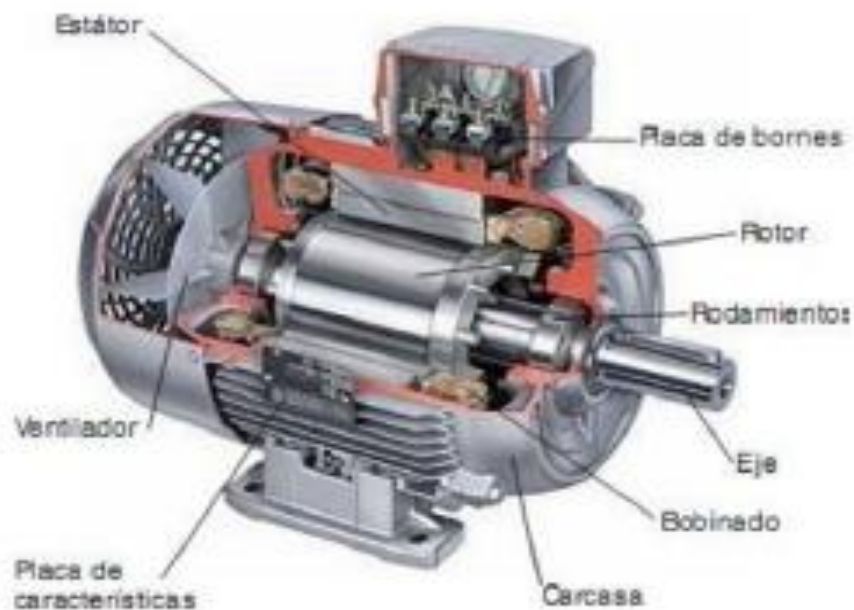


Figura 5. Motor de inducción de jaula de ardilla
Fuente: Martín & García, 2016

Motor de inducción monofásico. - el uso que se le da cada vez es más para trabajos específicos por su poca potencia. Para este tipo de motores se utiliza un sistema auxiliar de arranque y esto se debe a que hay un problema en común que es el arranque del motor, puesto que para producir el par de arranque se requiere un tipo de medio auxiliar que reaccione con la onda de C.A de la alimentación y una vez producido la reacción se genera el par y por este motivo se está reduciendo cada vez su uso en la industria. (Martín & García, 2016)

Los usos más comunes son los domésticos como ventiladores, refrigeradoras, lavadoras. En la Figura 6 podemos ver el motor de inducción monofásico.



Figura 6. Motor de inducción monofásico
Fuente: Enriquez Harper, 2001

Motor de inducción de doble jaula de ardilla.- fue dado a conocer por Dolivo-Dobrowollsky en 1883 y desarrollado comercialmente por Boucherot, reúne las ventajas de un buen rendimiento y un importante par de arranque, sin un excesivo valor de la corriente de la punta de conexión (Cortes Cherta, 1994).

Es un motor trifásico en jaula de ardilla que se caracteriza por tener dos rotores concéntricos, la jaula externa limita la corriente durante el arranque y la jaula interna mejora el funcionamiento del motor en arranque.

Motor de rotor bobinado o anillos rozantes. - los anillos servían para limitar las corrientes durante el arranque y mejorar el par de arranque. Actualmente, esa función se realiza mediante un variador de frecuencia. Estos motores son muy recomendados para aplicaciones que requieren elevada potencia, se utilizan en la industria de materias primas.



Figura 7. Rotor de doble jaula de ardilla
Fuente: Cortes Cherta, 1994

2.3. Fallos más comunes en los motores de inducción

En esta parte nos enfocaremos en analizar las causas que pueden provocar que el motor se averíe. Las fallas más frecuentes en el motor de inducción son: los rodamientos con un 42%, y esto se debe al desgaste por carga de trabajo, otro factor es un mal diseño. Las fallas en la bobina del estator con un 37%, y esto le ocurre al motor por que ha sido expuesto a una sobre carga por mucho tiempo, factores como la humedad también afectan el aislamiento a la bobina del estator. La falla por excentricidad con un 10%, esto se produce en situaciones puntuales como un mal alineamiento entre el eje del motor y una carga móvil. (Del Pozo Gallego, 2018)

Los fallos más comunes lo podemos observar en la Figura 8

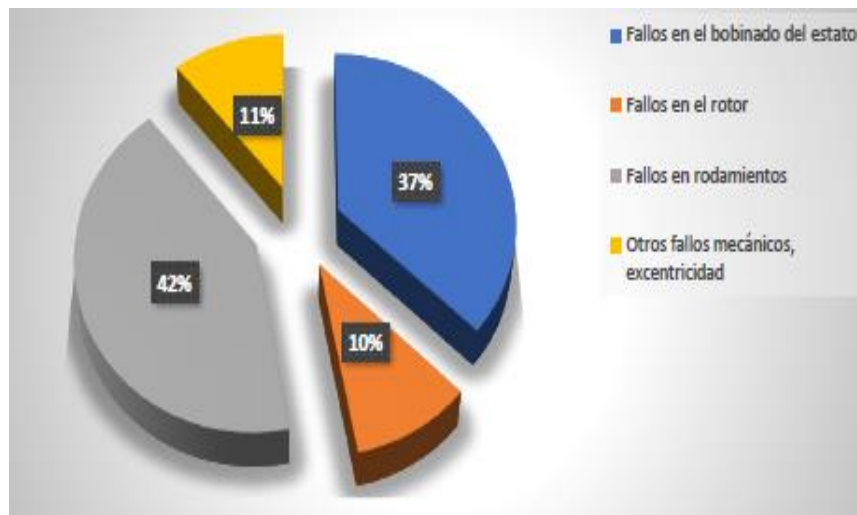


Figura 8. Fallas comunes en el motor de inducción
 Fuente: Del Pozo Gallego, 2018

2.3.1. Fallas en el aislamiento en las bobinas del estator

El deterioro del aislamiento en las bobinas del estator se debe a diferentes causas y estas pueden ser: factores ambientales, eléctricos, mecánicos, etc. en la siguiente Figura 9. se presenta las diferentes causas de fallo que sufre el estator en referencia a su aislamiento.

La metodología para poder conocer la resistencia del aislamiento es un test que se realiza aplicando una tensión a la bobina durante un minuto para posteriormente tomar el valor medido.

Según (La Torre García, Universidad Nacional de San Agustín, & Empresa de Generación Eléctrica de Arequipa, 2017) : “El estándar 43-2000 del IEEE establece el nivel de aislamiento mínimo corregido a 40 °C, así como la tensión de prueba a ser aplicada de acuerdo a la tensión nominal de la máquina rotativa” el estándar lo podemos observar en la Tabla 5.

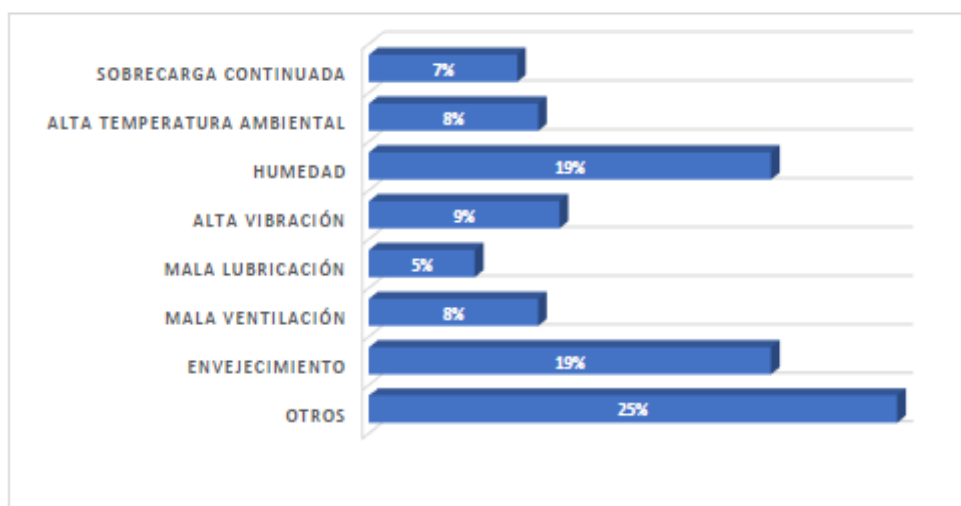


Figura 9. Factores más comunes que deterioran el aislamiento del motor
Fuente: IEEE,85

Tabla 5. Valores mínimos seguros recomendadas para maquinas rotativas

Tipo de motor	Resistencia de aislamiento	Tensión nominal del motor	Tensión de prueba
Bobinado antes de 1970	$> 1M\Omega + 1M\Omega/kv.$	Menos de 1000 V.	500 V.
Bobinado después de 1970	$> 100 M\Omega.$	1000 V. a 2500 V.	500 V. a 1000 V.
Bobinado preformado y armadura de CC	$> 5 M\Omega.$	2501 V. a 5000 V.	1000 V. a 2500 V.
		5001 V. a 12000 V.	2500 V. a 5000 V.
		12001 V. a más	5000 V. a 10000 V.

Fuente: La Torre Garcia et al., 2017

2.3.2. Fallas en los rodamientos

Los cojinetes o llamados también rodamientos son elementos mecánicos encargados de unir partes distintas de una maquina en movimiento rotativo entre sí, de este modo los rodamientos transmiten la fuerza y movimiento a los componentes, y a la vez ayuda a reducir la fricción que hay entre ellos, es por eso que los rodamientos son unos de los elementos más importantes en la mayoría de

los equipos. Soportan grandes exigencias con respecto a su capacidad de carga, funcionamiento, precisión, niveles de ruido, fricción y calor de fricción, vida y confiabilidad. Así mismo, los rodamientos están constituidos por diferentes elementos que pasamos a mencionar a continuación:(SFK, 2017)

Anillo Interior. - Este elemento esta encargada de soportar la fatiga ocasionada por los cambios de presión en diferentes zonas donde se produce el contacto del componente interior.

Anillo Exterior. - Este elemento esta encargada de soportar la fatiga ocasionada por los cambios en la presión en la zona de contacto del componente exterior.

Elemento Rodante. - Son elementos que están contenidos en una jaula entre el anillo interior y el exterior, estos son los encargados de transferir la carga y movimiento entre los anillos interior y exterior. Los más usados común mente son los rodamientos de bola, aunque existen otros tipos como los de rodillo o de agujas.

Jaula. - Es una estructura que mantiene separado los elementos que componen el anillo interior como el exterior, de modo que se mantengan guiados por la pista de rodadura que conforma el anillo exterior conjuntamente con el interior y equiespaciado para que la carga sea homogénea (Serano, 2019).

Los elementos descritos lo podemos observar en la Figura 10.

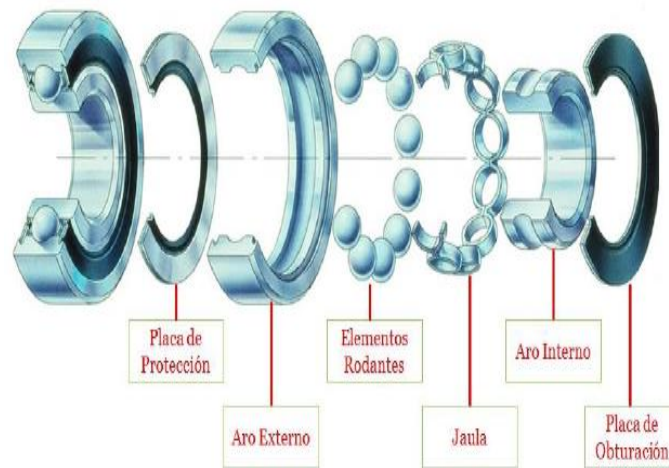


Figura 10. Componentes del rodamiento
Fuente: SFK. 2017

Cada componente del rodamiento es susceptible a fallar debido a desgaste, una pésima lubricación, fracturas debido a fallas aparecidas durante el proceso de fabricación, etc. El rodamiento tiene la peculiaridad de que cada falla deja su propia huella especial en el ella. En consecuencia, al revisar y examinar cuidadosamente el rodamiento dañado, en la mayoría de los casos se puede establecer la causa raíz y definir acciones correctivas para evitar que vuelva a suceder, existen diversas técnicas basadas en el mantenimiento predictivo muy efectivas que permiten una buena y temprana detección. (SFK, 2017),(Serano, 2019). En la siguiente Figura 11. Veremos representada gráficamente el versus del porcentaje de cambios de rodamientos por mantenimiento preventivo entre el cambio de rodamiento por falla.

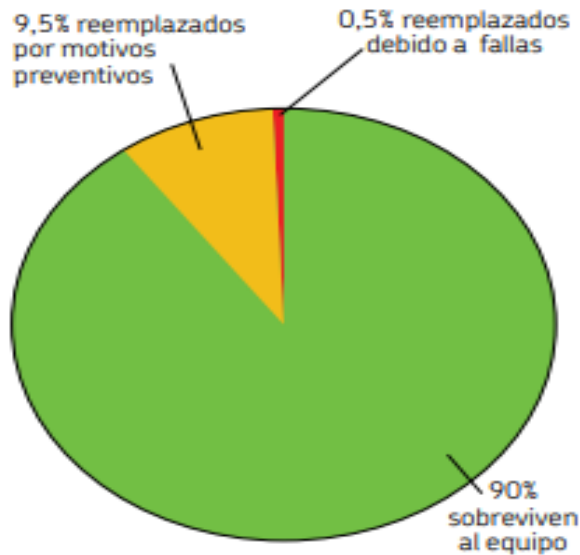


Figura 11. Falla de rodamientos
Fuente: SFK, 2017

2.3.3. Fallas por causa de excentricidad

Cuando ocurre una falla por excentricidad sucede que la distancia del entrehierro no es uniforme. Cuando el entrehierro no es uniforme, se genera un espacio lleno de aire y este actúa como un elemento dieléctrico que resiste al campo magnético. Se produce una atracción máxima, cuando entre el estator y el rotor existe un área muy pequeña de entrehierro, y mínima cuando el área existente entre el estator y el rotor es de mayor entrehierro. El desequilibrio que generan estas fuerzas puede ocasionar un daño severo al eje del rotor, lo peor que puede pasar es el roce entre el eje del rotor y el estator que ocasionaría un grave daño al motor. En la figura 12. mostraremos los diferentes tipos de excentricidad .(Del Pozo Gallego, 2018).

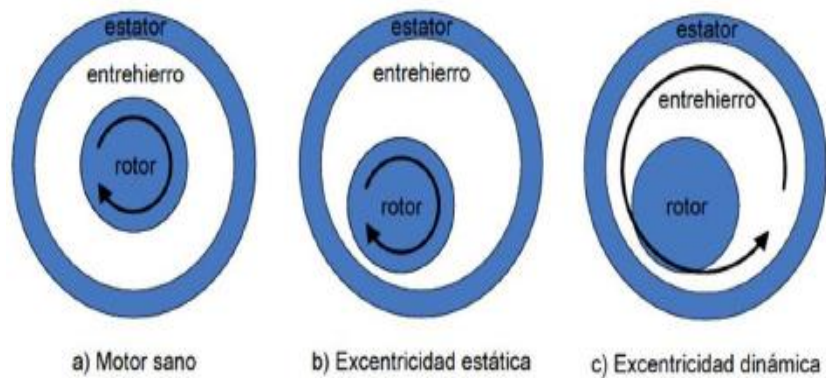


Figura 12. Tipos de Excentricidad
 Fuente: Del Pozo Gallego, 2018)

2.4. Técnicas de mantenimiento

Se entiende por mantenimiento a un conjunto de acciones o técnicas que logran alargar la vida útil de un equipo, asegurar el servicio a un coste mínimo y garantizar una máxima seguridad.

La importancia del mantenimiento industrial tiene su origen en la necesidad de las empresas en mantener sus equipos y maquinarias trabajando continua y eficientemente. Existen diferentes tipos de mantenimiento, las más frecuentes son:

2.4.1. Mantenimiento Correctivo

Es un conjunto de acciones direccionadas a la reparación del equipo cuando ha dejado de funcionar, Este mantenimiento puede ser planificado o no planificado, que lleva asociado unos costes bastante altos. (Da Costa Burga, 2010)

Esta técnica es basada en la intervención, tras suceder una falla manifestada mediante una parada de un equipo eléctrico, mecánico, instalación eléctrica o hidráulica, es decir cuando se produce una parada inesperada de la producción en la fábrica. Por otro lado, el mantenimiento correctivo, presenta muchas desventajas, como falta de prevención de las fallas, los suministros, inseguridad al personal, y sumado a esto el alto costo por las paradas súbitas e improductivas. Actualmente,

se utiliza solo cuando el comparativo de costos de las paradas de producción sean menores a diferentes tipos de mantenimiento preventivo, por lo que se recomienda utilizarlo en sistemas secundarios que no afectan en la producción. (Picazo Rodenas, 2016)

2.4.2.Mantenimiento Preventivo

Es un conjunto de acciones encaminadas a reducir la probabilidad de fallo y degradación de un equipo o máquina. Es un mantenimiento planificado, que lleva asociado unos costos bastante altos.

Este tipo de mantenimiento nació para dar una solución a los inconvenientes que tenía el mantenimiento correctivo. Esta técnica se basa en realizar las inspecciones a los equipos y máquinas antes de que se suceda la falla, considerando intervalos de tiempo previamente definidos, de manera de prevenir la falla antes de que ocurra (Da Costa Burga, 2010). El periodo de tiempo se calcula mediante una metodología que basa su análisis en la estadística, partiendo de los históricos de sucesos disponibles. El principio del mantenimiento consiste en el reemplazo o cambio de piezas o componentes del equipo, independientemente del estado del que se encuentra, así como otros procedimientos de limpieza y lubricación. A todo esto, se le conoce también como mantenimiento programado o sistemático. Los puntos débiles o inconvenientes que se observa en este mantenimiento es, la pérdida económica al no aprovechar la vida útil de las piezas y recursos.(Picazo Rodenas, 2016).

2.4.3.Mantenimiento Predictivo

Es un conjunto de acciones encaminadas a pronosticar el punto futuro de falla del componente de una máquina. Así, el tiempo muerto del equipo se minimiza, este tipo de mantenimiento reduce considerablemente los costos. Consiste en una serie

de pruebas que está orientado a realizar un seguimiento al funcionamiento de los equipos con el fin de detectar signos de advertencia que indiquen que alguna de sus partes no esté realizando su trabajo de la manera adecuada, una vez detectada y localizada las fallas, se puede, de manera oportuna, programar las intervenciones y reparaciones de las máquinas sin que este afecte el proceso de producción. Otros de sus beneficios es prolongar la vida útil de las piezas que comprenden las máquinas (Serano, 2019). Las técnicas que más utilizan en las industrias son las siguientes:

Termografía. - Esta técnica se fundamenta en el estudio del comportamiento de la temperatura de las máquinas, con el fin de determinar su correcto funcionamiento. Las máquinas emiten desde su superficie energía, que se desplaza en forma de ondas electromagnéticas, estas son a su vez directamente proporcionales a su temperatura, estas ondas se pueden captar con un instrumento que tiene la propiedad de transformar estas ondas electromagnéticas en espectros visibles para su estudio y análisis respectivo (Botero, Cañon, & Olarte, 2010). En la Figura 13. se observa la cámara termográfica utilizado para generar una imagen de radiación infrarroja.



Figura 13. Cámara Termográfica
Fuente: SFK TKTI 31

Análisis por Ultrasonido. - Esta técnica, se basa en el estudio y análisis de las ondas de sonido de alta frecuencia que las maquinas emiten cuando comienzan a presentar fallas. Las maquinas cuando comienzas a presentar problemas emiten sonidos con frecuencias superiores a los 20khz. Están son muy altas a comparación del rango de frecuencia (20Hz – 20 kHz) que el oído humano pueda escucharlas. Por otro lado, las ondas ultrasónicas tienen la propiedad de atenuarse rápidamente y esto se debe a su corta longitud de onda, y por consecuencia se puede detectar con mayor facilidad la fuente que lo emite aun estando en ambientes con mucho ruido. Los instrumentos de medición encargados de poder medir las ondas ultrasónicas se llaman medidores ultrasónicos. Gracias a este instrumento la persona puede oír por medio de audífonos u observarla en una pantalla para su respectivo análisis y estudio (Botero et al., 2010). Esto se puede observar en la figura 14.

Esta técnica nos permite:

Detectar la fricción en máquinas rotativas

Detectar fallas y/o fugas en válvulas

Detectar fugas de fluido

Detectar perdidas de vacío

Detectar arco eléctrico



Figura 14. Medidor de Ultrasonido
Fuente: CST Perú S.A.C.

Análisis y estudio de la onda espectral de la corriente del estator del motor. - Esta técnica se complementa con el análisis de vibraciones, ya que esta última tiene un alcance de diagnóstico limitado como por ejemplo las fallas eléctricas incipientes, excentricidades, barras rotas de los rotores producidas por demasiado esfuerzo. Una de las aplicaciones de este análisis es la detección y monitoreo de cortocircuitos que hay entre las espiras en el devanado estático de los motores de inducción (Stavrou, Sedding, & Penman, 2001).

Cuando se presenta un fallo de rodamiento a un motor de inducción, se debe a que existe una distancia que no es uniforme en el entrehierro del motor, por consecuencia se produce un campo magnético desequilibrado. Por lo tanto, se presentan diferentes armónicos y esto nos hace concluir que la corriente de alimentación al motor también presenta un desequilibrio. Analizando los armónicos que produce la corriente se puede determinar el estado del rodamiento. Las ventajas que presenta esta técnica son las siguientes:

Con esta técnica no requiere que el motor este parado.

Tiene una alta efectividad en el diagnóstico de falla del equipo.

La toma de corriente que alimenta al equipo se realiza desde el sub cuadro que energiza al equipo, que en este caso viene a hacer el motor.

El riesgo de accidentes se reduce considerablemente para el operario, ya que no tiene contacto con el equipo.

En el caso particular del rodamiento del motor se puede monitorear su condición en cada instante.

Se produce una mejora en la transferencia de datos.

Los armónicos que se originan de las corrientes que circulan por el estator y del rotor, pueden ser hallados respectivamente mediante las siguientes ecuaciones (Stavrou et al., 2001):

$$f_{sc} = (j_{rt} \times R) \left(\frac{1-s}{p} \right) \pm 2j_{sa} \pm i_{st} f_1 \quad \text{Ec. (5)}$$

$$f_{rc} = ((j_{rt} \times R \pm k) \left(\frac{1-s}{p} \right) \pm 2j_{sa} \pm i_{rt} \times S) f_1 \quad \text{Ec. (6)}$$

En donde:

f_1 : frecuencia de la alimentación del motor.

i, j, k : número entero.

p : número de polos del motor.

R : número de ranuras del rotor.

s : número de ranuras del estator.

S : deslizamiento.

Análisis de vibraciones mediante procesamiento de señales. – Es una de las técnicas más utilizadas, esto gracias a lo fácil de su implementación y su alto grado de eficacia. Mediante diferentes técnicas de procesamiento de señales, se puede analizar y estudiar la vibración del cojinete o rodamiento y sus elementos que la componen. Gracias a esta técnica es posible detectar el fallo con exactitud desde

que se origina, esto debido a la monitorización continua. La recopilación de datos se consigue utilizando unos sensores conocidos como acelerómetros que se encargan de transformar una señal de vibración en una señal de corriente. Esta se registrada mediante un equipo de muestreo que es capaz de convertir la señal analógica de entrada a una señal digital la cual es procesada adecuadamente por un software en una computadora (Serano, 2019). Para que tenga validez este método, es necesario conocer algunos datos del equipo como son: la velocidad de giro del motor, el tipo de rodamiento, de correas, el número de alabes, etc. Es muy importante determinar las áreas o lugares de las máquinas en donde se tomarán las mediciones para su respectivo análisis. El equipo analizador mostrado en la Figura 15 es el más adecuado para la realización del estudio.

Los fallos detectados por esta técnica son los siguientes (Botero et al., 2010):

- Fallo en el desalineamiento del eje,
- Desbalance del motor,
- Resonancia en el motor,
- Detección de Solturas mecánicas,
- Detección de rodamientos en mal estado,
- Detección de problemas en bombas,
- Detección de problemas en engranes,
- Detección de fallas eléctricos asociados a motores,
- Detección de problemas de bandas transportadoras,



Figura 15. Analizador de vibraciones
Fuente: SFK CMXA 75

La Case Western Reserve University, utilizo la técnica de análisis de vibraciones para diagnosticar la falla del cojinete o rodamiento del motor de inducción (Zhang, Zhang, Wang, & Habetler, 2020), será esta base de datos de acceso libre conjuntamente con la inteligencia artificial y Machine Learning el punto de partida de este trabajo de grado, para diseñar un modelo de mantenimiento predictivo.

2.5. Machine Learning

El aprendizaje automático (ML o Machine Learning en inglés) es una ciencia, que está incluida dentro de la rama de la inteligencia artificial la cual podemos ver en la Figura 16, que tiene como objetivo desarrollar técnicas que permitan a la maquina tener la habilidad de aprender de la experiencia.(Molnar, 2020) En los últimos años se vio incrementado de una forma exponencial la aplicación de estas técnicas, usándolas en actos tan cotidianos como llamar un taxi o pedir un delivery por aplicativo instalado en nuestro móvil. La inteligencia artificial, el Machine Learning y el Deep Learning ya están dentro de todos los campos de trabajo donde exista una gran cantidad de datos a estudiar, puede abarcar desde la medicina, la finanzas, el estudio del comportamiento humano, la industria etc. (Mansanet, 2017). El aprendizaje automatizado o Machine Learning esta tan desarrollado que puede detectar comportamientos inusuales en las personas. Podemos mencionar algunos ejemplos como pueden ser los diagnósticos de enfermedades, detectar fraudes cuando se usa tarjetas de crédito, análisis del mercado

financiero, clasificar las secuencias de ADN, reconocimiento de patrones en imágenes, reconocimiento de voz, etc. La inteligencia artificial es utilizada para el desarrollo de la sociedad en su conjunto como, por ejemplo, evitando accidentes, diagnosticando la tuberculosis en sus inicios(Curioso & Brunette, 2020), como se describe en el apartado 1.5.1 del presente trabajo, entre otras aplicaciones.

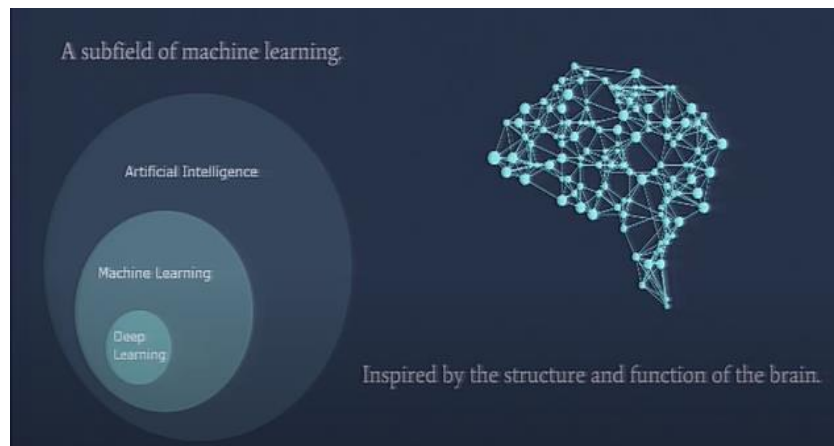


Figura 16. Sub campo de la inteligencia artificial
Fuente: Mansanet, 2017

Las categorías en Machine Learning son las siguientes: (Mansanet, 2017)

Aprendizaje Supervisado o Supervised Learning

Aprendizaje no supervisado o Unsupervised Learning

Aprendizaje Semi-supervisado o Semi-supervised Learning

Aprendizaje por Reforzamiento o Reinforcement Learning

2.6. Aprendizaje Supervisado

Esta técnica, consiste en la identificación y predicción de características propias (labels o etiquetas) a través de un conjunto de datos de entrenamiento para después de manera autónoma los identifique y pronostique por sí mismo, esto se puede ver claramente en la Figura 17.

Los algoritmos supervisados enfrentan dos problemas comunes al momento de ser entrenados con datos y estos son (Müller & Guido, 2016),(López & Manuel, 2019):

Sobreajuste (Overfitting): Esto se presenta cuando los datos de un modelo se han entrenado demasiado. Esto origina un impacto negativo en el rendimiento del modelo y el aprendizaje con los datos nuevos.

Correlación (Correlation): Esto 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(López & Manuel, 2019).



Figura 17. Captura de patrones
Fuente: Mansanet, 2017

El aprendizaje automático supervisado, se puede clasificar en dos tipos, los cuales estudian los problemas de Machine Learning, los cuales se detallan a continuación:

Clasificación. - El objetivo es predecir mediante la clasificación las etiquetas de clase, la cual viene a ser una opción predeterminada en una lista de posibilidades. Esto queda dividido en clasificación binaria, esta intenta responder una pregunta de si o no. Clasificación multiclase, consiste en clasificar entre más de dos clases que puede tolerar respuestas intermedias como bueno, leve o grave según sea el caso (Müller & Guido, 2016), (López & Manuel, 2019).

Regresión. - El objetivo de la regresión es número entero o un numero flotante en términos matemáticos y en términos de programación; por ejemplo, se puede predecir el ingreso semestral de un individuo considerando su edad, educación y lugar donde vive (Müller & Guido, 2016), (López & Manuel, 2019).

Según Nagy, (2018): "la tarea de la regresión debe predecir los valores de las etiquetas (label) en función de los valores de las características (features)".

Entre los algoritmos más utilizados dentro del aprendizaje supervisado se encuentran:

KNN vecinos más cercanos.

Regresión logística.

Regresión por mínimos cuadrados.

Árboles de decisiones.

SVM.

2.7. Aprendizaje No supervisado

El aprendizaje no supervisado, se realiza cuando no tenemos data etiquetada ni clasificada. En consecuencia, toma un sentido de exploración para poder describir la estructura de los datos, y así pueda encontrar una correlación entre ellos y segmentarlos en grupos de datos similares (Clusters), este tipo de aprendizaje se está utilizando para el desarrollo del Deep Learning, y en un futuro no muy lejano relegara al Machine Learning, (Sansano, 2017). Los algoritmos más habituales en este tipo de aprendizaje son: KMeans, Análisis de componentes principales (PCA). Para el presente trabajo de grado, se utilizará para predecir las diferentes fallas del rodamiento del motor de inducción, las técnicas que comprenden el “Aprendizaje Supervisado de Clasificación” (Sansano, 2017).

2.8. Técnicas utilizadas en Machine Learning

2.8.1.K vecinos más cercanos (KNN)

Es un algoritmo muy conocido y básico, usado tanto para la clasificación como para la regresión en un espacio multidimensional. Puede ser utilizado para datos numéricos, discretos, y categóricos. Pero es un algoritmo que también puede cumplir otra función, como es el caso de tratar con datos incompletos. Tiene una aplicación intensa en el reconocimiento de patrones, la mimería de datos y reconocimiento de intrusos.(He & Wang, 2007).

Es un algoritmo de aprendizaje no paramétrico, esto significa que no hace suposiciones explícitas sobre la forma funcional de los datos, no halla la varianza,

el promedio etc., evitando modelar mal la distribución subyacente de los datos lo cual es una desventaja, comparado con otros modelos. Es también un algoritmo que está basado en la instancia, esto significa que memoriza las instancias de formación que posteriormente se utilizan como “conocimiento” para la fase de predicción.

La idea de KNN es que un valor puntual puede ser calculado mediante valores en puntos que son próximos a él, basados en otras variables. Aunque ahora se esté nombrando para describir la imputación de valores, también es necesario explicar su funcionamiento como clasificador y algoritmo de regresión. Este algoritmo además es usado en industrias para la predicción de detección de fallos, aunque los datos no estén etiquetados.

Con respecto al algoritmo de clasificación, si se desea clasificar Xx , si se tiene $x_1; \dots; x_k$ los k vecinos más próximos a Xx se han de regresar la expresión matemática se expresa en la Ec.(7):(He & Wang, 2007).

$$\hat{f}(x) \leftarrow \underset{v \in V}{\operatorname{argmax}} \sum_{i=1}^k \delta(v, f(x_i)) \quad \text{Ec. (7)}$$

Donde $\delta(a, b) = 1$, si $a=b$ y 0 en otros casos.

k es un número de vecinos (k). Es un hiper parámetro que se debe elegir en el momento de la construcción del modelo; si $K = 1$ se crea una partición de Voronoi. K no de ser múltiplo del número de clases, se recomienda elegir un numero impar si el número de clases es par, el valor que se obtiene es un estimado del valor de la función del punto que se desea clasificar, que adquiere el valor de sus vecinos más cercanos. En la Figura18 se muestra como el algoritmo segmenta los patrones. Se elige este algoritmo para este trabajo porque es uno de los más simples para el modelamiento.(López & Manuel, 2019)(Raschka & Mirjalili, 2017).

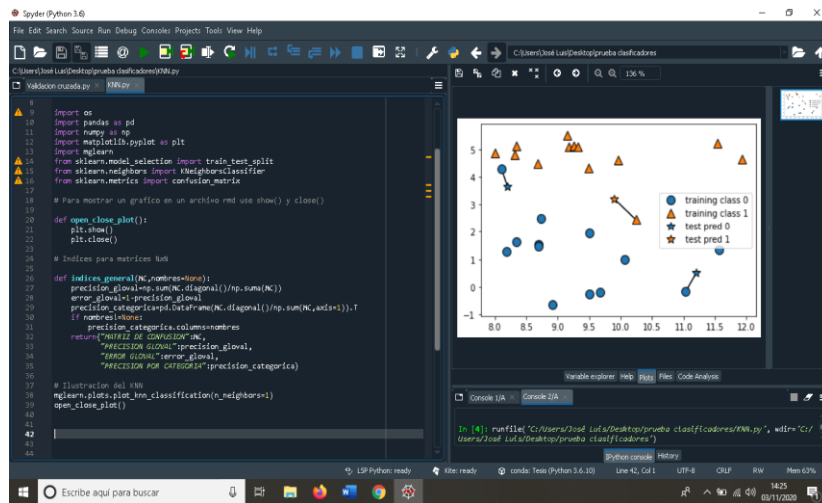


Figura 18. Reconocimiento de patrones (KNN)
Fuente: Elaboración Propia

2.8.2. Árbol de decisión

Los árboles de decisiones se definen mediante una partición continua de la entrada y una definición del modelo por región resultante. Según Murphy, 2012: “El algoritmo de clasificación de árbol de decisiones se puede representar de la siguiente manera” Ec. (10).

$$f(x) = E[y/x] = \sum_{n=1}^M w_n \mathbb{1}(x \in R_n) = \sum_{n=1}^M w_n \phi(x; v_n) \quad \text{Ec. (10)}$$

En donde:

R_n : representa la región n.

w_n : representa la respuesta media en la región.

v_n : realiza una codificación de elección en la variable, para que posteriormente pueda dividir el árbol desde la raíz hasta la hoja n.

Este tipo de algoritmos son versátiles porque pueden realizar tareas de regresión de clasificación, hasta incluso de multi- salida. siendo capaces de ajustar conjuntos complejos de datos (Géron, 2019),(López & Manuel, 2019)

El algoritmo que contempla los árboles de decisión presenta múltiples ventajas con respecto a otros algoritmos de clasificación, esto se debe a su fácil visualización y entendimiento del modelo entrenado resultante, otra ventaja es su invariabilidad con respecto al tamaño de datos y esto se debe a que el proceso por rama se realiza por separado, sumado a eso no es necesario realizar ningún proceso previo de normalización o escalamiento, se puede concluir que los árboles de decisiones trabajan muy bien con datos muy distintos entre sí, o combinaciones de características binarias y continuas (Müller & Guido, 2016),(López & Manuel, 2019).

Poda de Árboles de decisión. – Esto sucede cuando un árbol de decisión presenta Overfitting, esto nos indica que presenta muchas ramas y por consecuencia se torna compleja su interpretación y manejo, por lo tanto, se hace necesaria la poda del árbol, esto consiste en cortar de una forma sucesiva las ramas y nodos hasta conseguir un tamaño adecuado. Se presenta dos tipos de poda:

Prepoda. - Esto consiste en no se generar más ramas o nodos a partir de la condición. Una de las maneras para conseguir este cometido es comparar el error de clasificación de una rama ya extendida con una que aún no se ha expandido, si este es menor que el error de la rama expandida entonces permanece con el nodo sin expandir (poda).

Pospoda. - Esto se ejecuta cuando ya se generó todo el árbol de decisión. Una de las formas para ejecutar la poda es, desarrollar todo el árbol de decisión para posteriormente poder analizar con cuidado que ramas pueden eliminarse para reducir el error al momento de la clasificación.

Uno de los principales inconvenientes que presentan este tipo de modelo son: tienden a adaptarse de una forma excesiva y presenta un rendimiento de generalización deficiente.

Características:(Medrano Sanz, 2019)

Puede modelar interacciones;

Puede manejar naturalmente tanto la regresión como la clasificación (multiclase);

Puede manejar de forma natural tanto variables predictoras continuas como categóricas;

Puede manejar valores perdidos en las variables predictoras;

son robustos a valores atípicos en las variables predictoras;

son insensibles a las transformaciones monótonas de las variables predictoras;

Pueden escalar bien para tamaños de muestra grandes;

Pueden manejar bien las variables predictoras irrelevantes.

No son buenos para capturar relaciones que involucran combinaciones lineales de predictoras variables;

Son inestables en el sentido de que, si los datos se alteran ligeramente, el árbol puede cambiar sustancialmente;

No son tan precisos como algunos de los métodos desarrollados más recientemente.

Ni las máquinas vectoriales de soporte ni las redes neuronales tienen una alta calificación en ninguno de las características positivas mencionadas.



Figura 19. Partes de un árbol de decisión
Fuente: Elaboración propia

2.8.3. Random Forest

Este algoritmo fue desarrollado por Leo Breiman estadístico estadounidense distinguido en la Universidad de California, Berkeley. Fue miembro de la Academia Nacional de Ciencias de los Estados Unidos, conjuntamente con Adele Cutler. Este término aparece por primera vez en la propuesta de Tin Kam Ho de Bell Labs en 1995 (Breinan & Cutler, n.d.). En el modelo se combinan los árboles decisiones, quiere decir que se trabaja con un conjunto de árboles de decisiones no correlacionados y los promedia, cada árbol depende de los valores aleatorios de una muestra, de tal manera que es independiente a una misma distribución en todos los árboles que conforman el bosque. Al realizar una selección aleatoria de características para dividir las ramas o nodos produce una tasa de error que se comparan favorablemente con al algoritmo de AdaBoost, a su favor tienen que este tipo de algoritmos presentan una gran robustez con respecto al ruido. En sucesivas cuentas el método Random Forest se basa en tomar una muestra inicial al árbol de decisión y someterla a una prueba o testeo binario por cada nodo, a esto se le denomina Split, con la finalidad de encontrar la respuesta en el ultima rama. Esta técnica nos permite dar solución a problemas complejos, porque lo divide en pequeños problemas más simples, cuando llega a este punto el algoritmo optimiza

los parámetros de la función Split utilizando las muestras de entrenamiento., su ecuación Ec. (12) representativa es la siguiente:

$$\theta_k = \operatorname{argmax}_{\theta_j \in \tau_j} I_j \quad \text{Ec (12)}$$

La función de ganancia se representa en la siguiente ecuación Ec. (13)

$$I_j = H_{(j)} - \sum_{i \in \{1,2\}} \frac{|S_j^i|}{|S_j|} H_{(S_j^i)} \quad \text{Ec. (13)}$$

s representa un conjunto de muestras que presenta al nodo a dividir, y s' representa dos conjuntos que se originan de la escisión. Dicha función puede medir la entropía del conjunto, a su vez es dependiente del tipo de problema que se aborda.

Cabe mencionar que existen dos métodos para poder estimar la importancia de las variables que contiene la muestra y estas son: La Disminución de Presión Media (MDA) y Disminución Media de Gini (MDG). Este proceso se puede observar en la Figura 21. Para el clasificador Random Forest, varios autores como (Genuer, Poggi, Tuleau-Malot, & Villa-Vialaneix, 2017) y (Janitza, Tutz, & Boulesteix, 2014) confirman que el clasificador Random Forest es una herramienta efectiva en los procesos de predicción con datos no estructurados como es el caso de los recolectados por redes de sensores.

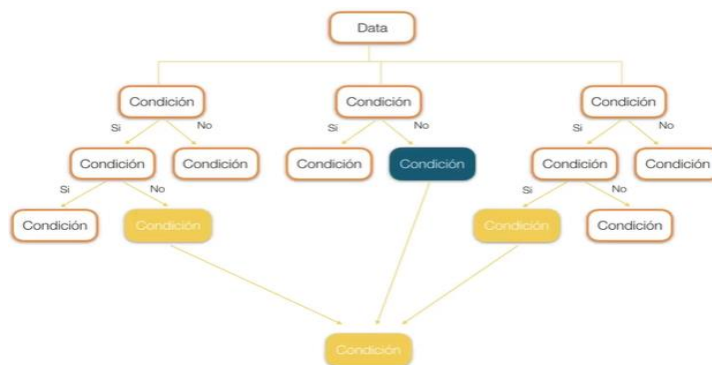


Figura 21. Árboles Aleatorios
Fuente: Elaboración propia

2.8.4. Máquina de vector de soporte (SVM)

Uno de las técnicas de clasificación de datos más usadas son las máquinas de vectores de soporte o en sus siglas en ingles support vector machines.

Este tipo de algoritmo realiza una clasificación sobre vectores que fueron transformados de un espacio inicial a un espacio superior, donde las muestras son separadas por un hiperplano ubicado en el espacio transformado. Se visualiza en la Figura 22.

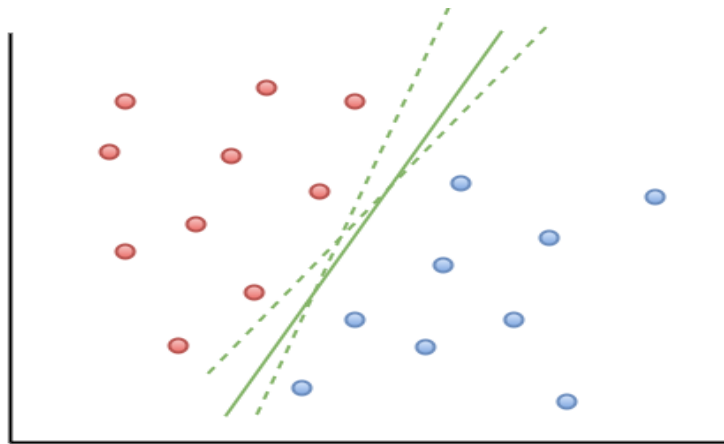


Figura 22. Hiperplanos de separación (SVM)
Fuente: Elaboración propia

Como se puede observar en la Figura 22, la SVM tiene como finalidad maximizar los márgenes mediante hiperplanos entre los grupos de datos a entrenar. La maximización del margen y minimización de la norma euclidiana son similares.

Cuando se presente un nuevo ensayo para su clasificación, el clasificador situara las muestras en su zona pertinente separada previamente por el hiperplano. Esta técnica es una de las más modernas inclusive más eficiente que las técnicas de regresión. Uno de sus beneficios es minimizar el costo computacional. El tiempo de entrenamiento de las redes neuronales y de regresión se realiza de forma más rápida. El algoritmo suele comportarse muy bien cuando la cantidad de datos es muy alta. (Del Pozo Gallego, 2016)(Zhang et al., 2020)

2.8.5.Regularización

En las técnicas de Machine Learning, especialmente en el aprendizaje supervisado, se enfrenta con un problema recurrente, que es el sobreajuste o también conocido como Overfitting, que esta detallado en el apartado 2.6, para solucionar este problema se utiliza la Regularización,

Este tipo de algoritmo, consiste en añadir una penalización a la función de coste, esta penalización genera modelos más simples que generalizan mejor. Cuando se usa la regularización se minimiza la complejidad del modelo a la vez que se minimiza la función de coste. Los modelos que son muy complejos tienden a sobre ajustar, es decir, tienden a encontrar una solución que funcionan muy bien para los datos de entrenamiento, pero muy mal para los datos nuevos. A continuación, desarrollaremos un ejemplo simple donde observaremos el sobreajuste y como la regularización puede ayudarnos a conseguir un mejor modelo.

En el ejemplo de la Figura 23. Se observa el modelo de regresión lineal de un color morado donde su función Hipótesis tiene la forma de: $h_0 = \theta_0 + \theta_1 x$, y se puede observar que no tiene buenas predicciones, el problema que sucede se llama bajo ajuste y esto se debe a la simplicidad de la función.

En el modelo de regresión polinómica de su función hipótesis de grado seis, se observa que la curva descrita de color verde, tiene un ajuste muy bueno con los datos del entrenamiento, sin embargo, no son muy buenas en las predicciones, la curva es bastante ondulada. Sin embargo, tomando una función de segundo grado descrita por una curvatura de color rojo menos compleja que la de sexto grado, observamos que las predicciones mejoran.

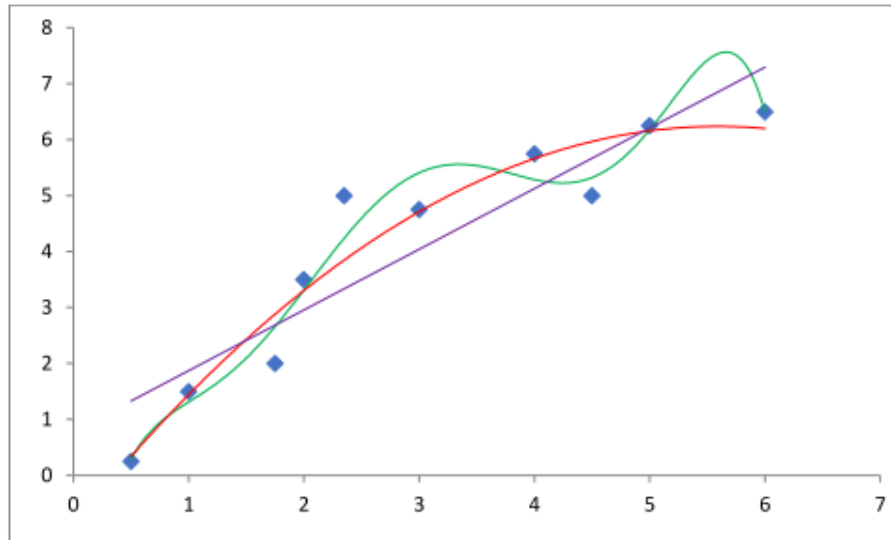


Figura 23. Sobreajuste, bajo ajuste, ajuste correcto
Fuente: Fuente: Elaboración propia

Existen 3 tipos de Regularización las cuales explicaremos a continuación:

Regularización Ridge (L2). - Al usar esta regularización añadiremos un término que penaliza la complejidad de la función de costes. Por ejemplo, lo aplicaremos a una regresión lineal:

$$J(\phi) = \frac{1}{2m} \sum_{i=1}^m (h_{\phi}(x_i) - y_i)^2 + \lambda \sum_{j=1}^n \phi_j^2 \quad \text{Ec. (14)}$$

En donde:

$\lambda \sum_{j=1}^n \theta_j^2$: es un parámetro de regularización Ridge. Lo que hace este parámetro es penalizar los términos de la función hipótesis que regulan las variables de mayor grado.

λ : es un parámetro que tiene como función la regularización. A mayor valor de lambda más simple será la función hipótesis. Si se simplifica demasiado la función hipótesis, y el valor de parámetro lambda aumenta considerablemente, este efecto provocaría un problema de ajuste bajo o underfitting. El parámetro lambda será clave a la hora de entrenar el modelo predictivo de clasificación.

Regularización Lasso (L1). - En este caso algunos parámetros de Phi de la función de costes se anularán, realizándose una selección de parámetros, esto es cuanto mayor sea el parámetro de regularización lambda, también será mayor los parámetros Phi que se anularan. La ecuación matemática de la función de costes cuando se utiliza una regresión lineal con regularización de Lasso es:

$$J(\phi) = \frac{1}{2m} \sum_{i=1}^m (h_{\phi}(x_i) - y_i)^2 + \lambda \sum_{j=1}^n |\phi_j| \quad \text{Ec. (15)}$$

En donde:

$\lambda \sum_{j=1}^n |\theta_j|$: es un parámetro de regularización Lasso. Lo que hace este parámetro es penalizar los términos de la función hipótesis que regulan las variables de mayor grado.

λ : es un parámetro que tiene como función la regularización. A mayor valor de lambda más simple será la función hipótesis. Si se simplifica demasiado la función hipótesis, y el valor de parámetro lambda aumenta considerablemente, este efecto provocaría un problema de ajuste bajo o underfitting. El parámetro lambda será clave a la hora de entrenar el modelo predictivo de clasificación.

Regularización Elastic Net. - Para poder complementar las técnicas de regularización de Ridge y Lasso, se desarrolló una técnica más eficiente denominada de regularización Elastic net. La técnica elimina el problema de estabilidad que presenta la técnica Lasso. El número de parámetros Phi que se ponen a cero es menor comparándose con la técnica Lasso. Como se observa, la ecuación matemática de la función de costes que utiliza la técnica Elastic net, para este caso se observa cómo se combina las técnicas de Ridge y Lasso(Del Pozo Gallego, 2016), la cual se presenta en la Ec.16

$$J(\phi) = \frac{1}{2m} \sum_{i=1}^m (h_{\phi}(x_i) - y_i)^2 + \lambda \sum_{j=1}^n |\phi_j| + \lambda_2 \sum_{j=1}^n \phi_j^2 \quad \text{Ec. (16)}$$

2.9. Métricas de evaluación de los algoritmos de clasificación

2.9.1. Matriz de Confusión

Este tipo de evaluación es una de las formas más sencillas para poder evaluar, la exactitud, la precisión del modelo a entrenar, este tipo de métricas se utiliza en problemas de clasificación donde hay salidas de dos o más tipos. por ejemplo, si queremos predecir si una persona tiene o no pulmonía. Lo que se hace es primeramente dar una etiqueta a nuestra variable objetivo. “1” cuando la persona tiene pulmonía y “0” cuando no lo tiene, como se muestra en la siguiente Tabla 6.

Tabla 6. *Matriz de Confusión*

		Predicción	
		Positivo	Negativo
Valor real	Positivo	Verdaderos positivos	Verdaderos negativos
	Negativo	Falsos positivos	Falsos negativos

Fuente: Elaboración propia

Donde:

Verdaderos positivos. – Esto sucede cuando el modelo predice como positivo una premisa que es positivo.

Falsos positivos. - Esto sucede cuando el modelo predice como positivo una premisa que es negativo.

Verdaderos negativos. - Esto sucede cuando el modelo predice como negativo una premisa que es negativo.

Falsos negativos. - Esto sucede cuando el modelo predice como negativo una premisa que es positivo.

2.9.2. Curva ROC y Área bajo la Curva (AUC)

La curva ROC (características de funcionamiento del receptor) es una de las evaluaciones importantes para verificar la eficiencia de cualquier modelo de clasificación.

La curva ROC descrita en la Figura 24 nos indica que tan bueno puede distinguir el modelo entre dos cosas. Los modelos más eficientes pueden distinguir con mayor precisión entre dos variables, mientras que un modelo ineficiente tendrá problemas para distinguir entre dos clases de variables. Esta técnica es conveniente por lo siguiente:

Es invariable con respecto a la escala, mide que tan eficientes es la clasificación de patrones al momento de realizar las predicciones en lugar de sus valores absolutos.

Es invariable con respecto al umbral de clasificación de patrones, realiza una medición de la calidad de las predicciones del modelo sin tener en cuenta que umbral de clasificación se pueda elegir.

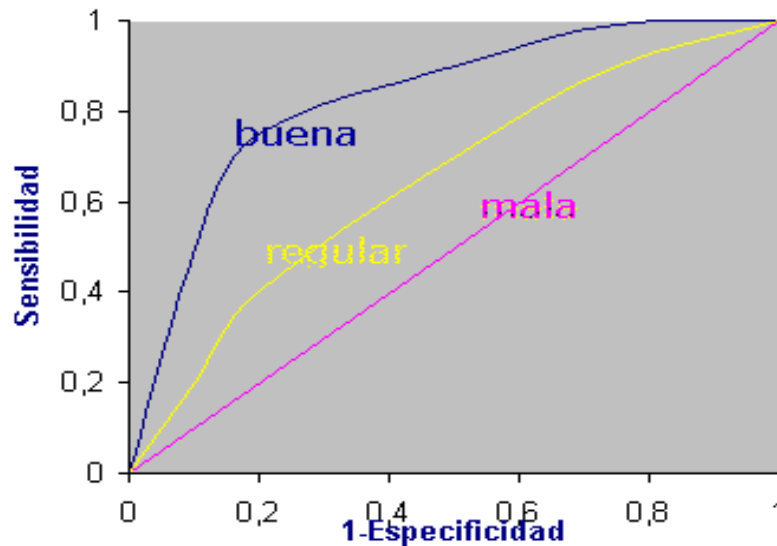


Figura 24. Tipos de curva ROC
Fuente: Elaboración propia

2.9.3.F1 Score

Esta medida de precisión se utiliza para poder evitar problemas de asimetría que pueda suceder. Como, por ejemplo, se puede tener ochenta ensayos con un rodamiento sano y nueve ensayos con rodamientos con falla. En consecuencia, el algoritmo clasificador estará prediciendo de una forma constante que el rodamiento está sano y más aun dándonos un porcentaje de exactitud del 99%, sin embargo, está claro que hay un error predatorio del algoritmo clasificador. Para evitar este problema de asimetría se utiliza el F1 score (Del Pozo Gallego, 2018).

Para determinar el F1 score se utilizará la Tabla 2 que describe la Matriz de confusión previo a esto definamos los siguientes conceptos para llegar a la fórmula de F1 score.

Precisión (P). – Nos indica la relación de verdaderos positivos entre los clasificados como positivos.

$$Precision = \frac{\text{verdaderos positivos}}{\text{verdaderos positivos} + \text{falsos positivos}} \quad \text{Ec. (17)}$$

Recall (R). - Nos indica la precisión para determinar los casos positivos de los casos realmente positivos. Este parámetro se le llama también sensibilidad o relación de verdaderos positivos.

$$Recall = \frac{\text{verdaderos positivos}}{\text{verdaderos positivos} + \text{falsos positivos}} \quad \text{Ec. (18)}$$

Por lo tanto:

F1 score. – Es una métrica que combina eficientemente el parámetro Precisión y el parámetro Recall para poder determinar el grado de importancia y eficacia del clasificador.

$$F1 \text{ score} = 2 \frac{PR}{P + R} \quad \text{Ec. (19)}$$

CAPÍTULO 3

DESARROLLO DE LA SOLUCIÓN

3.1. Introducción a la solución

En este apartado hablaremos del sistema de análisis de falla que envuelve a la máquina, en este caso al motor de inducción se describirá la secuencia que se sigue para su configuración. El sistema es una red de dispositivos físicos y computadoras que permite recopilar y compartir datos y se almacena en plataformas como por ejemplo la nube, esto ayuda a planificar las actividades de mantenimiento del motor de inducción. En nuestro caso, utilizaremos como base de datos el banco de pruebas de la Case Western Reserve University (CWRU), que comprende de un motor de inducción (sus parámetros en el Anexo 2), un elemento de medición que son los acelerómetros o sensores que transforman las vibraciones en señales eléctricas o analógicas (tipos de sensores recomendados en el Anexo 3), un sistema de adquisición de datos que realiza un acondicionamiento y procesamiento de señales analógicas a señales discretas; y final mente un programa para su entrenamiento atreves de los algoritmos de aprendizaje automatizado o Machine Learning, y este último proceso es el tema de solución que se estudia en esta tesis de grado. El diagrama de bloques se muestra en la figura 25.

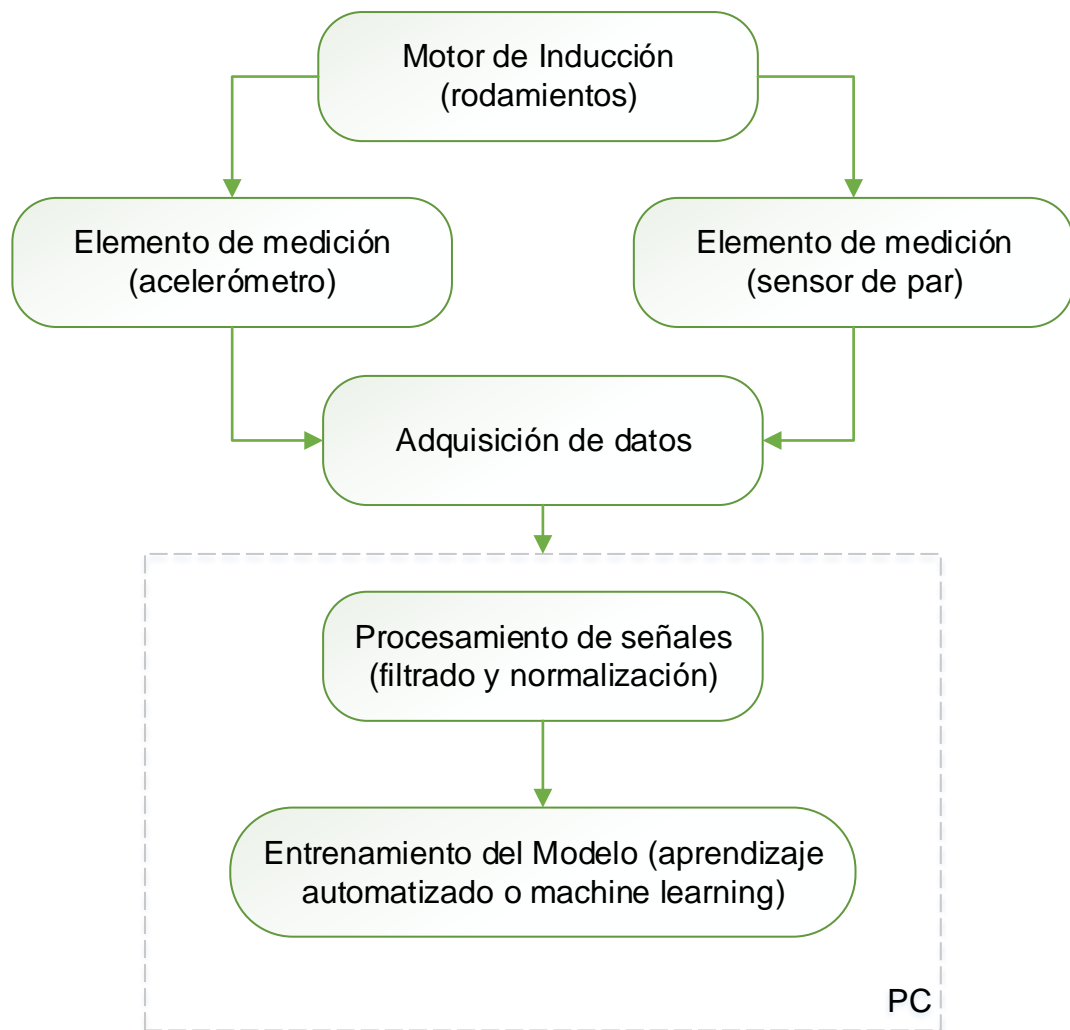


Figura 25. Diagrama de bloques del sistema de falla
Fuente: Elaboración propia

3.2. Adquisición de Datos

A continuación, se requiere un benchmark o punto de referencia para aplicar la teoría desarrollada de clasificación de características y patrones en el contexto de Machine Learning para la aplicación de estos métodos, en este caso en particular se tomará muestras de fallo del Centro de Datos de Rodamientos de la Case Western & University Reserve (CWRU), Las muestras de vibración se recogieron utilizando una grabadora DAT. de 16 canales y se procesaron posteriormente en un entorno Matlab. Todos los archivos de muestras están en formato Matlab (*.mat). Los datos digitales se recopilaban a 12.000 muestras por segundo y se usará para validar los modelos predictivos de Machine Learning.

3.2.1. Banco de Ensayos

Como se muestra en la figura 27, el banco de ensayos de Case Western Reserve University (CWRU), consta de un motor eléctrico de inducción de 2 HP, un sensor de par que registra el momento alrededor del eje en todo instante durante el funcionamiento, un acelerómetro de base magnética colocado en la base del rodamiento el cual recopilara las muestras, mientras que un dinamómetro está acoplado a la derecha encargado de variar la carga del motor.

El procedimiento que realizó la CWRU fue lo siguiente: introdujo fallas de un solo punto en los rodamientos de prueba utilizando mecanizado de electro descarga con diámetros de falla de 7 milésimas de pulgada, 14 milésimas de pulgada, 21 milésimas de pulgada, tanto en la pista interior y exterior del elemento rodante del rodamiento, Para este caso se recopilaron 2700 muestras, cada muestra tomada corresponde a un estado del rodamiento del motor de inducción, para una carga de 1 y 2 HP, con una velocidades de 1720 RPM y con una frecuencia de muestreo de 12 kHz, ver detalles en la tabla 8, las muestras se distribuyen de la siguiente manera:

300 ensayos que se realizaron al rodamiento sano;

800 ensayos que se realizaron al rodamiento con fallo leve;

800 ensayos que se realizaron al rodamiento con fallo moderado;

800 ensayos que se realizaron al rodamiento con fallo grave;

Los detalles de los ensayos en el Anexo 4. Posteriormente en la figura 26 se puede observar la PC portátil, modelo Ideapad 310 de Lenovo, que dispone de un procesador Intel Core i5 con 8 GB de memoria RAM. El sistema operativo es Windows 10 Professional de 64 bits. Esta se utilizará para el tratamiento de señales y entrenamiento de los modelos predictivos.

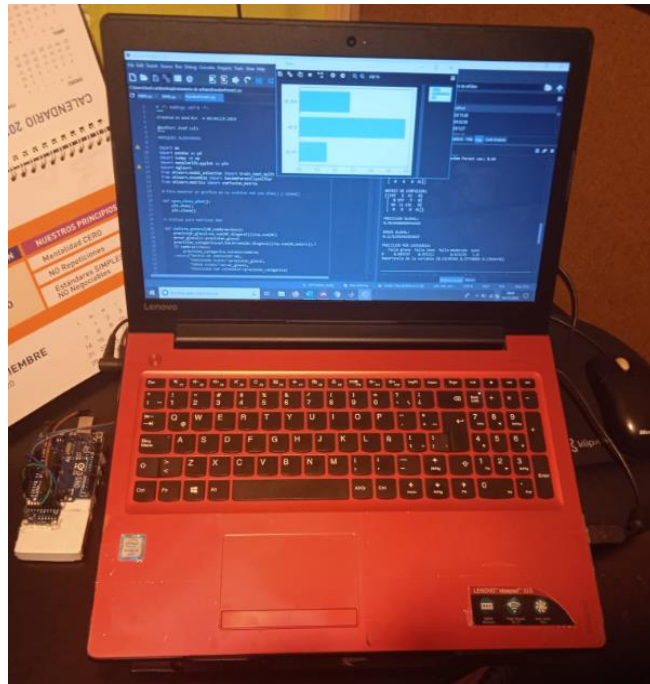
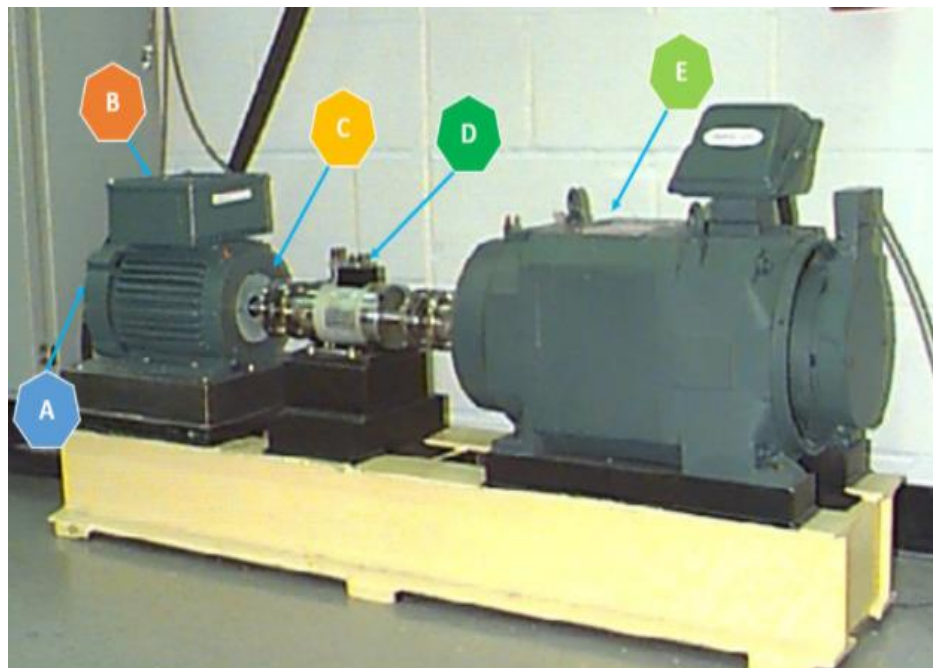


Figura 26. Ordenador Portátil
Fuente: Elaboración propia



Anillo de prueba de rodamiento. A = cojinete del extremo del ventilador; B = motor electrónico; C = cojinete del extremo de transmisión; D = transductor de par; E = dinamómetro.

Figura 27. Banco de ensayos
Fuente: Centro de datos Reserve University (CWRU), n.d.

Para comenzar a modelar los algoritmos de predicción, se usarán señales de tipo Drive End Bearing (Rodamiento extremo del eje), que se muestrea con una frecuencia de 12kHz, estas son las señales que corresponden a los rodamientos ubicados a los extremos del eje, que desde ahora lo mencionaremos con sus siglas “_DE”. En la tabla 7 se presentan las características de los rodamientos a trabajar.

Tabla 7. Características del rodamiento

Modelo	Drive End Bearing 6205-2RS JEM SKF				
Dimensiones del rodamiento (mm)	Diámetro interior	Diámetro exterior	Espesor	Diámetro de bola	Diámetro de pista
	25,0012	51,9988	15,0012	7,9400	39,0398

Fuente: Elaboración propia

En la tabla 8, se expone los detalles de los datos con extensión(.mat), que se trabajaran en los tres modelos de predicción a utilizar en el presente trabajo de fin de grado.

Tabla 8. Data de fallo de rodamiento

Diámetro de falla(mm)	Grado de falla	Carga del motor(HP)	Aprox. Velocidad del motor(rpm)	Par de fuerza(N.m)	Datos de la Pista interior(_PIT)	Datos de la Bola del rodamiento (_B)	Datos de la posición de Pista exterior relativa a la zona de carga(zona de carga ortogonal a las 3:00)(_PEX)
0,007"	leve	1	1772	4.02	IR007_1(279.mat)	B007_1(283.mat)	OR007A3_1(299.mat)
0,014"	moderado	1	1772	4.02	IR014_1(275.mat)	B0014_1(287.mat)	OR0014A3_1(309.mat)
0,021"	grave	1	1772	4.02	IR0021_1(271.mat)	B0021_1(291.mat)	OR0021A3_1(316.mat)

Fuente: Elaboración propia

3.3. Preparación de los datos

La preparación de los datos antes del entrenamiento es un procedimiento muy importante para alcanzar la mayor eficiencia en la captura de patrones y conseguir el objetivo que es la predicción de fallas del rodamiento que concierne a este estudio, esto consiste en realizar un tratamiento a la señal entregada por el acelerómetro (sensor ubicado en la placa del motor de inducción), teniendo la señal en función del tiempo se le realiza la transformada rápida de Fourier para transformar la señal del dominio del tiempo al dominio de la frecuencia. Al obtener los datos de las amplitudes de los armónicos se procede a filtrar con un filtro pasa bajo para poder dejar más limpia la señal, una vez obtenido los datos que se deben analizar se procederá a normalizar la señal, para el tratamiento de la señal se utilizará el software Matlab R2016a, la cual se contempla en la figura 28 y 29.

La normalización de datos consiste en aplicar una serie de reglas a las relaciones obtenidas tras el paso del modelo entidad-relación al modelo-relacional, este método tiene como objetivo minimizar la redundancia de datos. Para normalizar los datos y ponerlos todos dentro de un rango se utilizará el siguiente método matemático representada en la ecuación Ec (20).

$$X = \frac{X - \mu}{S} \quad \text{Ec. (20)}$$

Donde:

X = nos indica el valor a normalizar.

μ = es la media estadística valores a normalizar.

S = X max – X min, es la diferencia de los valores a normalizar.

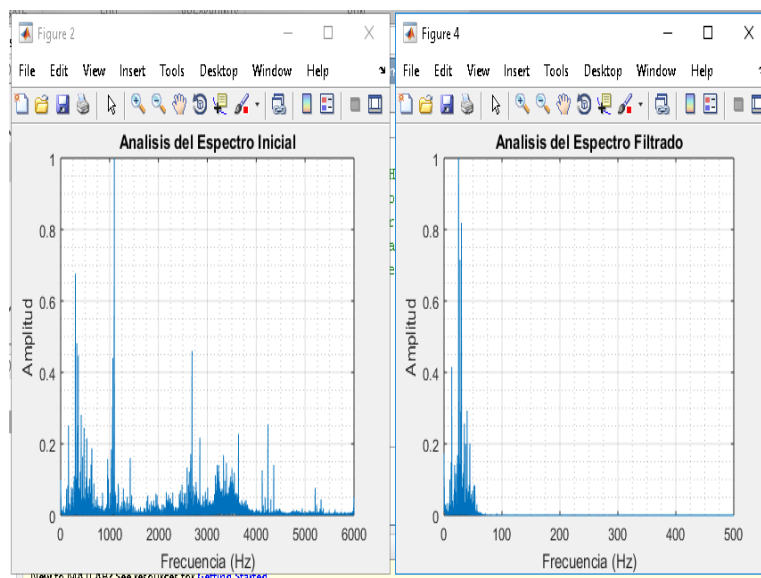


Figura 28. Señal tipo Drive End Bearing
Fuente: Elaboración propia

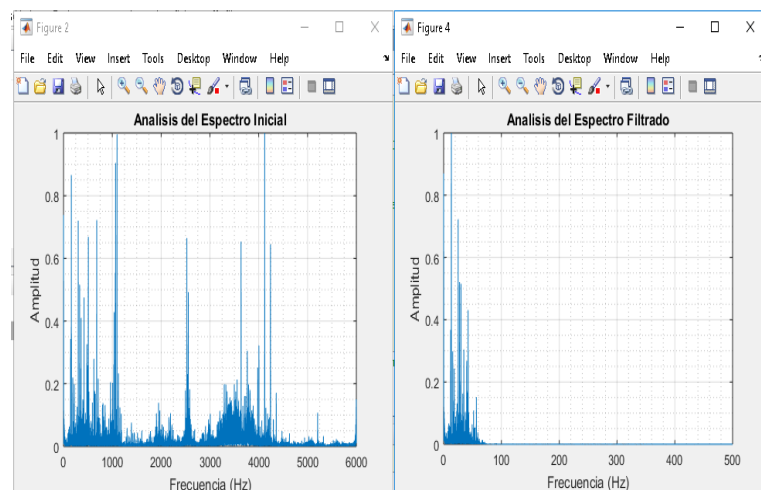


Figura 29. Señal tipo Fan End Bearing
Fuente: Elaboración propia

3.4. Selección del lenguaje de programación

Para el desarrollo del modelo predictivo es necesario una serie de recursos para obtener los resultados esperados, en el presente trabajo se utilizará el siguiente lenguaje de programación: Lenguaje de programación Python. - Se eligió este lenguaje de programación por ser muy flexible y tener un entorno muy amigable para el desarrollo de la programación, entre las ventajas más interesante tenemos:

Python es un lenguaje de propósito general enfocado a la reutilización de código.

Python es un lenguaje orientado a objetos que permiten una fácil integración a los sistemas de una organización.

Python es considerado un lenguaje muy expresivo, esto quiere decir que sus programas son muy compactos y bastante corto en su código a comparación de otros lenguajes.

Python es muy legible y elegante en su sintaxis ofrece un entorno interactivo que facilita la realización de pruebas y es considerado un lenguaje de muy alto nivel.

Se utilizará en este trabajo de tesis la versión 3.6 de Python. La ventaja de Python es que es un software libre que se puede descargar de su página oficial o a través de Anaconda.

Librerías de Python para Machine Learning. - Las librerías que se utiliza para implementar algoritmos de machine Learning son las siguientes:

Scikit-learn. Es un paquete que contiene los principales métodos de Data Mining y Machine Learning.

NumPy. Es un paquete fundamental para el cálculo científico, donde se ve el uso de matrices, algebra lineal, transformada de Fourier, entre otros.

SciPy. Es un paquete especializado para calculo científico avanzados como optimización de funciones, distribuciones de probabilidades, entre otros.

Matplotlib. Es un paquete especializado en la realización de gráficos.

Ipython. Proporciona la conexión con el intérprete del lenguaje Python estándar. Es la consola de Python.

Pandas. Es un paquete para el manejo de análisis de datos, implementa el concepto de DataFrame.

En la figura 30 se puede observar las librerías instaladas para este trabajo.


```

Anaconda Prompt (anaconda) - python
(tesis) C:\Users\José Luis>python
Python 3.8.5 (default, Sep 3 2020, 21:29:08) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import scipy
>>> scipy.__version__
'1.5.0'
>>> import numpy
>>> numpy.__version__
'1.19.1'
>>> import matplotlib
>>> matplotlib.__version__
'3.3.1'
>>> import pandas
>>> pandas.__version__
'1.1.3'
>>> import sklearn
>>> sklearn.__version__
'0.23.2'
>>> Para esta Tesis se instalaron estas librerías para poder trabajar con Machine Learning en Python

```

Figura 30. Librerías de Python
Fuente: Elaboración propia)

3.5. Método de entrenamiento

Cuando nos referimos a un método de entrenamiento hablamos de un modo de segmentar los grupos de entrenamiento (Train) y evaluación (Test), al generar el algoritmo generalmente se segmenta en forma aleatoria en una proporción que le podemos indicar, como por ejemplo 70% de Train y 30% de Test, esta dinámica se puede visualizar en la figura 31.

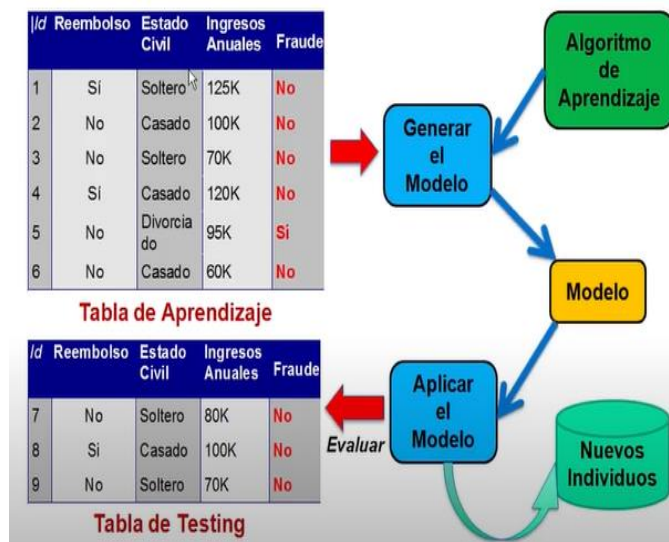


Figura 31. Método de entrenamiento
Fuente: Promidad.com

3.6. Entrenamiento y Simulación

En este apartado describiremos el proceso que se desarrolló para obtener el modelo de aprendizaje supervisado de clasificación para predecir los fallos de los rodamientos del motor de inducción.

Para clasificar y entrenar se utilizaron varios enfoques y técnicas, primeramente, se utilizó para modelar el modelo la técnica de K vecinos más cercanos (KNN) que es una técnica básica fácil de entender y funciona bien cuando la muestra están bien etiquetados que es nuestro caso en particular, tiene buenos resultados para este tipo de clasificación. Luego se utilizará la técnica SVM (Soporte de maquina vectorial) que tiene como ventajas realizar con mayor precisión y rapidez en comparación de otros algoritmos como el de árboles de decisiones las predicciones y finalmente la técnica de Random Forest, que es muy poderosa para este tipo de muestra de datos. Se analizará el resultado de la clasificación de todos ellos posteriormente se realizará una comparativa para determinar cuál es el mejor para esta tarea de clasificación. Se estudiará los rodamientos en la `posición del extremo de la transmisión o de eje (Drive End Bearing).

3.7. Entrenando el modelo para el caso del rodamiento extremo del eje (_DE)

Lo primero que realizaremos es definir el porcentaje de datos de entrenamiento que se utilizara para entrenar el modelo, en esta ocasión tomaremos un 70% para el entrenamiento o aprendizaje (Train) y un 30% de testing tomados de 300 muestras de rodaje sano, 800 muestras de rodamiento con fallo leve(0.07mm), 800 muestras de rodamiento con fallo moderado(0.14mm) y 800 muestras de rodamiento con fallo grave (0.21mm).

3.7.1. Entrenamiento con el modelo KNN

Para este entrenamiento realizaremos tres pasos, primero definamos el elemento nuevo a clasificar: vector $X = (x_1; x_2; \dots x_n)$

Paso 1: Primeramente, definiremos las características que tiene el paquete de muestra creado, se encuentran en la figura 32, luego se identifica la K muestras más cercanas a X, independiente de su clase. es decir, se hallará la distancia (se

define una métrica, para este caso usaremos la Distancia Euclidiana) de un z a un X_n .

Paso 2: Entre los K vecinos se contará los elementos que pertenecen a cada clase. es decir se tomara las menores K distancias.

Paso 3: Se asigna X a cada clase más común entre los vecinos más cercanos, El parámetro a considerar para este entrenamiento es el K que nos indica el número de vecinos más cercanos a la variable a predecir, no debe de ser múltiplo del número de clases y se recomienda un número impar para evitar las votaciones iguales. Se deben de tomar las siguientes consideraciones, no existe un método estándar para determinar un valor óptimo de K , valores muy pequeños son sensibles al ruido, los valores grandes son inmunes al ruido, pero corre el riesgo que si hay pocas muestras en una de las categorías estas no sean seleccionadas nunca, utilizaremos el conocido thumb rules para la selección de K para este trabajo y nos indica lo siguiente : se determina el K como numero impar más cercano a la raíz cuadrada de " n " donde " n " es el número total de datos del número total de la clase más pequeña. Para nuestro estudio la clase más pequeña es la de los rodamientos sanos que tienen 300 muestras que sacándole la raíz cuadrada nos da 17.3 u. y por lo tanto se tomaría valores $K = 3, 5, 7, 9, 11, 13, 15$.

Una vez entrenado el modelo con cada valor k , se obtuvo el valor más óptimo de predicción con el $K=7$. La simulación se presenta en el anexo 1.

```

In [19]: datos=pd.read_csv('Train_DE.csv',delimiter=',',decimal=".")
In [20]: print(datos.shape)
(2700, 4)

In [21]: print(datos.head())
   DE_PIT  DE_B  DE_PEX CLASE
0  0.699187  0.0    0.0  sano
1  0.031075  0.0    0.0  sano
2  0.048198  0.0    0.0  sano
3  0.040059  0.0    0.0  sano
4  0.027383  0.0    0.0  sano

In [22]: print(datos.info())
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2700 entries, 0 to 2699
Data columns (total 4 columns):
#   Column  Non-Null Count  Dtype
---  -
0   DE_PIT  2700 non-null    float64
1   DE_B    2700 non-null    float64
2   DE_PEX  2700 non-null    float64
3   CLASE   2700 non-null    object
dtypes: float64(3), object(1)
memory usage: 84.5+ KB

```

Figura 32. Características de las muestras para entrenar KNN
Fuente: Elaboración propia

En la tabla 9 se presenta la matriz de confusión realizada al modelo para determinar si el aprendizaje fue óptimo,

Tabla 9. Matriz de Confusión de KNN_DE

KNN		PREDICCIÓN			
		Fallo grave	Fallo leve	Fallo moderado	Sano
VALOR REAL	Fallo grave	198	4	37	5
	Fallo leve	3	232	9	0
	Fallo moderado	42	10	176	4
	Sano	0	0	2	89

Fuente: Elaboración propia

Como podemos observar el promedio global es de un 86% que es un valor aceptable para el modelo, pero para concluir si verdaderamente nos es útil debemos observar los índices de calidad por categoría.

Índices de Calidad del modelo: los índices nos indicaran con mayor certeza que tan bueno y preciso es el modelo a diferencia del cálculo de precisión global que puede ser engañoso.

Tabla 10. *Índices de Calidad del Modelo KNN_DE*

Precisión por Categoría en %				Precisión Global en %
Fallo grave	Fallo leve	Fallo moderado	Sano	KNN
0.81	0.95	0.76	0.98	0.8569

Fuente: Elaboración propia

Como se observa en la tabla 10 las predicciones por clase sano, leve, grave son buenas y aceptables la clase moderado con 76% no es están buena pero aceptable. El modelo de los KNN para este tipo de datos es bueno.

3.7.1.1. Entrenamiento con el modelo Support Vector Machine (SVM)

Como según se ha expuesto en el capítulo dos de este trabajo de fin de grado la clasificación mediante máquinas de vectores soporte es diferente al resto de maneras de clasificar. El objetivo de este modelo donde se utiliza SVM de clasificación, es encontrar un hiperplano, a comparación de otros métodos donde se utiliza la distancia (KNN) o árboles para poder clasificar, que separe a las clases con el mayor margen posible minimizando el error, y para esto utilizaremos su conocido truco de Kernel o núcleo que transforma un espacio de datos de entrada en la forma requerida, ofreciendo una forma más eficiente y menos costos de transformar los datos en dimensiones superiores. Se utilizará los Kernel lineal, polinómica y de base radial (Rbf) o conocido también como gaussiano, y de ellos escogeremos el que tiene mayor eficiencia en la predicción de las clases. Para poder realizar una clasificación mediante el uso de SVM, primeramente, se tiene que realizar, como el resto de los clasificadores, su creación, para esto consideremos las características que tiene el paquete de datos creado, figura 33.

```

Console 1/A x
In [2]: datos=pd.read_csv('Train_DE.csv',delimiter=',',decimal='.')
In [3]: print(datos.shape)
(2700, 4)
In [4]: print(datos.head())
  DE_PIT  DE_B  DE_PEX  CLASE
0  0.699187  0.0    0.0  sano
1  0.031075  0.0    0.0  sano
2  0.048198  0.0    0.0  sano
3  0.040059  0.0    0.0  sano
4  0.027383  0.0    0.0  sano
In [5]: print(datos.info())
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2700 entries, 0 to 2699
Data columns (total 4 columns):
#   Column  Non-Null Count  Dtype
---  ---
0   DE_PIT  2700 non-null    float64
1   DE_B    2700 non-null    float64
2   DE_PEX  2700 non-null    float64
3   CLASE   2700 non-null    object
dtypes: float64(3), object(1)
memory usage: 84.5+ KB
None

```

Figura 33. Características de los datos SVM
Fuente: Elaboración propia

Posterior mente se evaluará el modelo predictivo para los rodamientos con el Kernel Lineal para el Drive End Bearing (DE), los resultados del entrenamiento del modelo se visualizarán en la Matriz de confusión y en el cuadro de Índices de calidad expresadas en las tablas 11 y 12, respectivamente.

Tabla 11. Matriz de Confusión SVM_Lineal_DE

SVM(Lineal)		PREDICCIÓN			
		Fallo grave	Fallo leve	Fallo moderado	Sano
VALOR REAL	Fallo grave	11	0	243	0
	Fallo leve	0	3	247	0
	Fallo moderado	1	3	217	0
	Sano	0	0	86	0

Fuente: Elaboración propia

Tabla 12. Índices de Calidad del Modelo SVM_Lineal_DE

Precisión por Categoría en %				Precisión Global en %
Fallo grave	Fallo leve	Fallo moderado	Sano	SVM (Lineal)
0.0433	0.012	0.981	0	0.2848

Fuente: Elaboración propia

Una vez terminado el entrenamiento del modelo donde se utilizó el Kernel lineal se observa que las predicciones son muy malas tanto en la predicción global como en las categóricas en donde solo llego a predecir con un 98% los de fallo moderado y casi 0 % las demás categorías, por lo tanto, el modelo no funciona y se descarta. Adicionalmente se observa que el modelo reconoce solo una categoría que es el de fallo moderado despreciando a todas las demás categorías

Ahora se tomará para el entrenamiento del modelo predictivo, el Kernel Polinomial, a continuación, se visualizará en la Matriz de confusión y en el cuadro de Índices de calidad expresadas en las tablas 13 y 14, respectivamente los resultados obtenidos.

Tabla 13. Matriz de Confusión SVM_Poly_DE

SVM(Poly)		PREDICCIÓN			
		Fallo grave	Fallo leve	Fallo moderado	Sano
VALOR REAL	Fallo grave	7	234	3	0
	Fallo leve	0	241	3	0
	Fallo moderado	6	217	9	0
	Sano	0	91	0	0

Fuente: Elaboración propia

Tabla 14. Índices de Calidad del Modelo SVM_Poly_DE

Precisión por Categoría en %				Precisión Global en %
Fallo grave	Fallo leve	Fallo moderado	Sano	SVM(Poly)
0.028	0.9877	0.0387	0	0.3168

Fuente: Elaboración propia

Una vez terminado el entrenamiento del modelo en donde se utilizó el Kernel Polinomial, se obtiene unas predicciones muy malas como en el caso anterior que se utilizó el Kernel lineal. Se observa que tanto la precisión global como en el índice de calidad por categoría, las predicciones comparativamente son muy malas también, donde solo llego a predecir con un 98% la clase de fallo leve y casi 0 % las demás clases. Por lo tanto, el modelo no funciona y se descarta.

Como en el caso anterior vemos una peculiaridad en el modelo, esto es que ahora toma a todos los datos como falla leve a diferencia del anterior que tomo a todos como moderado, el modelo no diferencia las diferentes clases

Después de los modelos fallidos tomaremos para el entrenamiento del modelo predictivo, el Kernel RBF, a continuación, se visualizará en la Matriz de confusión y en el cuadro de Índices de calidad expresadas en las tablas 15 y 16, respectivamente los resultados obtenidos.

Tabla 15. Matriz de Confusión SVM_Rbf_DE

SVM (Rbf)		PREDICCIÓN			
		Fallo grave	Fallo leve	Fallo moderado	Sano
VALOR REAL	Fallo grave	7	233	4	0
	Fallo leve	0	241	3	0
	Fallo moderado	5	215	12	0
	Sano	0	90	1	0

Fuente: Elaboración propia

Tabla 16. *Índices de Calidad del Modelo SVM_Rbf_DE*

Precisión por Categoría en %				Precisión Global en %
Fallo grave	Fallo leve	Fallo moderado	Sano	SVM(Rbf)
0.0286	0.9877	0.0517	0	0.3205

Fuente: Elaboración propia

Como en los casos anteriores este también tiene una muy mala predicción, a continuación, se mostrará un comparativo del rendimiento de los Kernel en los tres casos de estudio

Tabla 17. *Cuadro Comparativo*

Comparativo de Kernel %		
Rbf	Lineal	Polinomial
0.3205	0.2848	0.3168

Fuente: Elaboración propia

Como se observa en la tabla comparativa 17, el Kernel Rbf es ligeramente mejor que los demás, pero totalmente ineficiente para las predicciones de los fallos del rodamiento.

3.7.2. Entrenamiento con el modelo Random Forest

Para poder realizar una clasificación utilizando árboles de decisión, se estudiará el universo de datos obtenidos y determinar si se ajusta a este modelo. En este trabajo se utilizará un conjunto de árboles de decisión, que trabaja para realizar la clasificación, de ahí se deriva su nombre Random Forest (RF). El objetivo que se busca con este algoritmo de aprendizaje es optimizar las muestras a través de un bosque de árboles de decisión, al elegir este modelo para clasificar cuatro tipos de fallas, se asocia una característica a cada clase de falla. Los primeros valores, hasta

el 300, será “sano”, los valores del 800 hasta el 1100 tendrán una característica “leve”. Los siguientes 1101 hasta el 1900 tendrán una característica “moderado” y 1901 hasta 2700 tendrán la característica de “grave”.

Posteriormente, se definirá los datos de entrenamiento a utilizar, luego se utilizará las muestras del Bootstrap (conjunto de datos por nodo) para construir los árboles que pertenecerán al bosque aleatorio. Para este caso en particular se elige 40 árboles con el parámetro `n_estimators`, y con ello se realizará el entrenamiento del modelo predictivo referido a los rodamientos del motor de inducción. Luego se determina la profundidad máxima del árbol con el parámetro siguiente (`max_depth=None`) este parámetro nos indica que los nodos se expanden hasta que las hojas sean puras o hasta que todas las hojas contengan el mínimo de muestras necesarias para dividir el nodo interno. Posteriormente se calificará la importancia de las variables o la variable que ayuda a mejorar la predicción del modelo y esto se define a través de criterio de Gini seleccionado en el algoritmo presentado en el anexo 1. Las características que tiene el conjunto de datos creado se encuentran en la figura 34.

```
In [2]: datos=pd.read_csv('Train_DE.csv',delimiter=',',decimal=".")
In [3]: print(datos.shape)
(2700, 4)
In [4]: print(datos.head())
  DE_PIT  DE_B  DE_PEX CLASE
0  0.699187  0.0    0.0 sano
1  0.031075  0.0    0.0 sano
2  0.048198  0.0    0.0 sano
3  0.040059  0.0    0.0 sano
4  0.027383  0.0    0.0 sano
In [5]: print(datos.info())
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2700 entries, 0 to 2699
Data columns (total 4 columns):
#   Column  Non-Null Count  Dtype
---  ---
0   DE_PIT  2700 non-null    float64
1   DE_B    2700 non-null    float64
2   DE_PEX  2700 non-null    float64
3   CLASE   2700 non-null    object
dtypes: float64(3), object(1)
memory usage: 84.5+ KB
None
```

Figura 34. Características de los datos Random Forest
Fuente. Elaboración propia

Una vez creado y entrenado el dataset de árboles de decisión, estos integraran el bosque aleatorio que se utilizará como clasificador de los datos. Inicial mente se prueba con los datos de test de prueba, para poder empezar a determinar la capacidad de clasificación del modelo. A continuación, se presentará las métricas de validación del modelo expresadas en la matriz de confusión, índices de calidad e importancia de la variable y están dadas en la tabla 18,19 y 20 respectivamente.

Donde:

DE_P.PIT = Pista interior del rodamiento del extremo del eje.

DE_P.PEX = Pista exterior del rodamiento del extremo del eje.

DE_B = Elemento rodante o bola del rodamiento del extremo del eje.

Tabla 18. *Matriz de Confusión Random Forest_DE*

Random Forest		PREDICCIÓN			
		Fallo grave	Fallo leve	Fallo moderado	Sano
VALOR REAL	Fallo grave	197	5	42	0
	Fallo leve	0	237	7	0
	Fallo moderado	30	11	191	0
	Sano	0	0	0	91

Fuente: Elaboración propia

Tabla 19. *Índices de Calidad del Modelo Random Forest_DE*

Precisión por Categoría en %				Precisión Global en %
Fallo grave	Fallo leve	Fallo moderado	Sano	Random Forest
0.807	0.9713	0.823	1	0.8828

Fuente: Elaboración propia

Tabla 20. *Importancia de la variable del Modelo Random Forest_DE*

Importancia de las variable de fallo en %		
DE_P.PIT.	DE_B.	DE_P.PEX
0.1519	0.577	0.273

Fuente: Elaboración propia

En la figura 35 y en la tabla 20 se gráfica y observa respectivamente la importancia de cada variable para que este modelo pueda predecir eficientemente, como se observa la clase predominante para ayudar al modelo es el fallo en bola del rodamiento que tiene un 57% de importancia.

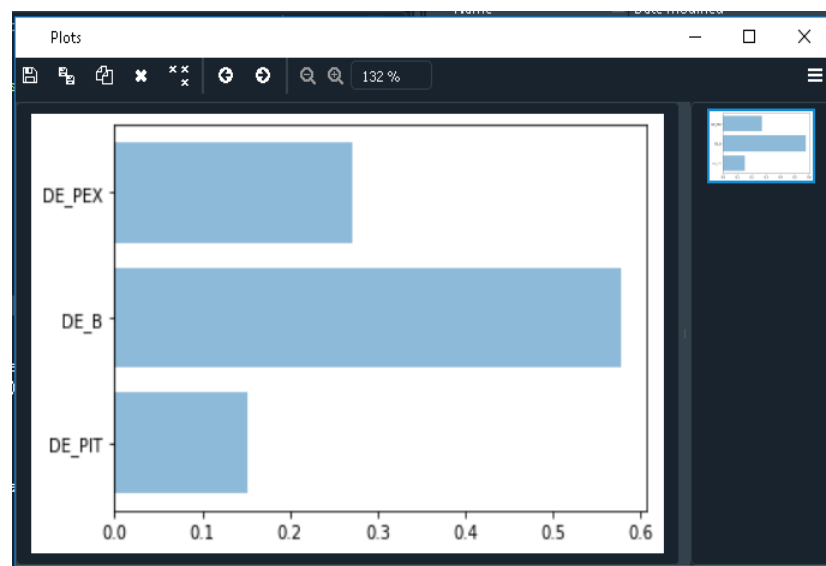


Figura 35. Gráfica de importancia de variable
Fuente: Elaboración propia

El porcentaje de clasificación que se obtiene para este tipo de fallas es bueno y mucho mejor que los modelos presentados anterior mente El modelo entrenado con Random Forest tiene una alta tasa de predicción en las clases pasando todos el 80%, que es un indicativo muy bueno, sumando a eso se tiene como predicción global un 88% de eficacia.

CAPITULO 4

ANÁLISIS DE RESULTADOS

4.1. Resultados

El tiempo de análisis de funcionamiento de los modelos seleccionados compuestos por los algoritmos de Machine Learning es corto en relación al procesamiento de Datos; gracias a que previamente se realizó la “limpieza” de ruidos de las muestras extraídas de la base de datos de la CWRU, esto se detalla en el apartado 3.3 “Preparación de Datos”. en la Figura 36, se ve cómo se desarrolla en el tiempo un proceso típico de un modelo.

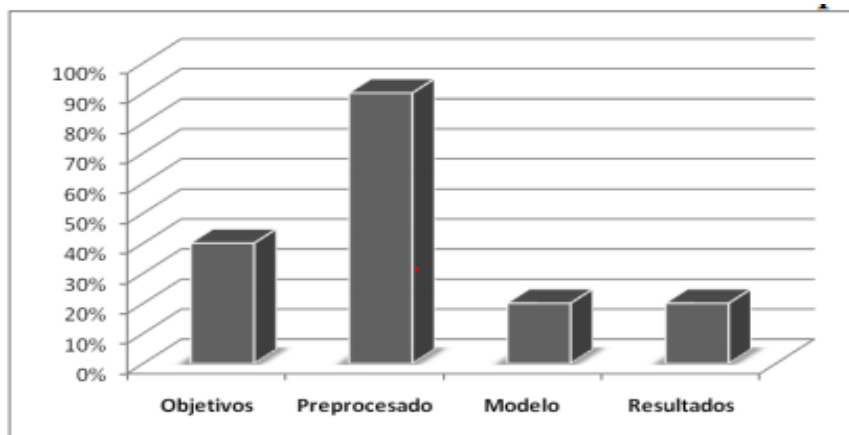


Figura 36. Tiempo estimado de análisis por etapa
Fuente: Corso & Lorena, 2009

4.1.1. Comparativo de eficiencia por modelo predictivo

En este punto se analizará las métricas que validan los resultados obtenidos en el entrenamiento de los modelos seleccionados. Se analizará la eficiencia de todas las matrices de confusión e índices de calidad por modelos (K vecinos más cercanos (KNN), Support Vector Machine (SVM), Bosques Aleatorio (Random Forest)), Esto con el fin de encontrar el modelo predictor más eficiente para el mantenimiento predictivo propuesto, esto en términos de sus índices de calidad tanto global como por categoría. Los comparativos se observarán en las siguientes tablas 21 y 22.

Tabla 21. *Comparativo de Matrices de Confusión*

KNN		PREDICCIÓN			
		Fallo grave	Fallo leve	Fallo moderado	Sano
	Fallo grave	198	4	37	5
VALOR	Fallo leve	3	232	9	0
REAL	Fallo moderado	42	10	176	4
	Sano	0	0	2	89

Random Forest		PREDICCIÓN			
		Fallo grave	Fallo leve	Fallo moderado	Sano
	Fallo grave	197	5	42	0
VALOR	Fallo leve	0	237	7	0
REAL	Fallo moderado	30	11	191	0
	Sano	0	0	0	91

SVM (Rbf)		PREDICCIÓN			
		Fallo grave	Fallo leve	Fallo moderado	Sano
	Fallo grave	7	233	4	0
VALOR	Fallo leve	0	241	3	0
REAL	Fallo moderado	5	215	12	0
	Sano	0	90	1	0

Fuente: Elaboración propia

Tabla 22. *Comparativo de modelos*

Comparativo Global en %		
Precisión Global(KNN)	Precisión Global(SVM)	Precisión Global(Random Forest)
0.8569	0.3205	0.8828

Fuente: Elaboración propia

Como se puede observar en la Tabla 22, los pruebas realizados a los algoritmos funcionan bien con dos modelos (KNN, RF) los cuales no presentan sobreajuste en el momento del entrenamiento que es muy importante acotar, porque cuando se tenga datos nuevos para predecir, lo harán de una forma buena y correcta, sin embargo se debe elegir el mejor modelo predictor y para esto se analiza el comparativo de predicciones globales y categóricas de los dos modelos, descritos en la tabla 22 y 23, donde se observa que el modelo de RF es mayor en porcentaje de predicción que el modelo de KNN.

Tabla 23. *Comparativo de categoría por modelos*

Modelo	Comparativo por Categoría en %		
	KNN	SVM(Rbf)	Random Forest
Fallo grave	0.81	0.0286	0.807
Fallo leve	0.95	0.9877	0.9713
Fallo moderado	0.76	0.0517	0.823
Sano	0.98	0	1

Fuente: Elaboración propia

4.1.2. Comparativo de eficiencia de fallo por categoría

Para estar totalmente seguros del modelo a utilizar observamos y analizamos el comparativo de las categorías por Fallo, cuando el rodamiento no presenta fallo (sano) el mejor modelo es de RF con 100% de efectividad siendo el peor modelo el de SVM con 0% , en el caso del fallo leve el mejor modelo es de SVM con 98.7 %

de eficacia en segundo lugar es de RF con un 97% de acierto, en el caso de fallo moderado se observa que el mejor modelo es RF con un 82% de acierto y el peor fue de SVM con 5%, y finalmente el caso de fallo grave nuevamente el mejor modelo fue el de RF empatado en porcentaje con el KNN, por consiguiente como mejor opción de predicción se escoge Random Forest y como segunda opción KNN vecinos más cercanos.

CONCLUSIONES

Se consiguió diseñar un modelo de mantenimiento predictivo para los motores de inducción, utilizando la inteligencia artificial (Machine Learning) y el Big data o análisis de datos para predecir las fallas del motor, estas dos técnicas forman parte de la industria 4.0, que está revolucionando el mantenimiento en el mundo.

Se obtuvo una adecuada base de datos del repositorio del banco de pruebas de fallas del Centro de rodamientos de la University of Case Western Reserve (CWRU). Todos los archivos de muestras están en formato Matlab (*.mat). Que posteriormente fue convertida en un archivo (.csv) que se utilizó para el entrenamiento de los modelos.

Se desarrolló el programa utilizando el lenguaje de programación Python, con el IDE Spyder como interface de desarrollo, siendo la más adecuada para modelar los diferentes algoritmos de Machine Learning estudiados en el presente trabajo.

Se obtuvieron tres algoritmos de clasificación (K vecinos más cercanos, Máquina de Vector de Soporte y Random Forest), estos fueron seleccionados teniendo en cuenta el marco teórico desarrollado en el capítulo 2, donde es sustentado por revistas especializadas como herramientas efectivas para los procesos de predicción de fallas, cuando se realiza un mantenimiento predictivo.

Se desarrollo e implemento el índice de calidad dentro del programa a entrenar para determinar el mejor modelo predictivo, tanto por categoría como de forma global.

Se selecciona el mejor modelo una vez concluido los comparativos en los índices de calidad globales y categóricos, dentro de los tres algoritmos elegidos previamente; el modelo con mayor grado de predicción en fallo de rodamiento por categoría es en este caso Random Forest, con un porcentaje de predicción de un 80% en fallos graves, 97% de fallos leves, 82% de fallos moderados y un 100 % de sanos. Complementando con las predicciones globales por modelo, Random Forest tiene el porcentaje más alto de predicción con un 88%. Por lo tanto, el mejor modelo de mantenimiento predictivo en motores en este estudio es Random Forest.

RECOMENDACIONES

Se recomienda el estudio e investigación de más algoritmos de Machine Learning que se están desarrollando actualmente sobre el mantenimiento predictivo.

Se recomienda el estudio e implementación de modernas técnicas de adquisición, acondicionamiento y procesamiento de datos, que son importantes para realizar una buena predicción de fallos.

Se recomienda el estudio de algoritmos y técnicas cada vez más modernas como por ejemplo el Aprendizaje Profundo o Deep Learning.

BIBLIOGRAFÍA

- Botero, M., Cañon, B., & Olarte, W. (2010). Técnicas De Mantenimiento Predictivo Utilizadas En La Industria. *Scientia Et Technica*, 16(45), 223–226. Retrieved from <http://www.redalyc.org/articulo.oa?id=84917249041%5Chttp://www.redalyc.org/pdf/849/84917249041.pdf>
- Breinan, L., & Cutler, A. (n.d.). Random forest. Retrieved March 24, 2021, from https://es.wikipedia.org/wiki/Random_forest
- Candanedo, I. S., González, S. R., & Muñoz, L. (2018). Diseño de un Modelo Predictivo en el Contexto Industria 4.0. *KnE Engineering*, 3(1), 543–551. <https://doi.org/10.18502/keg.v3i1.1458>
- Cano, G., Luque, M. ;, Sendra, A. ;, Ruiz, L., Ramón, J., Roldán, C., ... Manuel, C. : (2019). Mantenimiento Predictivo Mediante Tecnicas de Machine Learninig. *AEIPRO, 23 International Congress on Project Management and Engineering*, 03(020), 721–730. Retrieved from <http://dspace.aeipro.com/xmlui/handle/123456789/2293>
- Case Western, & University Reserve (CWRU). (n.d.). Download a Data File | Bearing Data Center. Retrieved November 5, 2020, from Centro de datos de rodamientos de la University Case Western Reserve (CWRU). website: <https://csegroups.case.edu/bearingdatacenter/pages/download-data-file>
- Corso, I., & Lorena, C. (2009). Aplicación de algoritmos de clasificación supervisada usando Weka. In *investigacion.frc.utn.edu.ar*. Retrieved from https://www.investigacion.frc.utn.edu.ar/labsis/Publicaciones/congresos_labsis/cynthia/CNIT_

2009_Aplicacion_Algoritmos_Weka.pdf

- Cortes Cherta, M. (1994). *Máquinas de corriente alterna asíncrona*. Retrieved from [https://books.google.com.pe/books?id=n4T3ipxRZD8C&pg=PA173&dq=motor+con+doble+jaula+de+ardilla&hl=es&sa=X&ved=2ahUKEwjDmY2gpKHsAhXos1kKHcJ-CvsQ6AEwAHoECAAQAg#v=onepage&q=motor con doble jaula de ardilla&f=false](https://books.google.com.pe/books?id=n4T3ipxRZD8C&pg=PA173&dq=motor+con+doble+jaula+de+ardilla&hl=es&sa=X&ved=2ahUKEwjDmY2gpKHsAhXos1kKHcJ-CvsQ6AEwAHoECAAQAg#v=onepage&q=motor+con+doble+jaula+de+ardilla&f=false)
- Curioso, W. H., & Brunette, M. J. (2020). Inteligencia artificial e innovación para optimizar el proceso de diagnóstico de la tuberculosis. *Rev Peru Med Exp Salud Publica*, 37(3), 356–360. <https://doi.org/10.17843/rpmesp.2020.373.5585>
- Da Costa Burga, M. (2010). *Aplicacion del Mantenimiento Centrado en la Cofiability a Motores a Gas de Dos Tiempos en Pozos de Alta Produccion* (universidad Catolica Del Peru). Retrieved from http://tesis.pucp.edu.pe/repositorio/bitstream/handle/20.500.12404/567/DA_COSTA_BURGA_MARTÍN_MANTENIMIENTO_MOTORES_GAS.pdf?sequence=1&isAllowed=y
- Del Pozo Gallego, C. (2016). *Aplicación de técnicas de Machine Learning con regularización al diagnóstico de fallos en motores de inducción* (Universidad de Valladolid). Retrieved from <https://uvadoc.uva.es/handle/10324/18063>
- Del Pozo Gallego, C. (2018). *Mantenimiento predictivo en motores de inducción aplicando técnicas de inteligencia artificial y teoría de desequilibrios electromagnéticos*. Universidad de Valladolid.
- Enriquez Harper, G. (2001). *Motores de Inducción/ Motors of Induction*. Retrieved from [https://books.google.com.pe/books?id=IBS13ls30R8C&pg=SL4-PA501&dq=motor+de+inducción+monofasico&hl=es&sa=X&ved=2ahUKEwiHq6aoqKHsAhVGo1kKHRoiCO0Q6AEwA3oECAUQAg#v=onepage&q=motor de induccion monofasico&f=false](https://books.google.com.pe/books?id=IBS13ls30R8C&pg=SL4-PA501&dq=motor+de+inducción+monofasico&hl=es&sa=X&ved=2ahUKEwiHq6aoqKHsAhVGo1kKHRoiCO0Q6AEwA3oECAUQAg#v=onepage&q=motor+de+inducción+monofasico&f=false)
- Genuer, R., Poggi, J.-M., Tuleau-Malot, C., & Villa-Vialaneix, N. (2017). Random Forests for Big Data. In *Elsevier*. Retrieved from <http://hadoop.apache.org>
- Géron, A. (2019). *Aprendizaje automático práctico con Scikit-Learn, Keras y TensorFlow: conceptos, herramientas y técnicas para crear sistemas inteligentes*. Retrieved from [https://books.google.es/books?hl=es&lr=&id=HHetDwAAQBAJ&oi=fnd&pg=PP1&dq=Geron,+A.+\(2018\).+HANDS-ON+MACHINE+LEARNING+WITH+SCIKIT-](https://books.google.es/books?hl=es&lr=&id=HHetDwAAQBAJ&oi=fnd&pg=PP1&dq=Geron,+A.+(2018).+HANDS-ON+MACHINE+LEARNING+WITH+SCIKIT-)

LEARN,+KERAS,+AND+TENSORFLOW&ots=0LneYrfiVs&sig=0Hof0nwKdgVVMX0A1dDtOz
M7YFQ

- He, Q. P., & Wang, J. (2007). Fault Detection Using the k-Nearest Neighbor Rule for Semiconductor Manufacturing Processes. *IEEE TRANSACTIONS ON SEMICONDUCTOR MANUFACTURING*, 20(4). <https://doi.org/10.1109/TSM.2007.907607>
- IRIM, I. R. (2018). LA PÉRDIDA DE PRODUCCIÓN POR UN MAL MANTENIMIENTO. 14. Retrieved from <http://www.renovetec.com/irim/14-revista-irim-6/284-la-perdida-de-produccion-por-un-mal-mantenimiento>
- Janitza, S., Tutz, G., & Boulesteix, A.-L. (2014). Random Forests for Ordinal Response Data: Prediction and Variable Selection. In *Elsevier*. Retrieved from <http://www.stat.uni-muenchen.de>http://www.ibe.med.uni-muenchen.de/organisation/mitarbeiter/070_drittmittel/janitza/index.html.
- José, I., & Rosales Fernández, H. (2016). *MOTORES ELECTRICOS PARA LA INDUSTRIA*. Retrieved from http://www.usmp.edu.pe/vision2016/pdf/materiales/MOTORES_ELECTRICOS_PARA_LA_IN.pdf
- La Torre Garcia, G., Universidad Nacional de San Agustín, & Empresa de Generación Eléctrica de Arequipa. (2017). Evaluación práctica del aislamiento eléctrico en máquinas rotativas de gran potencia. *Investigación Aplicada e Innovación Tecsup*, 11, 44–51.
- López, Y., & Manuel, L. (2019). *Diseño de un modelo de predicción de éxito para proyectos tecnológicos con financiación en masa aplicando técnicas de Machine Learning*. Retrieved from Universidad Católica de Santiago de Guayaquil website: <http://repositorio.ucsg.edu.ec/handle/3317/12617>
- Mansanet, J. (2017). Machine Learning y casos de uso. Retrieved from DECHARLAS.COM website: <https://www.youtube.com/watch?v=p5OhjKQpVYM>
- Martin, J. C., & Garcia, P. (2016). *Motores eléctricos (Automatismos industriales)*. Retrieved from <https://books.google.com.pe/books?id=lrNcDAAQBAJ&printsec=frontcover&dq=motores+electricos&hl=es&sa=X&ved=2ahUKEwj8l7jDnaHsAhWjzlkKHak-Ay0Q6AEwAXoECAUQAg#v=onepage&q&f=false>
- Medrano Sanz, J. I. (2019). *Estudio de técnicas de clasificación para detección y diagnóstico de*

- fallos*. Retrieved from <https://uvadoc.uva.es/handle/10324/37836>
- Molnar, C. (2020). *Aprendizaje automático interpretable*. Retrieved from https://books.google.es/books?hl=es&lr=&id=jBm3DwAAQBAJ&oi=fnd&pg=PP1&dq=machine+learning&ots=EgsV-rFDS_&sig=_beyJSpfPuJa2bw73jq1qha5h0o
- Müller, A. C., & Guido, S. (2016). *Introduction to machine learning with Python: a guide for data scientists*.
- Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. Cambridge, MA: MIT Press. Retrieved from [https://books.google.es/books?hl=es&lr=&id=RC43AgAAQBAJ&oi=fnd&pg=PR7&dq=Murphy,+K.+P.+\(2012\).+Machine+learning:+a+probabilistic+perspective.+Cambridge,+MA:+MIT+Pre+ss.+Nagy,&ots=umfucCSt_c&sig=OW_8zdB1PjyuuMNCjq8DLawcgMo](https://books.google.es/books?hl=es&lr=&id=RC43AgAAQBAJ&oi=fnd&pg=PR7&dq=Murphy,+K.+P.+(2012).+Machine+learning:+a+probabilistic+perspective.+Cambridge,+MA:+MIT+Pre+ss.+Nagy,&ots=umfucCSt_c&sig=OW_8zdB1PjyuuMNCjq8DLawcgMo)
- Nagy, Z. (2018). *Artificial Intelligence and Machine Learning Fundamentals: Develop real-world applications powered by the latest AI advances*. Retrieved from [https://books.google.es/books?hl=es&lr=&id=IW5_DwAAQBAJ&oi=fnd&pg=PR1&dq=Nagy,+Z.+\(2018\).+BEGINNING+AI,+MACHINE+LEARNING+AND+PYTHON:+get+started+with+the+development+of+real-world+...+applications+that+are+powered+by+the+latest+ai+adv.+Place+of+publicatio](https://books.google.es/books?hl=es&lr=&id=IW5_DwAAQBAJ&oi=fnd&pg=PR1&dq=Nagy,+Z.+(2018).+BEGINNING+AI,+MACHINE+LEARNING+AND+PYTHON:+get+started+with+the+development+of+real-world+...+applications+that+are+powered+by+the+latest+ai+adv.+Place+of+publicatio)
- Peñaranda Castañeda, C. (2018). Perú debe superar desafíos para acceder a la cuarta revolución industrial. Retrieved from LA CAMARA website: https://apps.camaralima.org.pe/repositorioaps/0/0/par/r823_2/iedep.pdf
- Picazo Rodenas, M. J. (2016). *Diagnóstico de máquinas eléctricas mediante técnicas de termografía infrarroja* (Universitat Politècnica de València). <https://doi.org/10.4995/Thesis/10251/62317>
- Raschka, S., & Mirjalili, V. (2017). Aprendizaje automático de Python: aprendizaje automático y aprendizaje profundo con Python. *Igi-Global.Com*, 2. Retrieved from <https://www.igi-global.com/pdf.aspx?tid%3D267132%26ptid%3D254262%26ctid%3D17%26t%3Dpython+machine+learning%3A+machine+learning+and+deep+learning+with+python%2C+scikit-learn%2C+and+tensorflow+2%2C+third+edition%26isxn%3D>
- Sansano, E. (2017). Deep Learning: Una introducción práctica. Retrieved from DECHARLAS.COM website: <https://www.youtube.com/watch?v=kVFflnP9S8Y>

- Serano, S. P. (2019). *Aplicación de Técnicas de Procesamiento para el Mantenimiento Basado en la Condición*.
- SFK. (2017). fallos rodamientos skf. Retrieved October 7, 2020, from PUB BU/1317186 ES website:
https://www.google.com.pe/search?source=hp&ei=21V9X4LXMeSP5wKGn5_4Bg&q=fallos+rodamientos+skf&oq=FALLOS+DE+UN+RODAMIENTO&gs_lcp=CgZwc3ktYWIQARgBMgYIABAWEB4yBggAEBYQHjoICAAQsQMqgwE6AggAOgUILhCxAzofCAAQsQM6AgguOggILhCxAxCTAjoECAAQckoGCAQSAjlySgUIBxIBMUoFCAg
- Sitton, I., Hernandez, E., Rodriguez, S., Santos, M. T., & Gonzales, A. (2018). Machine learning predictive model for Industry 4.0. *2018 First International Conference on Artificial Intelligence for Industries (AI4I)*, 1(July), 537–548. <https://doi.org/10.1007/978-3-319-95204-8>
- Stavrou, A., Sedding, H. G., & Penman, J. (2001). Current monitoring for detecting inter-turn short circuits in induction motors. *IEEE Transactions on Energy Conversion*, 16(1), 32–37.
<https://doi.org/10.1109/60.911400>
- Zhang, S., Zhang, S., Wang, B., & Habetler, T. G. (2020, January 24). Deep Learning Algorithms for Bearing Fault Diagnostics - A Comprehensive Review. *IEEE Access*, Vol. 8, pp. 29857–29881. <https://doi.org/10.1109/ACCESS.2020.2972859>

ANEXOS

ANEXO 1, Programas utilizados para el entrenamiento de los modelos de predicción de motores de inducción.

ANEXO 1.1, Entrenamiento con el modelo KNN

@author: José Luis Contreras

```
import os

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import mglearn

from sklearn.model_selection import train_test_split

from sklearn.neighbors import KNeighborsClassifier

from sklearn.metrics import confusion_matrix

# Para mostrar un gráfico en un archivo rmd use show () y close ()

def open_close_plot ():

    plt. show ()

    plt. Close ()

# Indices para matrices NxN

def indices_general (MC, nombres=None):

    precision_global=np.sum (MC. diagonal ()) /np.sum (MC)

    error_global=1-precision_global

    precision_categorica=pd. DataFrame (MC. diagonal () /np.sum (MC, axis=1)). T

    if nombres! =None:

        precision_categorica. columns=nombres

return {"MATRIZ DE CONFUSIÓN":MC,

        "PRECISIÓN GLOBAL": precision_global,

        "ERROR GLOBAL": error_global,

        "PRECISIÓN POR CATEGORÍA": precision_categorica}

#llamando Datos en tabla para el modelo

datos=pd. read_csv ('Train_DE.csv', delimiter=',', decimal=".")

print (datos. shape)
```

```

print (datos. head ())

print (datos.info ())

#Variable a pronosticar en X
X=datos.iloc[:, : 3]

print (X. head ())

#Variable a pronosticar en y
y=datos.iloc[:,3:4]

print (y. head ())

# Generación de tablas de entrenamiento(traen) y prueba(testing)
# test_size="..." nos indica que porcentaje de la tabla tomamos para entrenar
# random_state="0 o 42" nos indica que porcentaje de la tabla tomamos para entrenar
# "0"se activa la forma aleatoria cada vez que se corra el programa
# "42"se desactiva la forma aleatoria y se queda con la primera forma seleccionada cada vez que
se corra el programa

Train, Test, y_train, y_test=train_test_split(X,y,train_size=0.7, random_state=42)

print(X_train)

print(y_test)

#Mediante el constructor inicializa el atributo n_neighbors (este viene hacer el "K")
instancia_knn=KNeighborsClassifier(n_neighbors=7)

#Entrena el modelo llamando al método fit

#Nota: el modelo queda en un atributo de la instancia "instancia_knn"

instancia_knn.fit (X_train, y_train)

#imprime las predicciones en Testing

print ("Las predicciones en Testing son: {}". format (instancia_knn. predict(X_test)))

#Porcentaje de predicción Global

print ("Las predicciones en Testing KNN son: {:.2f}". format (instancia_knn. score (X_test, y_test)))

#MATRIZ DE CONFUSIÓN

predicción=instancia_knn. predict(X_test)

```

```

MC=confusion_matrix (y_test, prediccion)

print ("Matriz de Confusión: \n {}".format (MC))

# ÍNDICES DE CALIDAD DEL MODELO

índices=índices_general (MC, list (np. unique(y)))

for k in índices:

    print ("\n%s:\n%s"% (k, str(índices[k])))

```

ANEXO 1.2, Entrenamiento con el modelo SVM

```

import os

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import mglearn

from sklearn.model_selection import train_test_split

from sklearn.svm import SVC

from sklearn.metrics import confusion_matrix

#---Para mostrar un gráfico en un archivo rmd use show () y close ()

def open_close_plot ():

    plt. show ()

    plt. Close ()

#---Indices para matrices NxN

def índices_general (MC, nombres=None):

    precision_global=np.sum(MC. diagonal())/np.sum(MC)

    error_global=1-precision_global

    precision_categorica=pd.DataFrame(MC.diagonal()/np.sum(MC,axis=1)).T

    if nombres! =None:

        precision_categorica.columns=nombres

return {"MATRIZ DE CONFUSIÓN":MC,

        "PRECISIÓN GLOBAL":precision_global,

        "ERROR GLOBAL":error_global,

```

"PRECISIÓN POR CATEGORÍA": precision_categorica}

```
datos=pd.read_csv('Train_DE.csv',delimiter=',',decimal=".")

print (datos. shape)

print (datos. head ())

print (datos.info ())

#---elimina la variable categórica, deja la variable a pronosticar en X

X=datos.iloc[:,3]

print(X.head())

#---deja la variable a pronosticar en y

y=datos.iloc[:,3:4]

print(y.head())

#---Generación de tablas de entrenamiento(train) y prueba(testing)

X_train,X_test,y_train,y_test=train_test_split(X,y,train_size=0.7,random_state=42)

print(X_train)

print(y_test)

#---Mediante el constructor inicializa_SVM (USANDO PARÁMETROS POR DEFECTO)

#---Usando otros núcleos

#---uno de ellos es el Kernel:"linial","poly", "rbf", , por defecto usa

#---Kernel="rbf"

instancia_svm_rbf=SVC(kernel='rbf')

instancia_svm_linear=SVC(kernel='lineal')

instancia_svm_poly=SVC(Kernel='poly')

#---Visualizando parámetros

print(instancia_svm_rbf)

print(instancia_svm_linear)

print(instancia_svm_poly)

#---Entrena el modelo llamando al método fit

#---Nota: el modelo queda en un atributo de la instancia "instancia_svm"
```

```

instancia_svm_rbf.fit (X_train, y_train)

instancia_svm_lineal.fit (X_train, y_train)

instancia_svm_poly.fit (X_train, y_train)

#---imprime las predicciones en Testing

print ("Las predicciones en Testing son: {}".format(instancia_svm_rbf.predict(X_test)))

#---Porcentaje de predicción Global

print ("Las predicciones en Testing rbf son: {:.3f}". format(instancia_svm_rbf.score (X_test,y_test)))

#---MATRIZ DE CONFUSIÓN

predicción=instancia_svm_rbf.predict(X_test)

MC=confusion_matrix(y_test,prediccion)

print ("Matriz de Confusión: \n{}".format(MC))

#---ÍNDICES DE CALIDAD DEL MODELO

índices=indices_general (MC, list(np.unique(y)))

for k in índices:

    print ("\n%s:\n%s"% (k, str(índices[k])))

#####

#---imprime las predicciones en Testing

print ("Las predicciones en Testing son: {}".format(instancia_svm_linear.predict(X_test)))

#---Porcentaje de predicción Global

print ("Las predicciones en Testing linear son: {:.3f}".

format(instancia_svm_linear.score(X_test,y_test)))

#---MATRIZ DE CONFUSIÓN

predicción=instancia_svm_linear.predict(X_test)

MC=confusion_matrix(y_test,prediccion)

print ("Matriz de Confusión: \n {}".format (MC))

```

```

#---ÍNDICES DE CALIDAD DEL MODELO

índices=indices_general (MC, list(np.unique(y)))

for k in indices:

    print ("\n%s:\n%s"% (k, str(indices[k])))

#####

#---imprime las predicciones en Testing

print ("Las predicciones en Testing son: {}".format(instancia_svm_poly.predict(X_test)))

#---Porcentaje de predicción Global

print ("Las predicciones en Testing poly son: {:.3f}".
format(instancia_svm_poly.score(X_test,y_test)))

#---MATRIZ DE CONFUSIÓN

predicción=instancia_svm_poly. predict(X_test)

MC=confusion_matrix(y_test,prediccion)

print ("Matriz de Confusión: \n {}".format (MC))

#---ÍNDICES DE CALIDAD DEL MODELO

índices=indices_general (MC, list(np.unique(y)))

for k in indices:

    print ("\n%s:\n%s"% (k, str(indices[k])))

```

ANEXO 1.3, Entrenamiento con el modelo Random Forest

```

import os

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import mglearn

from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import confusion_matrix

# Para mostrar un gráfico en un archivo rmd use show () y close ()

```

```

def open_close_plot ():
    plt. show ()
    plt. Close ()

# Índices para matrices NxN

def indices_general (MC, nombres=None):
    precision_global=np.sum (MC. diagonal ()) /np.sum (MC)
    error_global=1-precision_global
    precision_categorica=pd.DataFrame(MC.diagonal()/np.sum(MC,axis=1)).T

if nombres! =None:
    precision_categorica. columns=nombres

return {"MATRIZ DE CONFUSIÓN":MC,
        "PRECISIÓN GLOBAL": precision_global,
        "ERROR GLOBAL": error_global,
        "PRECISIÓN POR CATEGORÍA": precision_categorica}

#llamando Datos en tabla para el modelo
datos=pd. read_csv ('Train_DE.csv', delimiter=',', decimal=".")

print (datos. shape)
print (datos. head ())
print (datos.info ())

#elimina la variable categórica, da la variable a pronosticar en X
X=datos.iloc[:, :3]
print (X. head())

# deja la variable a pronosticar en y
y=datos. iloc[:, 3:4]
print (y. head ())

# Generación de tablas de entrenamiento(train) y prueba(testing)
#test_size="..." nos indica que porcentaje de la tabla tomamos para entrenar
#random_state="0 o 42" nos indica que porcentaje de la tabla tomamos para entrenar
#"0"se activa la forma aleatoria cada vez que se corra el programa

```


#"42"se desactiva la forma aleatoria y se queda con la primera forma seleccionada cada vez que se corra el programa

```
X_train, X_test, y_train, y_test=train_test_split(X,y,train_size=0.7,random_state=42)
```

```
print (X_train)
```

```
print (y_test)
```

#Mediante el constructor inicializa la instancia bosques

```
instancia_bosque=RandomForestClassifier (n_estimators=40, max_depth=None
```

```
,min_samples_split=2, random_state=42)
```

#Entrena el modelo llamando al método fit

#Nota: el modelo queda en un atributo de la instancia "instancia bosques"

```
instancia_bosque.fit (X_train, y_train)
```

#imprime las predicciones en Testing

```
print ("Las predicciones en Testing son: {}". format (instancia_bosque. predict(X_test)))
```

#Porcentaje de predicción Global

```
print ("Las predicciones en Testing Random Forest son: {:.2f}". format (instancia_bosque. Score
```

```
(X_test, y_test)))
```

#MATRIZ DE CONFUSIÓN

```
predicción=instancia_bosque.predict(X_test)
```

```
MC=confusion_matrix(y_test,prediccion)
```

```
print ("Matriz de Confusión: \n {}". format (MC))
```

ÍNDICES DE CALIDAD DEL MODELO

```
índices=indices_general (MC, list (np. unique(y)))
```

for k in indices:

```
    print ("\n%s:\n%s"% (k, str(índices[k])))
```

#IMPORTANCIA DE LAS VARIABLES

```
importancia=instancia_bosque. feature_importances_
```

```
print ("importancia de la variable", importancia)
```

```

# GRAFICANDO LA IMPORTANCIA
etiquetas=X_train. columns. values
y_pos=np. arange(len(etiquetas))
plt. barh (y_pos, importancia, align='center', alpha=0.5)
plt. yticks (y_pos, etiquetas)
open_close_plot ()

```

ANEXO 2, Parámetros Eléctricos del motor utilizado por la CWRU.

Potencia	2 Hp.
Tensión nominal	3x230/400 V.
Frecuencia nominal	50 Hz
Corriente nominal	3.7 A
Velocidad nominal	1720 rpm
Índice de protección	IP55
Coseno/ factor de potencia	0.9
Tipo de conexión	Delta

ANEXO 3, Tipos de acelerómetro (sensores) utilizados para la adquisición de datos

Se propone utilizar sensores capacitivos (MEMS), por reducir costos y simplificar el despliegue con respecto a los sensores piezoeléctricos.

ADcmXL3021	 A photograph of the ADcmXL3021 MEMS accelerometer. It is a rectangular, silver-colored metal component with a white label on top that reads "ANALOG DEVICES ADcmXL3021BLJZ P4". A green ribbon cable is attached to the top right side of the component.
ADIS16628	 A photograph of the ADIS16628 MEMS accelerometer. It is a rectangular, silver-colored metal component with a white label on top that reads "ANALOG DEVICES ADIS16628". A green ribbon cable is attached to the bottom left side of the component.

ANEXO 4, Muestras utilizadas en el entrenamiento para validar los modelos predictivos.

Ensayos	DE_PIT	DE_B	DE_PEX	CLASE	Ensayos	DE_PIT	DE_B	DE_PEX	CLASE
1	0.699187	0	0	sano	1	0.551478	0.172453	0.272383	fallo leve
2	0.031075	0	0	sano	2	0.003201	0.098037	0.010486	fallo leve
3	0.048198	0	0	sano	3	0.00562	0.097106	0.002748	fallo leve
4	0.040059	0	0	sano	4	0.008118	0.102732	0.00582	fallo leve
5	0.027383	0	0	sano	5	0.00558	0.102508	0.004534	fallo leve
6	0.02289	0	0	sano	6	0.006123	0.100666	0.002297	fallo leve
7	0.011962	0	0	sano	7	0.001924	0.100204	0.003481	fallo leve
8	0.005413	0	0	sano	8	0.003169	0.095491	0.000421	fallo leve
9	0.030905	0	0	sano	9	0.001518	0.092567	0.00122	fallo leve
10	0.001912	0	0	sano	10	0.002478	0.089938	0.001339	fallo leve
11	0.016862	0	0	sano	11	0.002216	0.089912	0.001181	fallo leve
12	0.009273	0	0	sano	12	0.001157	0.086048	0.000959	fallo leve
13	0.006062	0	0	sano	13	0.001202	0.082287	0.000193	fallo leve
14	0.010203	0	0	sano	14	0.001144	0.078065	0.000256	fallo leve
15	0.00602	0	0	sano	15	0.001004	0.073733	0.000341	fallo leve
16	0.005709	0	0	sano	16	0.000609	0.069342	0.000986	fallo leve
17	0.012997	0	0	sano	17	0.000679	0.065354	0.000412	fallo leve
18	0.015458	0	0	sano	18	0.000777	0.060982	0.000544	fallo leve
19	0.003551	0	0	sano	19	0.000852	0.056371	0.000963	fallo leve
20	0.01046	0	0	sano	20	0.000383	0.051116	0.00043	fallo leve
21	0.007228	0	0	sano	21	0.000758	0.04593	0.000464	fallo leve
22	0.00164	0	0	sano	22	0.000982	0.041341	5.22E+00	fallo leve
23	0.005078	0	0	sano	23	0.00055	0.036534	0.000549	fallo leve
24	0.005484	0	0	sano	24	0.000418	0.032202	0.00069	fallo leve
25	0.0045	0	0	sano	25	0.0002	0.027773	0.000788	fallo leve
26	0.006847	0	0	sano	26	0.000596	0.022519	0.000435	fallo leve
27	0.005877	0	0	sano	27	0.000688	0.01796	0.000558	fallo leve
28	0.004339	0	0	sano	28	0.000327	0.014756	0.0005	fallo leve
29	0.003734	0	0	sano	29	0.000787	0.012208	0.000513	fallo leve
30	0.010043	0	0	sano	30	0.000368	0.011093	0.000167	fallo leve
31	0.004592	0	0	sano	31	9.28E-01	0.011303	0.000374	fallo leve
32	0.00572	0	0	sano	32	0.000285	0.012516	6.86E+00	fallo leve
33	0.00125	0	0	sano	33	0.000941	0.015762	0.000531	fallo leve
34	0.007195	0	0	sano	34	0.00057	0.019706	0.00011	fallo leve
35	0.004886	0	0	sano	35	0.000483	0.023332	0.000525	fallo leve
36	0.002164	0	0	sano	36	0.001156	0.026593	0.000126	fallo leve
37	0.00485	0	0	sano	37	0.000183	0.029786	0.00048	fallo leve
265	0.000709	0	0	sano	764	0.000681	0.003594	0.000349	fallo leve
266	0.000261	0	0	sano	765	0.000144	0.003455	0.00051	fallo leve
267	0.000427	0	0	sano	766	0.000816	0.003078	0.000215	fallo leve
268	0.000648	0	0	sano	767	0.000358	0.003249	0.000405	fallo leve
269	0.000278	0	0	sano	768	0.000562	0.002637	0.000364	fallo leve
270	0.000694	0	0	sano	769	0.000135	0.001738	0.000201	fallo leve
271	0.000346	0	0	sano	770	0.000559	0.000886	0.000309	fallo leve
272	0.000624	0	0	sano	771	0.000826	0.00163	0.000377	fallo leve
273	0.000971	0	0	sano	772	0.000776	0.005318	0.000503	fallo leve
274	0.000912	0	0	sano	773	0.000864	0.004078	0.000359	fallo leve
275	0.001043	0	0	sano	774	0.00056	0.000443	0.000765	fallo leve
276	0.000348	0	0	sano	775	0.001003	0.003957	0.000536	fallo leve
277	0.001035	0	0	sano	776	0.001857	0.0017	0.000773	fallo leve
278	0.000603	0	0	sano	777	0.001202	0.002422	0.002431	fallo leve
279	0.001193	0	0	sano	778	0.007679	0.003611	0.001682	fallo leve
280	5.93E-01	0	0	sano	779	0.001132	0.00394	0.000444	fallo leve
281	0.000928	0	0	sano	780	0.00116	0.003922	0.00022	fallo leve
282	0.000166	0	0	sano	781	0.000753	0.003778	4.02E+00	fallo leve
283	0.000219	0	0	sano	782	0.000723	0.003522	8.69E+00	fallo leve
284	0.000713	0	0	sano	783	0.000624	0.003653	0.000406	fallo leve
285	0.000188	0	0	sano	784	0.000616	0.003752	0.000154	fallo leve
286	0.001145	0	0	sano	785	5.49E+00	0.0048	0.000552	fallo leve
287	0.001568	0	0	sano	786	0.000133	0.004903	0.000375	fallo leve
288	0.001028	0	0	sano	787	0.000856	0.005125	0.000202	fallo leve
289	0.000469	0	0	sano	788	0.000748	0.004643	0.000146	fallo leve
290	0.001252	0	0	sano	789	0.00048	0.004241	0.000243	fallo leve
291	0.001101	0	0	sano	790	0.000684	0.004777	0.000248	fallo leve
292	0.001147	0	0	sano	791	0.00084	0.004119	0.000246	fallo leve
293	0.000178	0	0	sano	792	0.0005	0.004281	0.000293	fallo leve
294	2.69E+00	0	0	sano	793	0.000366	0.004347	0.00042	fallo leve
295	0.000283	0	0	sano	794	0.001072	0.004231	0.000104	fallo leve
296	0.001584	0	0	sano	795	0.000131	0.003079	0.000247	fallo leve
297	0.0018	0	0	sano	796	0.000332	0.00292	0.000229	fallo leve
298	0.002937	0	0	sano	797	0.000192	0.002487	0.000182	fallo leve
299	0.006906	0	0	sano	798	0.001464	0.001904	0.000418	fallo leve
300	0.000754	0	0	sano	799	0.00123	0.001727	0.000311	fallo leve
					800	0.000606	0.001117	0.000115	fallo leve

Ensayos	DE_PIT	DE_B	DE_PEX	CLASE	Ensayos	DE_PIT	DE_B	DE_PEX	CLASE
1	0.370486	0.28171	0.61603	fallo moderado	1	0.054304	0.117639	0.556699	fallo grave
2	0.004936	0.002986	0.010613	fallo moderado	2	0.020675	0.002655	0.011814	fallo grave
3	0.008655	0.007364	0.006793	fallo moderado	3	0.048784	0.001977	0.012272	fallo grave
4	0.004314	0.003174	0.005919	fallo moderado	4	0.012374	0.001323	0.004495	fallo grave
5	0.002029	0.002906	0.004297	fallo moderado	5	0.028636	0.001185	0.005916	fallo grave
6	0.00165	0.002144	0.002954	fallo moderado	6	0.028517	0.001212	0.006145	fallo grave
7	0.002658	0.00074	0.002363	fallo moderado	7	0.02117	0.001128	0.004701	fallo grave
8	0.001705	0.001259	0.002393	fallo moderado	8	0.011304	0.000339	0.005291	fallo grave
9	0.001354	0.001672	0.000397	fallo moderado	9	0.008686	0.000435	0.004803	fallo grave
10	0.000663	0.0008	0.000817	fallo moderado	10	0.007624	0.000432	0.002713	fallo grave
11	0.001019	0.000633	0.000869	fallo moderado	11	0.019415	0.000659	0.002162	fallo grave
12	0.001065	0.000599	0.002204	fallo moderado	12	0.00585	0.000675	0.002979	fallo grave
13	0.00091	0.001088	0.000451	fallo moderado	13	0.01397	0.000607	0.002416	fallo grave
14	0.001133	0.001009	0.001081	fallo moderado	14	0.013358	0.000691	0.002639	fallo grave
15	0.000606	0.000764	0.000477	fallo moderado	15	0.004151	0.000602	0.001434	fallo grave
16	0.000643	0.000415	0.000879	fallo moderado	16	0.001847	0.000237	0.001633	fallo grave
17	0.000406	0.001163	0.000435	fallo moderado	17	0.003451	0.000258	0.00162	fallo grave
18	0.000495	0.000158	0.000735	fallo moderado	18	0.0047	0.000431	0.00244	fallo grave
19	0.000484	0.000872	0.000822	fallo moderado	19	0.006093	0.000289	0.001089	fallo grave
20	0.000239	0.00051	0.000363	fallo moderado	20	0.015973	0.000288	0.000907	fallo grave
21	0.000433	0.000427	0.000133	fallo moderado	21	0.008559	0.00031	0.001204	fallo grave
22	0.000232	0.000792	0.000512	fallo moderado	22	0.010121	0.000345	0.001114	fallo grave
23	0.000468	0.000395	0.000272	fallo moderado	23	0.002953	0.000328	0.002054	fallo grave
24	0.000151	0.000616	0.000256	fallo moderado	24	0.008615	0.00026	0.001401	fallo grave
25	0.000213	0.000295	0.000109	fallo moderado	25	0.002827	0.000234	0.001251	fallo grave
26	0.000195	0.0005	0.000493	fallo moderado	26	0.011822	0.000317	0.001012	fallo grave
27	0.000739	0.000359	0.000468	fallo moderado	27	0.01138	0.000432	0.000841	fallo grave
28	0.000269	0.000456	0.000978	fallo moderado	28	0.007915	0.000373	0.000613	fallo grave
29	0.000206	0.000399	0.000445	fallo moderado	29	0.007169	0.000386	0.00118	fallo grave
30	0.000299	0.000374	0.000221	fallo moderado	30	0.001211	0.000219	0.000496	fallo grave
31	0.000338	0.000726	0.000543	fallo moderado	31	0.00289	0.000302	0.000999	fallo grave
32	0.000371	0.000214	0.00083	fallo moderado	32	0.002306	0.000297	0.001246	fallo grave
33	0.000164	0.000558	0.000439	fallo moderado	33	0.002007	0.00035	0.001075	fallo grave
34	0.000369	0.000529	0.000795	fallo moderado	34	0.007506	0.000295	0.00046	fallo grave
35	0.000307	0.000914	0.000125	fallo moderado	35	0.00511	0.000274	0.001268	fallo grave
36	0.000219	0.000612	0.000743	fallo moderado	36	0.003941	0.000291	0.000714	fallo grave
37	0.000269	0.0006	0.000582	fallo moderado	37	0.000843	6.98E-01	0.000483	fallo grave
765	0.000251	0.000272	0.000343	fallo moderado	765	0.000582	0.000151	0.00065	fallo grave
766	0.00047	0.000115	8.37E+00	fallo moderado	766	0.000568	8.78E+00	0.000487	fallo grave
767	0.001068	0.000703	0.000918	fallo moderado	767	0.000208	0.000173	0.000182	fallo grave
768	0.000511	0.000653	0.000544	fallo moderado	768	0.00078	0.00018	0.00092	fallo grave
769	0.000606	0.000107	0.000668	fallo moderado	769	0.000603	0.000154	0.000402	fallo grave
770	0.000689	0.00054	0.000376	fallo moderado	770	0.000625	0.000305	0.000359	fallo grave
771	0.000599	0.000629	0.00118	fallo moderado	771	0.000561	0.000293	0.000529	fallo grave
772	0.000421	0.000235	0.000796	fallo moderado	772	0.000835	0.000101	0.001892	fallo grave
773	0.001007	0.000477	0.001302	fallo moderado	773	0.000369	0.000199	0.002513	fallo grave
774	0.00061	0.000536	0.000749	fallo moderado	774	0.001906	6.60E-02	0.002291	fallo grave
775	0.001258	0.000753	0.001598	fallo moderado	775	0.000617	0.00035	0.001014	fallo grave
776	0.00059	0.000718	0.00247	fallo moderado	776	0.00062	0.000418	0.000607	fallo grave
777	0.000923	0.000614	0.005116	fallo moderado	777	0.001599	0.000278	0.000859	fallo grave
778	0.004058	0.001174	0.003165	fallo moderado	778	0.000866	0.00103	0.001285	fallo grave
779	0.002685	0.001037	0.001094	fallo moderado	779	0.001409	0.000479	0.001054	fallo grave
780	0.002025	0.001656	0.000646	fallo moderado	780	0.001111	0.000263	0.000932	fallo grave
781	0.001559	0.003276	0.000857	fallo moderado	781	0.000432	0.000127	0.000515	fallo grave
782	0.000882	0.000931	0.00077	fallo moderado	782	0.000801	0.000358	0.000628	fallo grave
783	0.001383	0.000731	0.000405	fallo moderado	783	0.00021	0.000339	0.000583	fallo grave
784	0.001278	7.98E+00	0.000472	fallo moderado	784	0.000244	0.000177	0.000975	fallo grave
785	0.00262	0.000441	0.000585	fallo moderado	785	0.000764	0.000285	0.000787	fallo grave
786	0.000892	0.000715	0.000591	fallo moderado	786	0.000876	0.000125	0.00092	fallo grave
787	0.001004	0.000634	0.000631	fallo moderado	787	0.000306	0.000172	0.000991	fallo grave
788	0.00074	0.000323	0.000174	fallo moderado	788	0.000374	0.000268	0.001598	fallo grave
789	0.000764	0.000367	0.000625	fallo moderado	789	0.001149	0.000133	0.000545	fallo grave
790	0.000311	0.000264	0.000669	fallo moderado	790	0.00068	0.000262	0.000801	fallo grave
791	0.000776	0.000184	0.000659	fallo moderado	791	0.00053	0.00017	0.001813	fallo grave
792	0.000709	0.000221	0.000418	fallo moderado	792	0.000394	0.00016	0.001042	fallo grave
793	0.000796	0.00023	0.000363	fallo moderado	793	0.001098	0.0002	0.000385	fallo grave
794	0.00046	0.000655	0.000731	fallo moderado	794	0.000796	0.000142	0.000725	fallo grave
795	0.000819	0.000176	0.000618	fallo moderado	795	0.000582	5.74E+00	0.000327	fallo grave
796	0.000481	8.16E+00	0.001268	fallo moderado	796	0.000688	0.00019	0.000215	fallo grave
797	0.00117	0.000421	0.000902	fallo moderado	797	0.000579	0.000141	0.001799	fallo grave
798	0.000764	0.000502	0.000838	fallo moderado	798	0.000518	0.000232	0.000748	fallo grave
799	0.000404	0.000242	0.000987	fallo moderado	799	0.001099	0.000372	0.00065	fallo grave
800	0.000296	0.000313	0.000533	fallo moderado	800				

GLOSARIO

Mantenimiento predictivo, Consiste en una serie de acciones que se toman y diferentes técnicas que se aplican con el único objetivo detectar las fallas en su estado incipiente.

Motor de inducción, Es una maquina eléctrica que costa de un estator y un rotor encargados de transformar la energía eléctrica en energía mecánica.

Rodamientos, Los rodamientos son elementos mecánicos encargados de unir partes distintas de una maquina en movimiento rotativo entre sí, de este modo los rodamientos trasmiten la fuerza y movimiento a los componentes, y a la vez ayuda a reducir la fricción que hay entre ellos.

Industria 4.0, Se refiere a una nueva etapa en la revolución industrial que se enfoca en gran medida en la interconectividad, el internet de las cosas IoT, el Bigdata, la automatización, el aprendizaje automatizado el análisis de datos en tiempo real entre otros.

Modelo Predictivo, Es un conjunto de procesos ejercidos a través de técnicas matemáticas y estadísticas de análisis de datos, todas ellas llevadas a un lenguaje computacional que ayudan a inferir la probabilidad de que ocurran determinadas situaciones previas a su consecución.

Machine Learning, es un sub conjunto de la inteligencia artificial, que hace referencia a la capacidad que tiene una máquina o software para aprender a tomar decisiones propias mediante un conjunto de algoritmos.