Predicción de fallas en equipos industriales mediante modelos de machine learning usando el dataset AI4I 2020

Yulieth Catalina Castro Molano

Asesor

Sixyel Jeyson Castañeda Coronado

Universidad Nacional Abierta y a Distancia UNAD

Escuela de Ciencias Básicas Tecnología e Ingeniería ECBTI

Especialización en Ciencia de Datos y Analítica

2026

Nota de Aceptación

_____

Nombre Director de Trabajo de Grado

_____ _____

Jurado Jurado

2026

Dedicatoria

Dedico este trabajo de grado a mis padres José Castro y Sonia Molano, por su amor, apoyo incondicional y por ser el pilar fundamental en mi vida. Gracias a su esfuerzo, dedicación y por enseñarme a no rendirme. Este logro también es de ustedes.

A mis sobrinos, Isabella Castro, Mariana Castro y Martín Sánchez, por llenar mi vida de alegría y esperanza. Son mi inspiración y motivación para superarme cada día, con el sueño de poder aportar a su futuro y ser un ejemplo para sus vidas.

A mi hermana, Anggi Castro, por ser mi amiga, mi confidente y mi apoyo constante en cada etapa de este camino. Gracias por estar siempre a mi lado

A mi hermano, Jhon Castro, quien desde el cielo me acompaña y me guía. Gracias por creer en mí

Finalmente, a mi perrita Lulú, por su compañía incondicional y por brindarme momentos de tranquilidad y alegría durante este camino.

Agradecimientos

Agradezco a la Universidad Nacional Abierta y a Distancia, por brindarme la oportunidad de formarme profesionalmente y por los conocimientos adquiridos a lo largo de este proceso académico.

A mi director de trabajo de grado, Sixyel Jeyson Castañeda Coronado, por su orientación, acompañamiento y compromiso, así como por guiarme en el enfoque y desarrollo de este proyecto, aportando sus conocimientos.

A mis compañeros de universidad, por el apoyo brindado, el trabajo en equipo y los aprendizajes compartidos, que enriquecieron significativamente mi proceso de formación.

Finalmente, agradezco a todas las personas que, de una u otra manera, hicieron parte de este proceso y contribuyeron a que hoy este logro sea posible.

Resumen

En la actualidad, la industria manufacturera enfrenta problemas relacionados con la ocurrencia de fallas inesperadas en los equipos de producción, las cuales generan tiempos de inactividad no planificados, incrementos en los costos de mantenimiento y disminución en la eficiencia operativa. En muchos casos, las organizaciones aún adoptan enfoques reactivos, interviniendo los equipos únicamente después de que se presenta la falla, lo que limita la capacidad de anticipación y la toma de decisiones estratégicas basadas en datos. En este contexto, el mantenimiento predictivo surge como una alternativa clave para mejorar la confiabilidad de los procesos industriales mediante el uso de técnicas de analítica de datos y aprendizaje automático.

Este proyecto desarrolló y evaluó un modelo predictivo orientado a la detección temprana de fallas en equipos industriales mediante técnicas de machine learning, empleando el dataset público AI4I 2020, el cual contiene datos relevantes sobre variables operativas de maquinaria industrial, tales como temperatura del aire, temperatura del proceso, velocidad de rotación, torque y desgaste de herramienta, así como indicadores asociados a fallas de los equipos. Este conjunto de datos permite abordar el problema desde un enfoque supervisado, facilitando la construcción de modelos de clasificación capaces de identificar patrones asociados a la ocurrencia de fallas.

La metodología adoptada se basó en el estándar CRISP-DM, ampliamente utilizado en proyectos de ciencia de datos industriales. En la fase de comprensión del negocio (definición de objetivos y KPIs asociados a disponibilidad y eficiencia operativa), comprensión de los datos (análisis exploratorio y verificación de consistencia de etiquetas), preparación (limpieza, transformación, codificación, normalización y selección de variables), modelado (entrenamiento

y comparación de modelos supervisados), evaluación (validación cruzada estratificada y métricas alineadas al costo asimétrico de errores), y despliegue (servicio de inferencia y visualización para soporte decisional). En modelado se implementaron y compararon modelos de regresión logística, Random Forest y XGBoost; se incluyeron técnicas para tratar el desbalance y ajuste de hiperparámetros. En la evaluación se enfatizó el recall para reducir los falsos negativos, es decir, las fallas reales no detectadas, junto con F1-score, accuracy, precision y ROC-AUC.

El desarrollo de este modelo predictivo evidencia el potencial de las técnicas de machine learning para apoyar la optimización de los procesos de mantenimiento, al permitir identificar patrones asociados a fallas y aportar información útil para la toma de decisiones.

Palabras clave**:** Predicción, fallas, análisis, mantenimiento, predictivo

Abstract

Currently, the manufacturing industry faces challenges related to unexpected equipment failures, which lead to unplanned downtime, increased maintenance costs, and decreased operational efficiency. In many cases, organizations still adopt reactive approaches, intervening only after a failure occurs, thus limiting their ability to anticipate problems and make strategic, data-driven decisions. In this context, predictive maintenance emerges as a key alternative for improving the reliability of industrial processes through the use of data analytics and machine learning techniques.

This project developed and evaluated a predictive model for the early detection of industrial equipment failures using machine learning techniques. It employed the public dataset AI4I 2020, which contains relevant data on operational variables of industrial machinery, such as air temperature, process temperature, rotational speed, torque, and tool wear, as well as indicators associated with equipment failures. This dataset allows us to address the problem from a supervised perspective, facilitating the construction of classification models capable of identifying patterns associated with failure occurrences.

The methodology adopted was based on the CRISP-DM standard, widely used in industrial data science projects. The process included the following phases: business understanding (defining objectives and KPIs associated with availability and operational efficiency), data understanding (exploratory analysis and label consistency verification), preparation (cleaning, transformation, coding, normalization, and variable selection), modeling (training and comparison of supervised models), evaluation (stratified cross-validation and metrics aligned with the asymmetric cost of errors), and deployment (inference and visualization service for decision support). Logistic regression, Random Forest, and XGBoost models were

implemented and compared in the modeling phase; techniques for addressing imbalance and hyperparameter adjustment were also included. The evaluation emphasized recall to reduce false negatives—that is, actual failures that went undetected—along with F1-score, accuracy, precision, and ROC-AUC.

The development of this predictive model demonstrates the potential of machine learning techniques to support the optimization of maintenance processes by identifying patterns associated with failures and providing useful information for decision-making.

Keywords**:** Prediction, failures, analysis, maintenance, predictive.

Tabla de Contenido

Lista de Tablas

Lista de Figuras

Introducción

En la actualidad, la industria manufacturera se enfrenta a desafíos significativos relacionados con la continuidad de las operaciones, la eficiencia de los procesos de producción y la reducción de costos asociados al mantenimiento de equipos. Las fallas inesperadas en la maquinaria industrial pueden ocasionar pérdida de recursos, demoras en la producción, paradas no programadas y un impacto económico importante para las empresas. Esta situación es crítica en las pequeñas y medianas empresas, debido a que suelen contar con recursos financieros, técnicos y humanos más limitados.

El mantenimiento predictivo surgió como una alternativa para anticipar posibles fallas mediante el análisis de datos provenientes de equipos de piso de planta, como sensores, registros operativos y variables de funcionamiento de los equipos. Estos datos al ser procesados de forma correcta permiten convertirse en información valiosa para identificar patrones, detectar anomalías y apoyar la toma de decisiones en mantenimiento industrial.

Este proyecto desarrolló y evaluó un modelo predictivo basado en técnicas de machine learning, orientado a la detección temprana de fallas en equipos industriales a partir del dataset público AI4I 2020. Este conjunto de datos, disponible en Kaggle, contiene 10.000 registros sintéticos que representan condiciones de mantenimiento predictivo en un entorno industrial. Incluye variables operativas como tipo de producto, temperatura del aire, temperatura del proceso, velocidad de rotación, torque y desgaste de herramienta, además de etiquetas asociadas a la ocurrencia de fallas y a diferentes modos de falla.

La metodología utilizada fue CRISP-DM, la cual permite organizar el proceso de ciencia de datos en fases estructuradas: comprensión del negocio, comprensión de los datos, preparación de los datos, modelado, evaluación y despliegue. En este trabajo se realizó el análisis

exploratorio del dataset, el entrenamiento de modelos de clasificación y evaluación del desempeño mediante métricas adecuadas para problemas de predicción de fallas, tales como recall, precision, F1-score, accuracy y matriz de confusión. Dado que en este tipo de problemas suele existir desbalance entre registros con falla y sin falla, se dio especial importancia al recall, ya que en mantenimiento predictivo es más crítico no detectar una falla real que generar una falsa alarma

Justificación

El proyecto se justificó en la necesidad del sector industrial de fortalecer la capacidad de anticipar fallas en sus equipos, debido a que esta problemática afecta directamente la eficiencia operativa, los costos de mantenimiento y la ejecución de los procesos productivos. En la actualidad, muchas organizaciones, especialmente las PYMES (Pequeñas y Medianas Empresas), continúan trabajando con estrategias de mantenimiento reactivo o preventivo, las cuales pueden resultar insuficientes frente a la complejidad de los sistemas de producción modernos.

Este proyecto se enfocó en la aplicación de técnicas de machine learning para resolver un problema real del sector industrial. Su desarrollo permitió integrar conocimientos de análisis de datos, modelado predictivo y evaluación de algoritmos, contribuyendo a la construcción de soluciones basadas en evidencia para apoyar la toma de decisiones en mantenimiento.

La predicción de fallas en equipos industriales mediante modelos de machine learning aportó al desarrollo de competencias investigativas y técnicas, al permitir la aplicación de una metodología reconocida como CRISP-DM en un caso práctico. También fortaleció la capacidad de interpretar datos industriales y transformarlos en información útil para la gestión operativa. La implementación de modelos predictivos puede contribuir a la mejora de la productividad empresarial, al reducir pérdidas asociadas a paradas no programadas y optimizar el uso de recursos.

Objetivos

Objetivo General

Desarrollar y evaluar un modelo predictivo basado en técnicas de machine learning para la detección temprana de fallas en equipos industriales utilizando el dataset AI4I 2020.

Objetivos Específicos

Analizar y comprender el comportamiento de las variables operativas del dataset AI4I 2020 mediante análisis exploratorio de datos

Preparar y transformar los datos mediante técnicas de limpieza, normalización y selección de variables.

Implementar modelos de clasificación como Random Forest, XGBoost y regresión logística.

Evaluar el desempeño de los modelos utilizando métricas como accuracy, recall y F1-score.

Planteamiento del Problema

La industria manufacturera actual enfrenta un problema crítico relacionado con la ocurrencia de fallas inesperadas en equipos industriales, las cuales generan tiempos de inactividad no planificados, incrementos en los costos de mantenimiento y pérdidas en la productividad. Estas fallas afectan directamente la eficiencia operativa y la competitividad de las organizaciones, especialmente en las MiPymes, donde los recursos son limitados.

En muchos casos, las empresas continúan utilizando estrategias de mantenimiento reactivo o preventivo, que no permiten anticipar de manera efectiva los fallos en los equipos. Esto conlleva a intervenciones tardías, afectando la continuidad de los procesos productivos.

Con el avance de la Industria 4.0, se ha incrementado la disponibilidad de datos provenientes de sensores industriales; sin embargo, estos datos no siempre son aprovechados adecuadamente para generar valor. Surge entonces la necesidad de implementar modelos predictivos que permitan identificar patrones asociados a fallas y anticiparse a su ocurrencia.

Marco de Referencia

Marco Teórico

La predicción de fallas en equipos industriales se centra en la necesidad de mejorar la confiabilidad, disponibilidad y eficiencia de los procesos productivos de la industria mediante el uso de datos. En los procesos industriales modernos, las fallas inesperadas pueden generar tiempos de inactividad no planificados, pérdidas económicas, disminución de la productividad y aumento de los costos de mantenimiento. Por esta razón, el mantenimiento predictivo se ha consolidado como una estrategia orientada a anticipar fallas a partir del análisis de variables operativas y del comportamiento histórico de los equipos.

En este contexto, la Industria 4.0 ha impulsado la incorporación de sensores, sistemas ciberfísicos, Internet de las Cosas Industrial, analítica avanzada e inteligencia artificial en los procesos productivos. Permitiendo monitorear continuamente el estado de las máquinas en tiempo real, recolectar datos operativos y sincronizar información entre el entorno físico de producción y los sistemas digitales de análisis. (*Lee, Bagheri y Kao (2015)* explican que los sistemas ciberfísicos en manufactura permiten monitorear y sincronizar información entre la planta física y el entorno informático, facilitando operaciones más eficientes, colaborativas y resilientes.

Marco Conceptual

Análisis de Datos y Aprendizaje Automático en la Industria

El análisis de datos y el aprendizaje automático están emergiendo como herramientas poderosas para mejorar la eficiencia operativa en las industrias. El análisis de datos permite a las empresas extraer información valiosa de grandes volúmenes de datos generados por los equipos de producción. Mediante técnicas de análisis estadístico y minería de datos, es posible identificar

patrones, tendencias y anomalías en los datos que pueden ayudar a mejorar el rendimiento operativo. El aprendizaje automático permite a las empresas construir modelos predictivos y prescriptivos basados en datos históricos y en tiempo real. Estos modelos pueden utilizarse para predecir fallos en los equipos, optimizar los parámetros de producción y tomar decisiones informadas para mejorar la eficiencia operativa.

Disponibilidad

Es el tiempo que el equipo está disponible y listo para operar durante el tiempo de producción programado. Se calcula dividiendo el tiempo de funcionamiento real del dispositivo por el tiempo total de programación. La disponibilidad se ve afectada por varios tipos de pérdida de tiempo, como el tiempo de inactividad no planificado debido a averías, el tiempo de preparación y ajuste y los tiempos de cambio de turno.

Eficiencia Operativa y Competitividad

La eficiencia operativa se define como la capacidad de una empresa para maximizar la producción utilizando de manera óptima sus recursos. En las industrias, la eficiencia operativa está estrechamente relacionada con la competitividad en el mercado global. Mejorar la eficiencia operativa permite a las empresas reducir costos, aumentar la productividad y mejorar la calidad de sus productos, lo que les otorga una ventaja competitiva significativa.

Industria 4.0

Es conocida como la integración de Tecnologías en procesos industriales conocido como el internet de las cosas o IoT, la inteligencia artificial y el análisis de grandes volúmenes de datos. Lo que permite la automatización de procesos o la conexión entre dispositivos y la recolección de datos en tiempo real, facilitando la toma de decisiones basada en datos.

Machine Learning

Es una rama de la inteligencia artificial que permite a sistemas aprender a partir de datos, a través de algoritmos, los modelos identifican patrones en los datos y generan predicciones. En el contexto industrial, se utiliza para predecir fallas, optimizar procesos y mejorar la toma de decisiones.

Mantenimiento Predictivo

Es la estrategia de usar datos en tiempo real para predecir fallas en los equipos antes de que estas ocurran. A diferencia del mantenimiento correctivo o preventivo que se enfocan en intervenir los equipos cuando ya existe un deterioro, optimizando recursos y reduciendo tiempos de inactividad

Regresión Logística

Es un algoritmo de clasificación supervisada utilizado para estimar la probabilidad de ocurrencia de un evento binario. su salida se encuentra limitada entre 0 y 1 mediante la función sigmoide, permitiendo interpretar el resultado como una probabilidad.

Ej. Predecir si una máquina presentará falla: falla / no falla. Se obtiene mediante la función logística o sigmoide

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0+\beta_1 X_1+\beta_2 X_2 \ldots \beta_N X_N)}}$$

Donde:

*P*representa la probabilidad estimada de que ocurra una falla.

e: Base del logaritmo natural (aproximadamente 2.718)

β_0= Intersección (ordenada al origen).

$\beta_1, \beta_2, \ldots, \beta_N$= Coeficientes o pesos de cada variable independiente.

$X_1, X_2, \ldots, X_N$=Variables independientes (predictores).

Random Forest

Es un algoritmo de aprendizaje supervisado basado en la construcción de múltiples árboles de decisión. Cada árbol se entrena utilizando una muestra aleatoria de los datos y un subconjunto aleatorio de variables. Ej en el proyecto Predecir si una máquina presentará falla: falla / no falla. La predicción de Random Forest puede representarse mediante:

$$\hat{y}(x) = moda\{T_1(x), T_2(x), \dots, T_B(x)\}$$

$\hat{y}(x)$= Representa la predicción final del modelo.

$T_B(x)$= corresponde a la predicción realizada por el árbol b.

B= Representa el número total de árboles construidos.

$moda$ = indica la clase que obtiene la mayoría de votos.

XGBoost

Es un algoritmo de ensamble que construye árboles de decisión de manera secuencial. Cada nuevo árbol intenta corregir los errores cometidos por los árboles anteriores. Esta característica permite alcanzar altos niveles de desempeño predictivo en problemas con datos estructurados o tabulares. La predicción de XGBoost se representa como la suma de las contribuciones de los árboles construidos:

$$\hat{\gamma}_i = \sum_{K=1}^{K} f_k(x_i)$$

$\hat{\gamma}_i$representa la predicción para la observación i.

K corresponde al número total de árboles.

f_k representa el árbol de decisión construido en la iteración k.

x_i corresponde al conjunto de variables predictoras de la observación i.

Métricas de Evaluación

Permiten cuantificar el desempeño de los modelos de clasificación implementados. En el contexto del mantenimiento predictivo, no basta con identificar correctamente los equipos que operan normalmente; también es fundamental detectar la mayor cantidad posible de fallas reales, una falla no identificada puede generar paradas no programadas, pérdidas económicas y daños en los equipos.

Matriz de Confusión

Las matrices de confusión permiten identificar los tipos de errores generados por el modelo. En mantenimiento predictivo, un falso positivo puede ocasionar una inspección innecesaria, mientras que un falso negativo puede permitir que una falla ocurra sin una alerta previa. Por esta razón, el análisis de la matriz de confusión es indispensable para seleccionar el modelo más adecuado desde una perspectiva operativa. Para interpretar las métricas utilizadas, se definen los siguientes elementos

Verdadero positivo (VP): equipo que realmente presenta falla y el modelo clasifica correctamente como falla.

Verdadero negativo (VN): equipo que no presenta falla y el modelo clasifica correctamente como no falla.

Falso positivo (FP): equipo que no presenta falla, pero el modelo lo clasifica incorrectamente como falla.

Falso negativo (FN): equipo que realmente presenta falla, pero el modelo lo clasifica incorrectamente como no falla.

Accuracy

Mide la proporción total de predicciones correctas realizadas por el modelo, considerando tanto los equipos con falla como los equipos sin falla. Su fórmula es:

$$Acurracy = \frac{VP + VN}{VP + VN + FP + FN}$$

Recall

Mide la proporción de fallas reales que fueron correctamente identificadas por el modelo. Su fórmula es:

$$Recall = \frac{VP}{VP + FN}$$

Precision

Mide la proporción de predicciones de falla que realmente correspondían a una falla. Su fórmula es:

$$Precision = \frac{VP}{VP + FP}$$

F1-Score

Es una métrica que combina precision y recall mediante su media armónica. Es especialmente útil cuando existe desbalance entre las clases y se requiere encontrar un equilibrio entre detectar fallas reales y evitar falsas alarmas. Su fórmula es:

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

ROC-AUC

Representa el comportamiento del modelo al comparar la tasa de verdaderos positivos frente a la tasa de falsos positivos para distintos umbrales de clasificación.

La tasa de verdaderos positivos(TPR) corresponde al *recall*:

$$TPR = \frac{VP}{VP + FN}$$

La tasa de falsos positivos(FPR) se calcula como:

$$FPR = \frac{FP}{FP + VN}$$

GridSearchCV

Es una herramienta de la librería Scikit-Learn en Python diseñada para automatizar y optimizar la selección de hiperparámetros de un modelo de aprendizaje automático mediante validación cruzada. Es una técnica utilizada para realizar la búsqueda sistemática de la mejor combinación de hiperparámetros de un modelo. Los hiperparámetros son valores definidos antes del entrenamiento y controlan el comportamiento del algoritmo,

Cross-Validation(Validación Cruzada)

Es una técnica utilizada para evaluar la capacidad de generalización de un modelo. En lugar de depender únicamente de una división fija entre entrenamiento y prueba, los datos se dividen en varios subconjuntos o pliegues. En cada iteración, una parte se utiliza para validar el modelo y las restantes para entrenarlo.

En una validación cruzada de K pliegues, la métrica promedio se calcula como:

$$\underline{M}_{CV} = \frac{1}{K}\sum_{K=1}^{K} M_K$$

K= Representa el número total de pliegues.

M_K= Corresponde a la métrica obtenida en el pliegue k.

$\underline{M}_{CV}$= Representa el desempeño promedio del modelo.

Metodología

La metodología CRISP-DM (Cross-Industry Standard Process for Data Mining) es un marco estructurado que guía los proyectos de minería de datos desde el inicio hasta la implementación. De acuerdo con el estudio de Haya(2022) al ser una metodología orientada a procesos que siguen una trayectoria bastante secuencial, facilita el entendimiento por parte del cliente de la planificación y ejecución del proyecto. Esta metodología se divide en seis fases interrelacionadas que cubren todo el ciclo de vida del proyecto. La primera etapa de "comprensión empresarial" se centra en comprender los objetivos comerciales y los requisitos del proyecto. Lo siguiente es la comprensión de la fase de datos, donde los datos disponibles se evalúan y exploran para determinar si son adecuados para el análisis. La etapa de preparación de datos implica limpiar, integrar y transformar los datos para su posterior análisis. Luego, la fase de modelado construye modelos analíticos para identificar patrones y tendencias en los datos. Estos modelos se evalúan durante la fase de evaluación para asegurar su calidad y validez. Finalmente, durante la fase de implementación, los resultados del análisis se implementan en la organización, lo que puede incluir la integración del modelo en los sistemas existentes y la capacitación de los empleados en su uso.

- Comprensión del negocio: Definición del problema de predicción de fallas y KPIs
- Comprensión de los datos: Análisis exploratorio del dataset AI4I 2020.
- Preparación de datos: Limpieza, tratamiento de valores nulos, normalización y selección de variables.
- Modelado: Implementación de modelos de clasificación
- Evaluación: Uso de métricas como recall (prioritario), accuracy y F1-score.

- Despliegue: Uso del modelo y documentación de resultados, dejando como trabajo futuro su integración en un sistema real de monitoreo industrial

Desarrollo del Proyecto

El dataset AI4I 2020 fue utilizado como fuente de información para el desarrollo del modelo predictivo. Este conjunto de datos contiene 10.000 registros relacionados con condiciones operativas de equipos industriales, incluyendo variables como temperatura del aire, temperatura del proceso, velocidad de rotación, torque, desgaste de herramienta y la variable objetivo Machine failure, la cual indica si se presentó o no una falla en el equipo. Este conjunto de datos permite abordar el problema desde un enfoque de aprendizaje supervisado, específicamente como una tarea de clasificación binaria.

Figura 1

Dimensiones y Primeras Filas

```
Dimensiones del dataset: (10000, 14)

Primeras filas:
   UDI Product ID Type  Air temperature [K]  Process temperature [K]  \
0    1     M14860    M                298.1                    308.6
1    2     L47181    L                298.2                    308.7
2    3     L47182    L                298.1                    308.5
3    4     L47183    L                298.2                    308.6
4    5     L47184    L                298.2                    308.7

   Rotational speed [rpm]  Torque [Nm]  Tool wear [min]  Machine failure  TWF  \
0                    1551         42.8                0                0    0
1                    1408         46.3                3                0    0
2                    1498         49.4                5                0    0
3                    1433         39.5                7                0    0
4                    1408         40.0                9                0    0

   HDF  PWF  OSF  RNF
0    0    0    0    0
1    0    0    0    0
2    0    0    0    0
3    0    0    0    0
4    0    0    0    0
```

Nota. Se realizo una consulta de las dimensiones y primeras filas y el resultado fue que las dimensiones del dataset: (10000, 14)

Figura 2

Información General del Dataset

```
Información general:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 14 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   UDI                     10000 non-null  int64
 1   Product ID              10000 non-null  object
 2   Type                    10000 non-null  object
 3   Air temperature [K]     10000 non-null  float64
 4   Process temperature [K] 10000 non-null  float64
 5   Rotational speed [rpm]  10000 non-null  int64
 6   Torque [Nm]             10000 non-null  float64
 7   Tool wear [min]         10000 non-null  int64
 8   Machine failure         10000 non-null  int64
 9   TWF                     10000 non-null  int64
 10  HDF                     10000 non-null  int64
 11  PWF                     10000 non-null  int64
 12  OSF                     10000 non-null  int64
 13  RNF                     10000 non-null  int64
dtypes: float64(3), int64(9), object(2)
memory usage: 1.1+ MB
None
```

Nota. Visualizamos el tipo de dato de cada columna y si hay valores nulos. Se evidencia que no hay valores nulos, porque todas las columnas tienen 10.000 registros completos.

Durante el análisis exploratorio se identificó que el dataset no presenta valores nulos, lo cual facilita el proceso de preparación de los datos. Sin embargo, se evidenció un fuerte desbalance en la variable objetivo, dado que el 96,61% de los registros corresponde a equipos sin falla, mientras que solo el 3,39% representa eventos de falla. Esta característica es común en problemas de mantenimiento predictivo, ya que las fallas suelen ser eventos poco frecuentes, pero de alto impacto operativo.

Figura 3

Consulta de Valores Nulos

```
print("\nValores nulos:")
print(df.isnull().sum())
```

```
Valores nulos:
UDI                        0
Product ID                 0
Type                       0
Air temperature [K]        0
Process temperature [K]    0
Rotational speed [rpm]     0
Torque [Nm]                0
Tool wear [min]            0
Machine failure            0
TWF                        0
HDF                        0
PWF                        0
OSF                        0
RNF                        0
dtype: int64
```

Nota. En la consulta se evidenció que no hay valores nulos, porque todas las columnas tienen 10.000 registros completos.

Figura 4

Distribución de las Variables

```
# Distribución de la variable objetivo
target_col = "Machine failure"
```

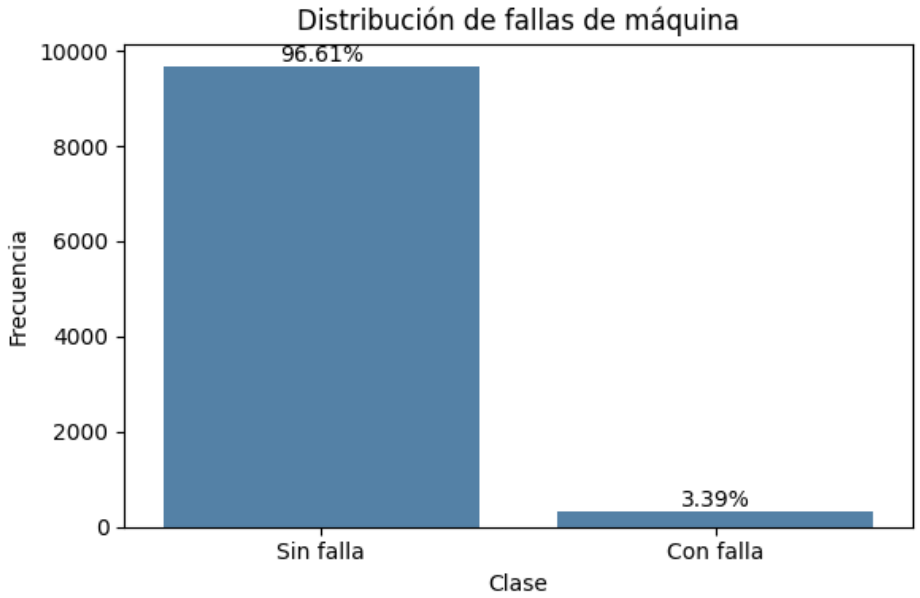
```
print("\nDistribución de la variable objetivo:")
print(df[target_col].value_counts())
print("\nDistribución porcentual:")
print(df[target_col].value_counts(normalize=True) * 100)
```

```
Distribución de la variable objetivo:
Machine failure
0    9661
1     339
Name: count, dtype: int64

Distribución porcentual:
Machine failure
0    96.61
1     3.39
Name: proportion, dtype: float64
```

Nota. La variable Machine failure es la variable dependiente, acá revisamos la distribución se identifica que la clase 0 corresponde a máquinas sin falla es del 96.61% y la clase 1 que son máquinas con falla de un 3.39%

Figura 5

Grafica Distribución de Fallas

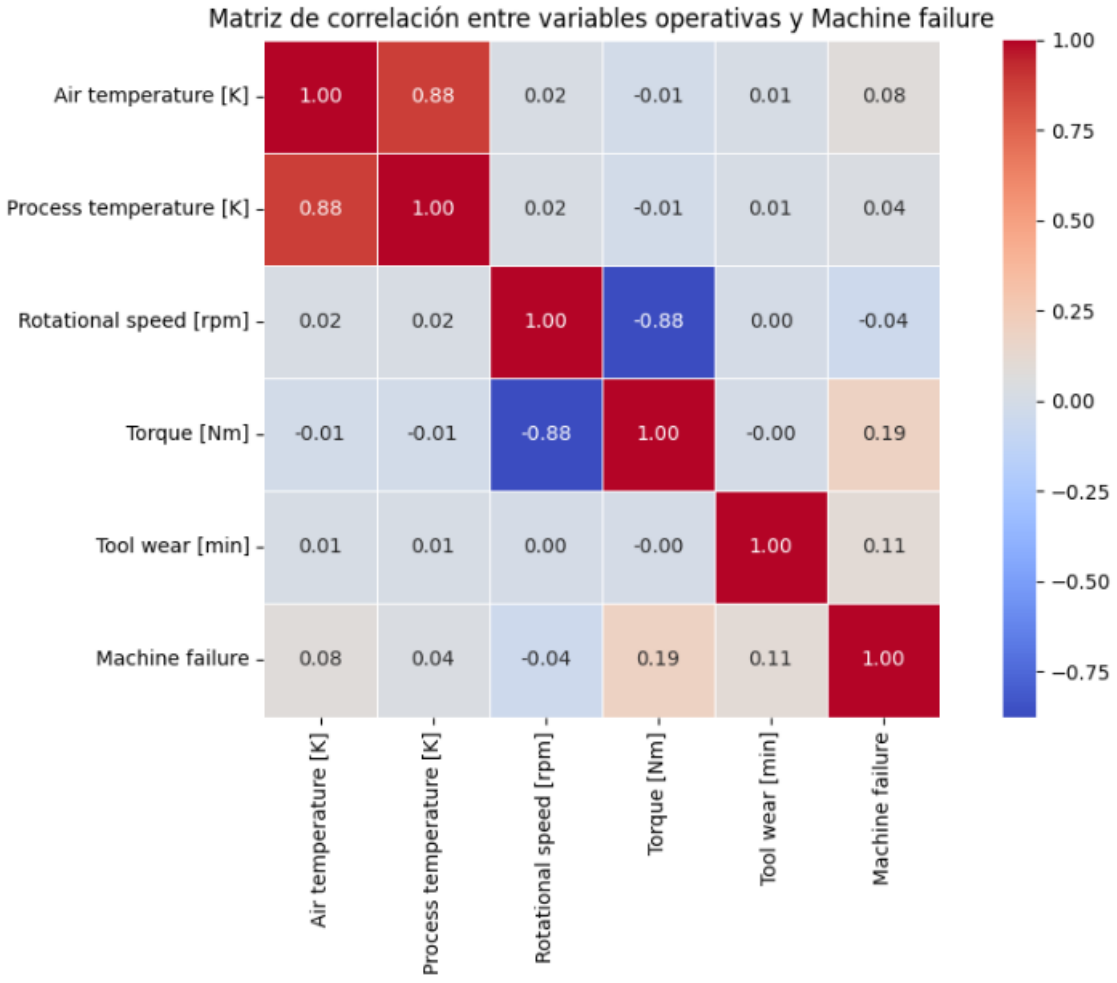


Nota. Este gráfico muestra visualmente que hay muchos más registros de no falla que de falla

Como complemento al análisis exploratorio, se profundizó en la descripción de las variables del dataset AI4I 2020, el comportamiento estadístico de las variables operativas, la comparación entre equipos con falla y sin falla, la relación entre variables mediante correlación y gráficos de dispersión, así como la revisión de los tipos específicos de falla disponibles en el conjunto de datos. Estos análisis permiten justificar las decisiones posteriores de preparación de datos y selección de modelos, especialmente porque el problema presenta un fuerte desbalance de clases y requiere priorizar la detección de fallas reales.

Figura 6

Matriz de Correlación



	Variable	Correlación con Machine failure
0	Torque [Nm]	0.191
1	Tool wear [min]	0.105
2	Air temperature [K]	0.083
3	Rotational speed [rpm]	-0.044
4	Process temperature [K]	0.036

Nota. La matriz de correlación permite revisar relaciones lineales iniciales entre las variables. Aunque la ocurrencia de fallas puede depender de patrones no lineales, este análisis permite justificar el uso de modelos como Random Forest y XGBoost.

Figura 7

Diccionario de Variables del Dataset AI4I 2020

	Variable	Descripción
0	UDI	Identificador único del registro. No aporta información técnica para la predicción.
1	Product ID	Identificador del producto. Debe excluirse porque funciona como código y no como condición operativa.
2	Type	Tipo o calidad del producto/máquina. Variable categórica que puede codificarse para el modelo.
3	Air temperature [K]	Temperatura ambiente o del aire medida en Kelvin.
4	Process temperature [K]	Temperatura del proceso medida en Kelvin.
5	Rotational speed [rpm]	Velocidad de rotación del equipo en revoluciones por minuto.
6	Torque [Nm]	Esfuerzo o carga mecánica aplicada al eje, expresada en Newton-metro.
7	Tool wear [min]	Desgaste acumulado de la herramienta en minutos.
8	Machine failure	Variable objetivo: 0 indica equipo sin falla y 1 indica equipo con falla.
9	TWF	Tool Wear Failure: falla asociada al desgaste de herramienta.
10	HDF	Heat Dissipation Failure: falla asociada a disipación de calor.
11	PWF	Power Failure: falla asociada a potencia mecánica anormal.
12	OSF	Overstrain Failure: falla asociada a sobreesfuerzo.
13	RNF	Random Failure: falla aleatoria.

Nota. Esta tabla permite diferenciar las variables operativas, la variable objetivo y las columnas que no deben ser usadas como predictoras para evitar fuga de información.

Figura 8

Resumen Estadístico de Variables Operativas

	Air temperature [K]						Process temperature [K]				...	Torque [Nm]				Tool wear [min]					
	count	mean	median	std	min	max	count	mean	median	std	...	median	std	min	max	count	mean	median	std	min	max
Machine failure																					
0	9661	299.974	300.0	1.991	295.3	304.5	9661	309.996	310.0	1.487	...	39.9	9.472	12.6	70.0	9661	106.694	107.0	62.946	0	246
1	339	300.886	301.6	2.071	295.6	304.4	339	310.290	310.4	1.364	...	53.7	16.374	3.8	76.6	339	143.782	165.0	72.760	0	253

Nota. La tabla presenta el resumen estadístico de las principales variables operativas del dataset AI4I 2020, agrupadas según la variable objetivo Machine failure.

Figura 9

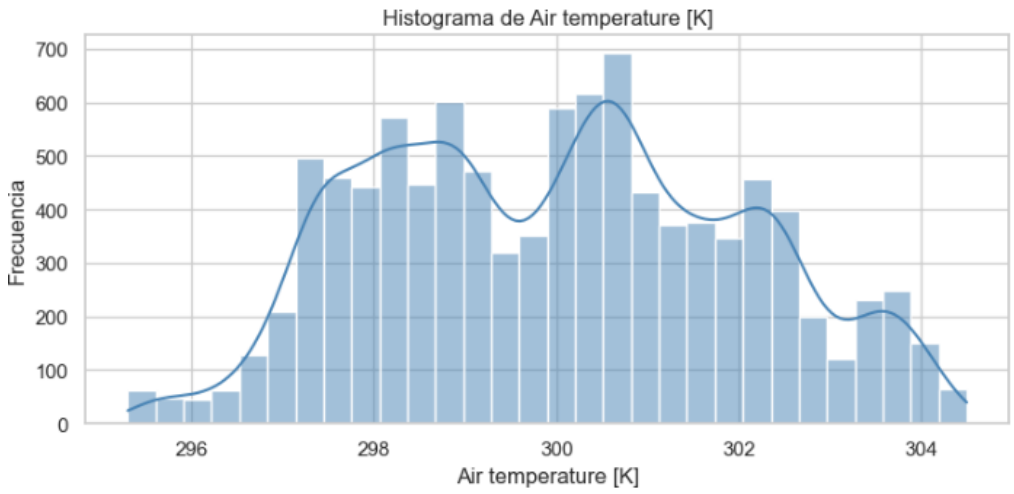
Comparación Estadística Entre Registros sin Falla y con Falla

	Variable	Media sin falla	Media con falla	Diferencia de medias	Mediana sin falla	Mediana con falla	Diferencia de medianas
0	Air temperature [K]	299.974	300.886	0.912	300.0	301.6	1.6
1	Process temperature [K]	309.996	310.290	0.295	310.0	310.4	0.4
2	Rotational speed [rpm]	1540.260	1496.487	-43.773	1507.0	1365.0	-142.0
3	Torque [Nm]	39.630	50.168	10.538	39.9	53.7	13.8
4	Tool wear [min]	106.694	143.782	37.088	107.0	165.0	58.0

Nota. La comparación evidencia diferencias entre las condiciones normales y las condiciones asociadas a falla, especialmente en torque, velocidad de rotación y desgaste de herramienta.

Figura 10

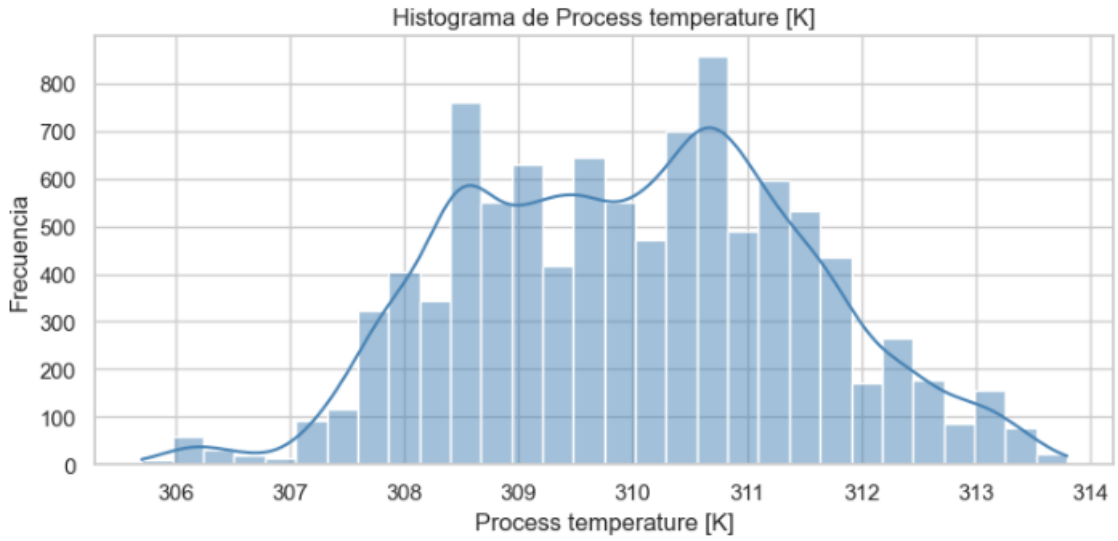
Histogramas por Frecuencia Air Temperature[k]



Nota. La variable Air temperature [K] muestra una distribución concentrada aproximadamente entre 295 K y 304 K. La mayor cantidad de registros se encuentra entre 297 K y 302 K, lo que indica que la temperatura ambiente del entorno de operación se mantiene dentro de un rango relativamente estable.

Figura 11

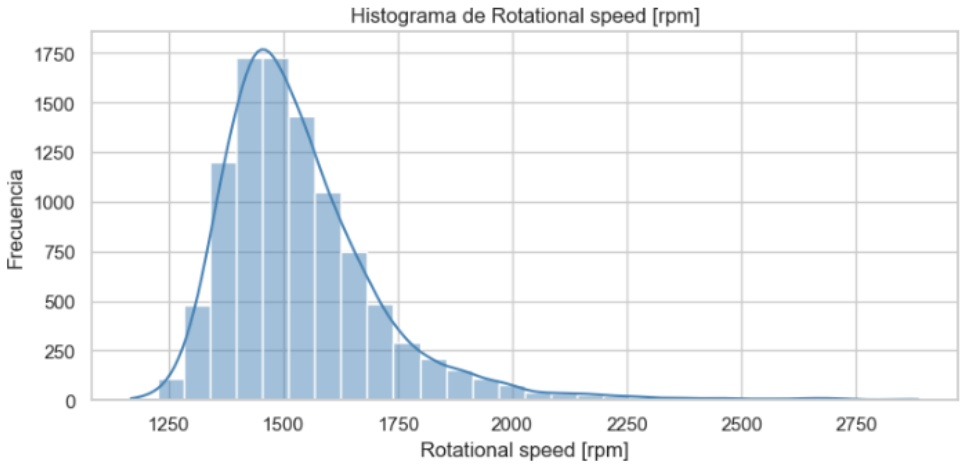
Histogramas por Frecuencia de Process Temperatura[k]



Nota. La variable Process temperature [K] se distribuye aproximadamente entre 306 K y 314 K, con mayor concentración entre 308 K y 311 K.

Figura 12

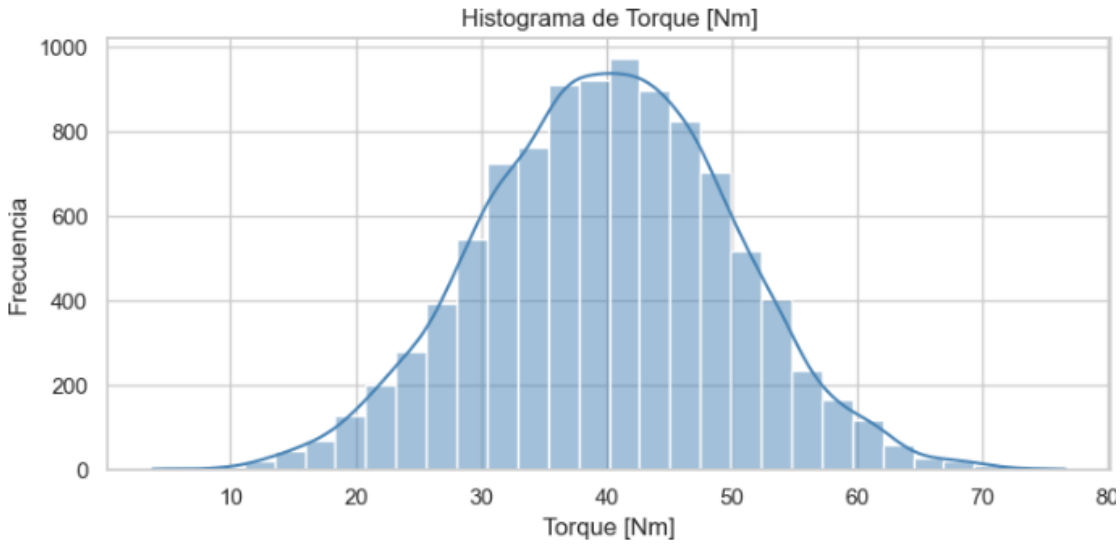
Histogramas por Frecuencia de Rotational Speed[rpm]



Nota. Presenta una distribución claramente asimétrica hacia la derecha. La mayoría de los registros se concentra entre 1300 rpm y 1700 rpm, con un pico cercano a 1450 rpm.

Figura 13

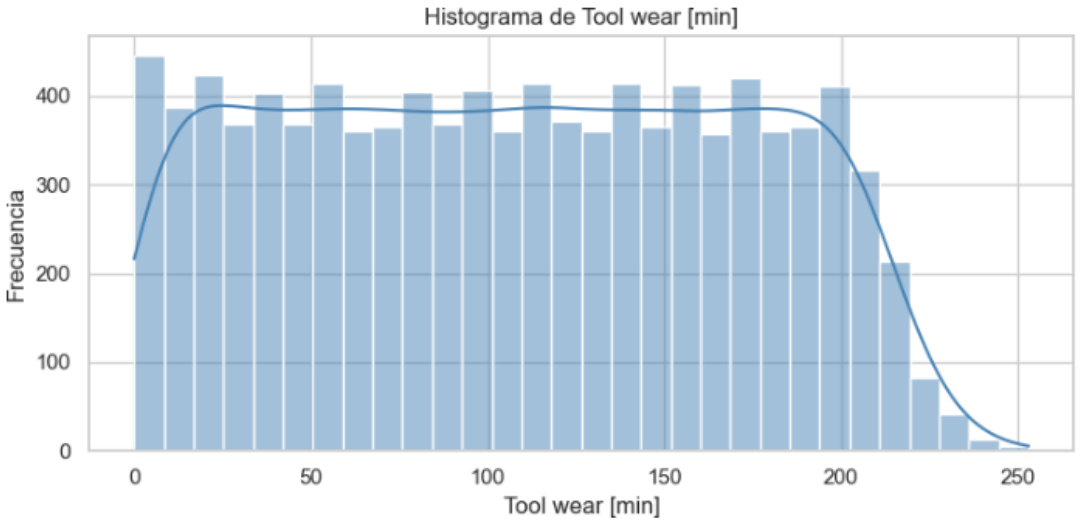
Histogramas por Frecuencia de Torque [Nm]



Nota. Presenta una distribución cercana a una forma normal o campana, concentrada principalmente entre 30 Nm y 50 Nm, con un pico alrededor de 40 Nm.

Figura 14

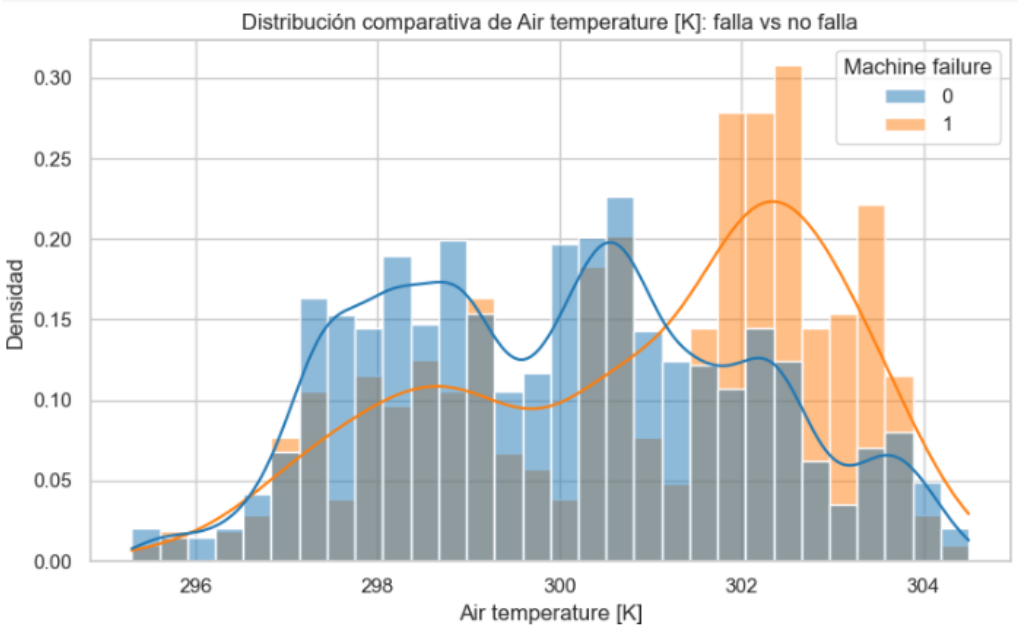
Histogramas por Frecuencia de Tool Wear[min]



Nota. Muestra una distribución bastante uniforme entre 0 y 200 minutos

Figura 15

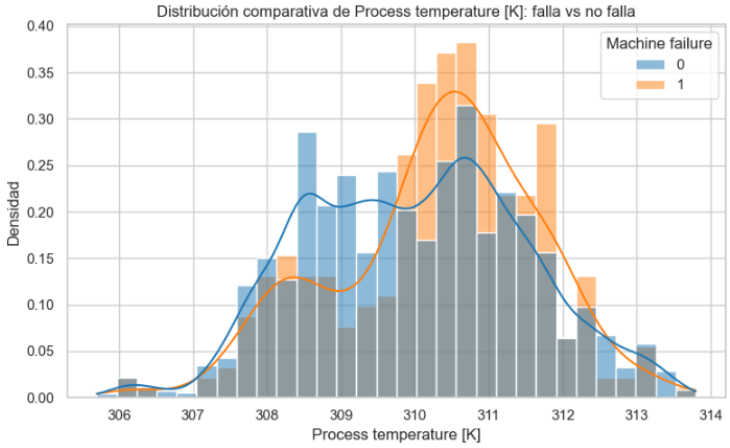
Histogramas Comparativos con Densidad Falla vs No Falla Air Temperature[k]



Nota. Los registros con falla tienden a concentrarse en rangos ligeramente superiores, especialmente alrededor de 302 K y 303 K

Figura 16

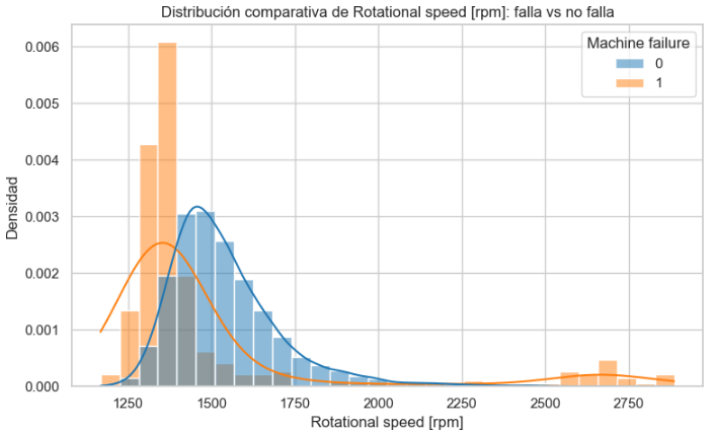
Histogramas Comparativos con densidad Falla vs No Falla de Process Temperatura[k]



Nota. Muestra una concentración importante de registros con falla entre 310 K y 311.5 K.

Figura 17

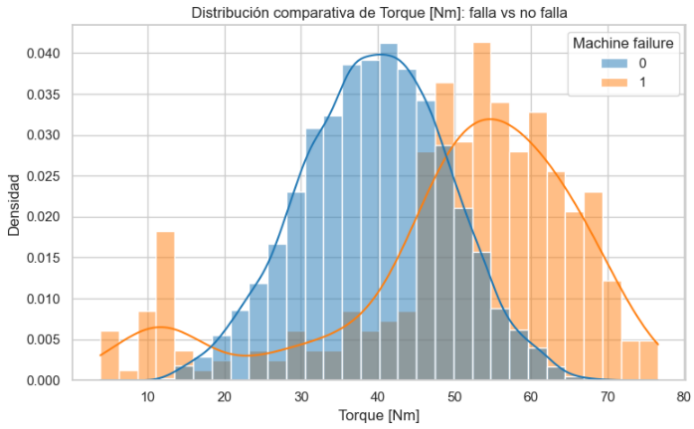
Histogramas Comparativos con Densidad Falla vs No Falla de Rotational Speed[rpm]



Nota. Muestra una diferencia más marcada entre máquinas con falla y sin falla. los registros con falla aparecen con mayor intensidad en velocidades más bajas, cerca de 1250 rpm a 1400 rpm, y también en algunos valores muy altos, por encima de 2500 rpm.

Figura 18

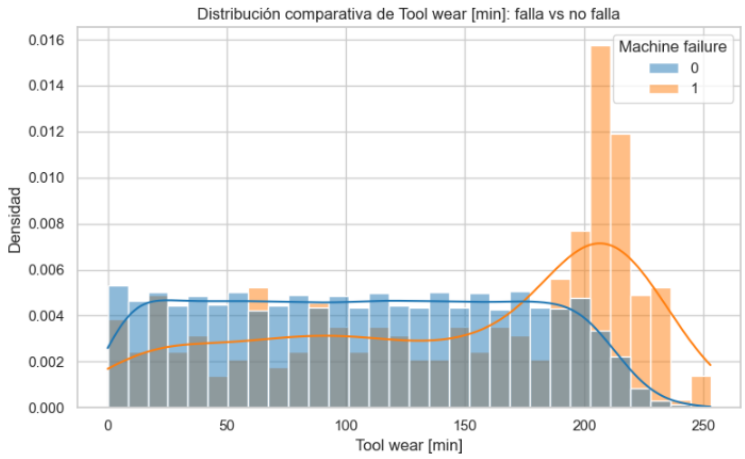
Histogramas Comparativos con Densidad Falla vs No Falla de Torque [Nm]



Nota Los registros con falla se concentran principalmente en valores más altos, aproximadamente entre 50 Nm y 70 Nm. También se observan algunos casos de falla con torque muy bajo, cerca de 10 Nm.

Figura 19

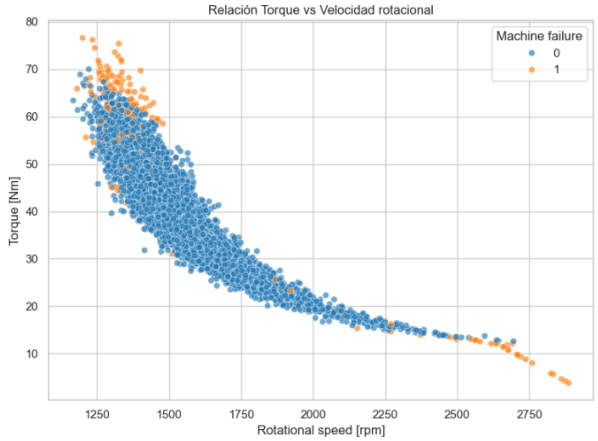
Histogramas Comparativos con Densidad Falla vs No Falla de Tool Wear [min]



Nota. Los registros con falla se concentran principalmente en valores altos, especialmente entre 190 y 230 minutos.

Figura 20

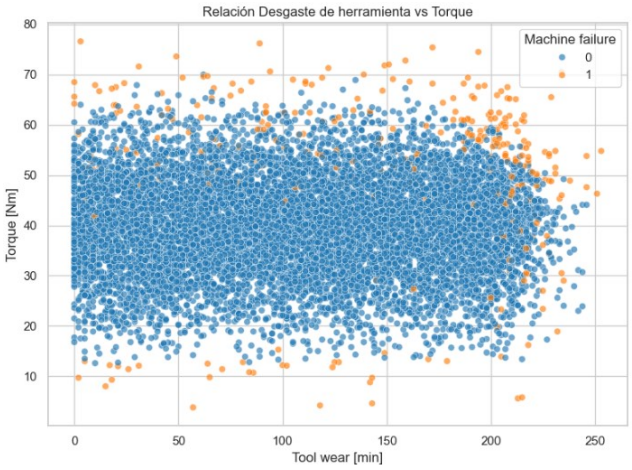
Gráficos de Dispersión Relación Torque vs Velocidad Rotacional.



Nota. A medida que aumenta la velocidad, el torque disminuye, muchas fallas aparecen cuando el torque es alto y la velocidad es baja, lo que indica una condición de esfuerzo mecánico importante.

Figura 21

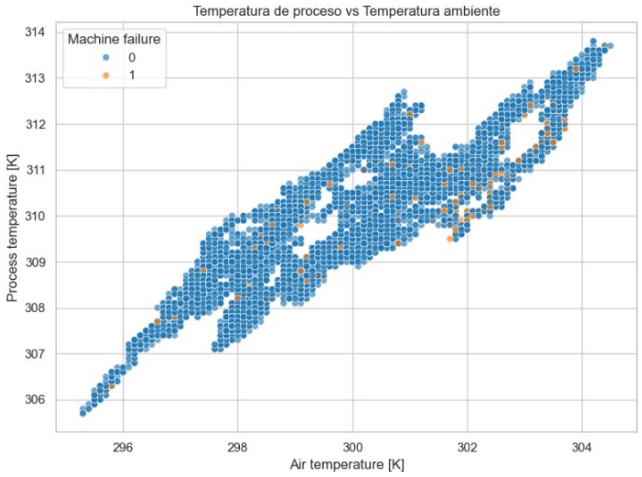
Gráficos de Dispersión Relación Desgaste de Herramienta vs Torque



Nota. No hay una relación directa muy marcada entre desgaste y torque, pero las fallas tienden a aparecer más cuando el torque es alto y cuando el desgaste también es elevado.

Figura 22

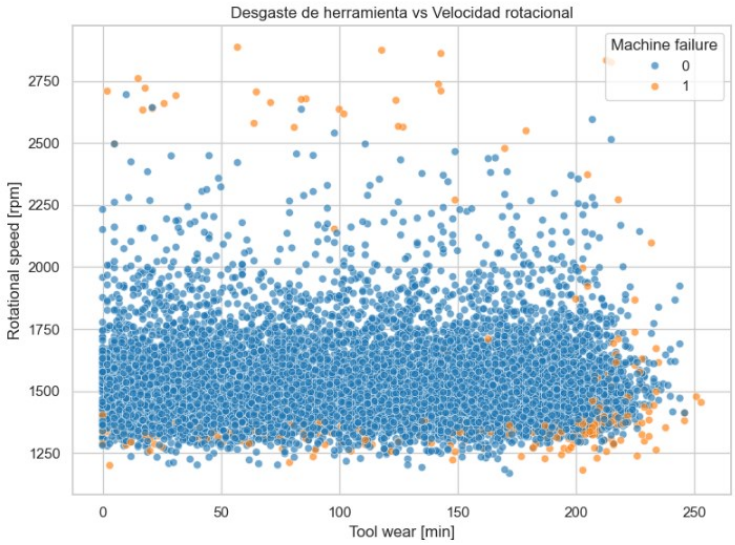
Gráficos de Dispersión Temperatura de Proceso vs Temperatura Ambiente



Nota. Las dos temperaturas están altamente relacionadas. Cuando una aumenta, la otra también. Esto tiene sentido operacional y puede ayudar al modelo a identificar condiciones térmicas asociadas a falla

Figura 23

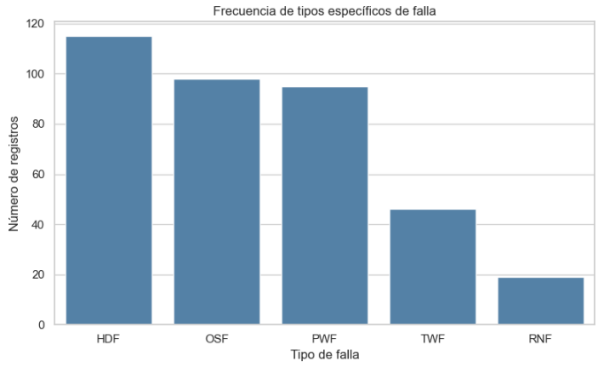
Gráficos de Dispersión Desgaste de Herramienta vs Velocidad Rotacional



Nota. La velocidad no cambia de forma clara según el desgaste, pero las fallas aparecen más cuando el desgaste de la herramienta es alto.

Figura 24

Gráfico de Frecuencia de Tipos Específicos de Falla



Machine failure	TWF	HDF	PWF	OSF	RNF
0	0	0	0	0	18
1	46	115	95	98	1

Nota. Aunque estas columnas no se usaron para entrenar el modelo, su revisión exploratoria permite comprender la naturaleza de los eventos de falla presentes en el dataset

En la fase de preparación de datos se eliminaron las columnas UDI y Product ID, debido a que corresponden a identificadores y no aportan información predictiva relevante. También, se eliminaron las columnas TWF, HDF, PWF, OSF y RNF, asociadas a tipos específicos de falla, con el fin de evitar fuga de información. La fuga de información ocurre cuando el modelo utiliza variables que no estarían disponibles antes de realizar la predicción, lo cual puede generar resultados artificialmente altos y poco realistas.

Posteriormente, se codificó la variable categórica Type, se separaron las variables predictoras de la variable objetivo y se dividió el conjunto de datos en entrenamiento y prueba, utilizando una división estratificada para conservar la proporción de fallas y no fallas en ambos subconjuntos

Figura 25

Eliminación de Columnas Identificadoras

```
# Eliminar columnas identificadoras que no aportan al modelo
drop_cols = []
for col in ["UDI", "Product ID"]:
    if col in df.columns:
        drop_cols.append(col)
df_model = df.drop(columns=drop_cols, errors="ignore").copy()
```

Nota. UDI y Product ID no aportan información técnica útil al modelo, porque son identificadores por lo que se eliminan

Figura 26

Definición de Variable

```
# Variable dependiente
y = df_model[target_col]
X = df_model.drop(columns=[target_col], errors="ignore")
```

Nota. Y contiene la variable objetivo: si hubo falla o no. X contiene las variables predictoras, es decir, las columnas usadas para intentar predecir la falla.

Figura 27

Eliminación de Columnas

```
failure_type_cols = ["TWF", "HDF", "PWF", "OSF", "RNF"]
existing_failure_type_cols = [col for col in failure_type_cols if col in X.columns]

if existing_failure_type_cols:
    print("\nSe eliminarán columnas de tipo de falla para evitar fuga de información:")
    print(existing_failure_type_cols)
    X = X.drop(columns=existing_failure_type_cols)
```

```
Se eliminarán columnas de tipo de falla para evitar fuga de información:
['TWF', 'HDF', 'PWF', 'OSF', 'RNF']
```

*Nota***.** Las columnas TWF, HDF, PWF, OSF y RNF indican tipos específicos de falla ya ocurridas. Si se dejan en el modelo, habría fuga de información.

Figura 28

Separación de Columnas Categóricas y Numéricas

```
# Separar tipos de columnas
categorical_cols = X.select_dtypes(include=["object"]).columns.tolist()
numeric_cols = X.select_dtypes(include=np.number).columns.tolist()

print("\nColumnas categóricas:", categorical_cols)
print("Columnas numéricas:", numeric_cols)
```

```
Columnas categóricas: ['Type']
Columnas numéricas: ['Air temperature [K]', 'Process temperature [K]', 'Rotational speed [rpm]', 'Torque [Nm]', 'Tool wear [min]'
```

Nota. Identificamos que columnas son categóricas y cuales son numéricas, La columna Type representa el tipo de máquina, y está en formato texto.

Figura 29

Codificación de Variable Categórica

```
# Codificar columnas categóricas simples manualmente si hace falta
# En AI4I normalmente solo 'Type' es categórica
X_encoded = X.copy()

for col in categorical_cols:
    le = LabelEncoder()
    X_encoded[col] = le.fit_transform(X_encoded[col].astype(str))
```

Nota. Se convierte la variable type a números**.**

En la fase de modelado se implementaron tres algoritmos de clasificación supervisada: regresión logística, Random Forest y XGBoost. La regresión logística fue utilizada como modelo base debido a su interpretabilidad. Random Forest fue seleccionado por su capacidad para capturar relaciones no lineales entre variables operativas. Por su parte, XGBoost fue incluido por su alto desempeño en problemas de clasificación con datos tabulares y su capacidad de optimización mediante ensambles secuenciales de árboles de decisión.

Tabla 1

Tabla Comparativa de Modelos

Modelo	Accuracy	Recall	Precision	F1-score	ROC-AUC
Regresión logística	0,8210	0,8382	0,1411	0,2415	0,9062
Random Forest	0,9780	0,7353	0,6579	0,6944	0,9704
XGBoost	0,9880	0,6912	0,9400	0,7966	0,9760
Random Forest ajustado	0,9705	0,8088	0,5446	0,6509	0,9733

Nota. XGBoost obtuvo el mayor accuracy, precision, F1-score y ROC-AUC, lo cual indica una alta capacidad general de clasificación. El recall adquiere mayor relevancia, ya que permite medir la proporción de fallas reales correctamente detectadas.

Aunque XGBoost obtuvo el mejor desempeño global en accuracy, precision, F1-score y ROC-AUC, el Random Forest ajustado fue seleccionado como el modelo más adecuado para el objetivo operativo del proyecto. Esta decisión se fundamentó en su recall de 80,88%, superior al de Random Forest y XGBoost, debido a que en mantenimiento predictivo los falsos negativos representan el error de mayor impacto al corresponder a fallas reales que el modelo no logra detectar.

Figura 30

Regresión Logística

```
============================================================
MODELO: Regresión Logística
============================================================
Accuracy : 0.8210
Recall   : 0.8382
Precision: 0.1411
F1-score : 0.2415
ROC-AUC  : 0.9062

Matriz de confusión:
[[1585  347]
 [  11   57]]

Reporte de clasificación:
              precision    recall  f1-score   support

           0       0.99      0.82      0.90      1932
           1       0.14      0.84      0.24        68

    accuracy                           0.82      2000
   macro avg       0.57      0.83      0.57      2000
weighted avg       0.96      0.82      0.88      2000
```

Figura 31

Grafica de Regresión Logística

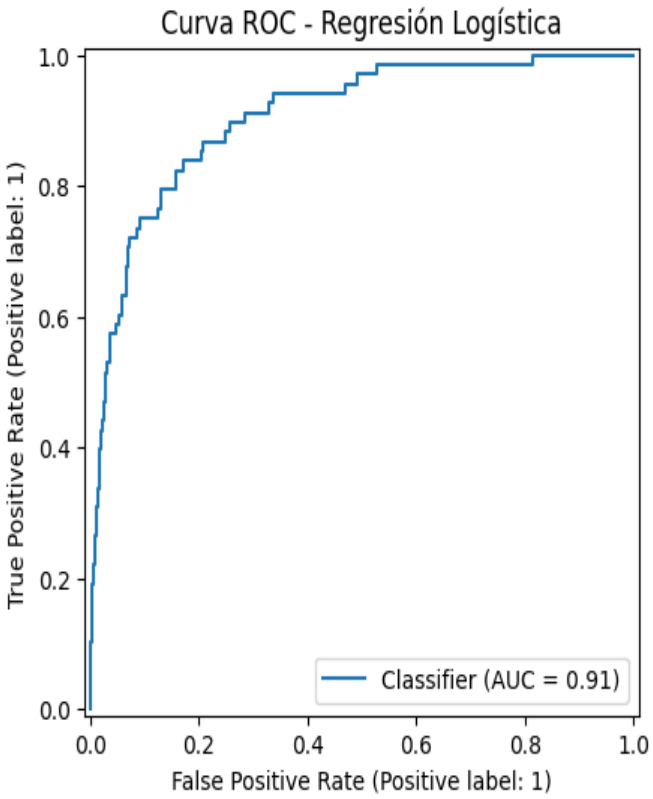


Figura 32

Random Forest

```
============================================================
MODELO: Random Forest
============================================================
Accuracy : 0.9780
Recall   : 0.7353
Precision: 0.6579
F1-score : 0.6944
ROC-AUC  : 0.9704

Matriz de confusión:
[[1906   26]
 [  18   50]]

Reporte de clasificación:
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      1932
           1       0.66      0.74      0.69        68

    accuracy                           0.98      2000
   macro avg       0.82      0.86      0.84      2000
weighted avg       0.98      0.98      0.98      2000
```

Nota. De 68 fallas reales, detectó 50 y generó solo 26 falsos positivos. Este modelo tiene menor recall que la regresión logística, pero es mucho más equilibrado, porque reduce las falsas alarmas.

Figura 33

Grafica de Random Forest

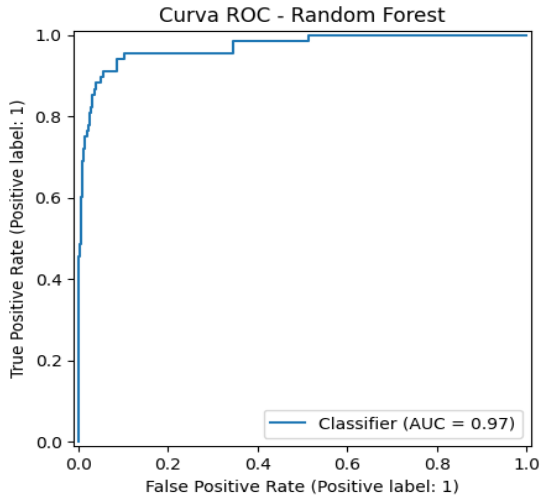


Figura 34

XGBoost

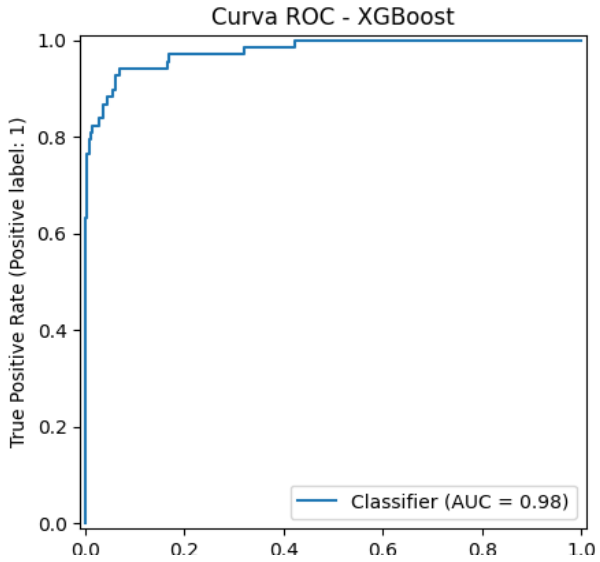
```
============================================================
MODELO: XGBoost
============================================================
Accuracy : 0.9880
Recall   : 0.6912
Precision: 0.9400
F1-score : 0.7966
ROC-AUC  : 0.9760

Matriz de confusión:
[[1929    3]
 [  21   47]]

Reporte de clasificación:
              precision    recall  f1-score   support

           0       0.99      1.00      0.99      1932
           1       0.94      0.69      0.80        68

    accuracy                           0.99      2000
   macro avg       0.96      0.84      0.90      2000
weighted avg       0.99      0.99      0.99      2000
```

Nota. Detectó 47 de 68 fallas y Dejó pasar 21 fallas. Generó solo 3 falsos positivos.

Tiene la mejor precisión: 94.00% y el mejor F1-score: 79,66%

Figura 35

Grafica de XGBoost



Los resultados muestran que XGBoost obtuvo el mayor accuracy, precision, F1-score y ROC-AUC, lo cual indica una alta capacidad general de clasificación. Sin embargo, desde el enfoque de mantenimiento predictivo, el recall adquiere mayor relevancia, ya que permite medir la proporción de fallas reales correctamente detectadas. En este sentido, la regresión logística presentó el mayor recall inicial, pero con una baja precisión, generando una alta cantidad de falsos positivos. Por su parte, el Random Forest ajustado logró un equilibrio más adecuado entre detección de fallas y reducción de falsas alarmas.

Figura 36

Comparación de Modelos

```
Resumen comparativo:
                Modelo  Accuracy    Recall  Precision  F1_score   ROC_AUC
0  Regresión Logística     0.821  0.838235   0.141089  0.241525  0.906246
1        Random Forest     0.978  0.735294   0.657895  0.694444  0.970383
2              XGBoost     0.988  0.691176   0.940000  0.796610  0.976046

Ranking de modelos:
                Modelo    Recall  F1_score  Accuracy   ROC_AUC
0  Regresión Logística  0.838235  0.241525     0.821  0.906246
1        Random Forest  0.735294  0.694444     0.978  0.970383
2              XGBoost  0.691176  0.796610     0.988  0.976046
```

Nota. Se ordena por el resultado de recall los modelos

Figura 37

Comparación de Modelos por Métrica

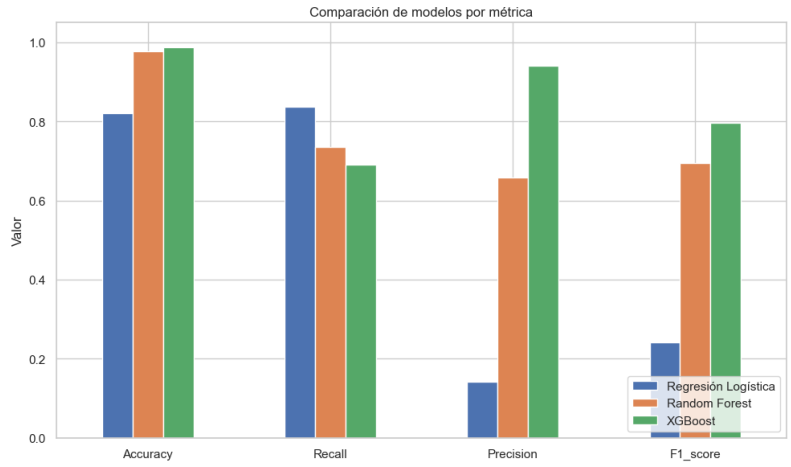


Figura 38

Ajuste de Hiperparámetros de Random Forest

```
Fitting 5 folds for each of 24 candidates, totalling 120 fits

Mejores parámetros RF:
{'classifier__max_depth': 8, 'classifier__min_samples_leaf': 2, 'classifier__min_samples_split': 2, 'classifier__n_estimators': 200}
Mejor recall CV: 0.6865993265993265
```

Nota. Se definen varias combinaciones para buscar el mejor Random Forest. Se evidencian los mejores parámetros encontrados y el Mejor recall en validación cruzada 0.6866

La matriz de confusión del modelo Random Forest ajustado muestra que el modelo clasificó correctamente 1886 registros sin falla y detectó 55 fallas reales. No obstante, se presentaron 13 falsos negativos, correspondientes a fallas reales no detectadas, y 46 falsos positivos, correspondientes a equipos clasificados como falla sin que realmente la presentaran. Desde el punto de vista industrial, los falsos negativos representan el error más crítico, ya que implican fallas no anticipadas que pueden generar paradas no programadas.

Figura 39

Evaluación del Mejor Modelo Ajustado- Random Forest

```
============================================================
MODELO: Random Forest Ajustado
============================================================
Accuracy : 0.9705
Recall   : 0.8088
Precision: 0.5446
F1-score : 0.6509
ROC-AUC  : 0.9733

Matriz de confusión:
[[1886   46]
 [  13   55]]

Reporte de clasificación:
              precision    recall  f1-score   support

           0       0.99      0.98      0.98      1932
           1       0.54      0.81      0.65        68

    accuracy                           0.97      2000
   macro avg       0.77      0.89      0.82      2000
weighted avg       0.98      0.97      0.97      2000
```

Nota. Se presentaron 13 falsos negativos de fallas reales no detectadas y 46 falsos positivos de equipos clasificados como falla sin que realmente la presentaran

Figura 40

Importancia de Variables

```
Importancia de variables:
                   Variable  Importancia
4              Torque [Nm]     0.340340
3   Rotational speed [rpm]     0.293523
5          Tool wear [min]     0.209961
1      Air temperature [K]     0.091849
2  Process temperature [K]     0.053237
0                     Type     0.011091
```

Nota. Las variables más influyentes para predecir fallas fueron Torque [Nm],Rotational speed [rpm], Tool wear [min]

Tabla 2

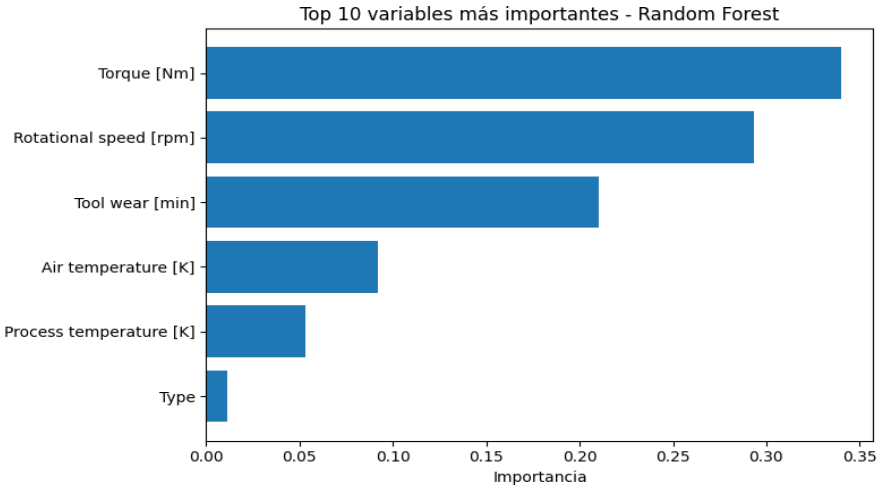
Tabla Análisis de Importancia de Variables

Variable	Importancia
Torque [Nm]	0,3403
Rotational speed [rpm]	0,2935
Tool wear [min]	0,2100
Air temperature [K]	0,0918
Process temperature [K]	0,0532
Type	0,0111

Nota. El torque puede reflejar condiciones de carga o esfuerzo mecánico, la velocidad de rotación puede asociarse con condiciones anormales de operación y el desgaste de herramienta representa el deterioro acumulado durante el proceso productivo.

Figura 41

Grafica de Top 10 de Variables



El análisis de importancia de variables permitió identificar que las variables con mayor influencia en la predicción de fallas fueron Torque [Nm], Rotational speed [rpm] y Tool wear [min]. Este resultado es coherente con el comportamiento físico de los equipos industriales, ya que el torque puede reflejar condiciones de carga o esfuerzo mecánico, la velocidad de rotación puede asociarse con condiciones anormales de operación y el desgaste de herramienta representa el deterioro acumulado durante el proceso productivo.

Recomendaciones

La primera recomendación es excluir de las variables predictoras las columnas que representan directamente el tipo de falla, tales como TWF, HDF, PWF, OSF y RNF, cuando la variable objetivo sea Machine failure. Estas columnas corresponden a etiquetas relacionadas con el resultado que se pretende predecir y su inclusión como entradas del modelo produciría fuga de información, generando métricas artificialmente elevadas y conclusiones poco válidas para un escenario real de predicción. De igual manera, se aconseja excluir identificadores como UID y Product ID, dado que no representan condiciones operativas del equipo.

Se recomienda profundizar en el análisis de variables como temperatura del aire, temperatura del proceso, velocidad de rotación, torque y desgaste de herramienta, debido a que estas variables representan condiciones físicas relevantes para el comportamiento de los equipos. Puede ser útil construir nuevas variables derivadas, como la diferencia entre temperatura del proceso y temperatura del aire, o una aproximación de la potencia mecánica a partir del torque y la velocidad de rotación.

Se recomienda analizar principalmente el recall, este permite identificar la capacidad del modelo para detectar eventos de falla y reducir falsos negativos. También se debe complementar el análisis con precision, F1-score, matriz de confusión y ROC-AUC, de manera que la selección del modelo considere tanto la detección de fallas como el control de falsas alarmas.

El dataset AI4I 2020 que se usó en este proyecto corresponde a un conjunto de datos sintético diseñado para representar condiciones de mantenimiento predictivo industrial. Se recomienda presentar los resultados como una validación metodológica y experimental del uso de modelos de machine learning para la predicción de fallas, evitando afirmar que el modelo puede ser implementado directamente en una planta real sin evaluaciones adicionales. Como

trabajo futuro, sería conveniente validar el modelo con datos reales provenientes de sensores, sistemas de control industrial o registros históricos de mantenimiento.

Conclusiones

En el desarrollo del trabajo se implementaron y evaluaron modelos de machine learning orientados a la detección temprana de fallas en equipos industriales, usando el dataset AI4I 2020. A partir del análisis exploratorio se identificó que el dataset presenta variables operativas relevantes, como lo son temperatura del aire, temperatura del proceso, velocidad de rotación, torque y desgaste de herramienta, las cuales permiten caracterizar el comportamiento de los equipos bajo diferentes condiciones de operación.

Se evidenció un fuerte desbalance en la variable objetivo, debido a que los registros asociados a fallas representan una proporción mínima del total de datos. Esta condición es propia de los entornos industriales reales, donde las fallas no ocurren con la misma frecuencia que las condiciones normales de operación, pero generan impactos significativos en la productividad, los costos de mantenimiento y la continuidad del proceso productivo.

La preparación de los datos fue un proceso importante para garantizar la calidad del modelado. La eliminación de variables identificadoras y de columnas asociadas directamente a tipos de falla permitió evitar fuga de información, fortaleciendo la validez del proceso predictivo. La codificación de variables categóricas, la normalización y la división estratificada de los datos permitieron construir un flujo de trabajo adecuado para el entrenamiento y evaluación de los modelos.

Los modelos implementados permitieron comparar diferentes enfoques de clasificación. La regresión logística presentó un recall alto, pero generó una cantidad considerable de falsos positivos. XGBoost obtuvo el mejor desempeño general en métricas como accuracy, precision, F1-score y ROC-AUC, aunque detectó una menor proporción de fallas reales. Por su parte, el

Random Forest ajustado logró un equilibrio adecuado entre la detección de fallas y la reducción de falsas alarmas, alcanzando un recall de 80,88% y un ROC-AUC de 97,33%.

Finalmente, se concluye que las técnicas de machine learning representan una herramienta útil para apoyar estrategias de mantenimiento predictivo en entornos industriales. El modelo desarrollado permite identificar patrones asociados a la ocurrencia de fallas y puede servir como base para sistemas de soporte a la toma de decisiones, orientados a reducir tiempos de inactividad, optimizar recursos y mejorar la eficiencia operativa.

Referencias Bibliográficas

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324

Carvalho, T. P., Soares, F. A., Vita, R., Francisco, R. P., Basto, J. P., & Alcalá, S. G. (2019). A systematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering*, 137, 106024. https://doi.org/10.1016/j.cie.2019.106024

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. En Proceedings of the 22nd ACM SIGKDD *International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). Association for Computing Machinery. https://doi.org/10.1145/2939672.2939785

Correa, J. (2015). *Escritura e investigación académica: Una guía para la elaboración del trabajo de grado*. https://bibliotecavirtual.unad.edu.co/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=2014981&lang=es&site=eds-live&scope=site

Géron, A. (2022). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow* (3.ª ed.). O'Reilly Media. https://www.oreilly.com/library/view/hands-on-machine-learning/9781098125967/

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. https://www.deeplearningbook.org/

Haya, P. (2022, 5 de enero). *La metodología CRISP-DM en ciencia de datos*. Instituto de Ingeniería del Conocimiento. https://www.iic.uam.es/innovacion/metodologia-crisp-dm-ciencia-de-datos/

AI4I 2020 Predictive Maintenance Dataset [Conjunto de datos]. (2020). *UCI Machine Learning Repository*. https://doi.org/10.24432/C5HS5C

Kumar, S., Gopi, T., Harikeerthana, N., Gupta, M. K., Gaur, V., Krolczyk, G. M., & Wu, C. (2023). Machine learning techniques in additive manufacturing: A state of the art review on design, processes and production control. *Journal of Intelligent Manufacturing, 34*, 21–55. https://doi.org/10.1007/s10845-022-02029-5

Lee, J., Bagheri, B., & Kao, H. A. (2015). A cyber-physical systems architecture for Industry 4.0-based manufacturing systems. *Manufacturing Letters, 3*, 18–23. https://doi.org/10.1016/j.mfglet.2014.12.001

Lei, Y., Yang, B., Jiang, X., Jia, F., Li, N., & Nandi, A. K. (2020). Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mechanical Systems and Signal Processing, 138*, 106587. https://doi.org/10.1016/j.ymssp.2019.106587

Mobley, R. K. (2002). *An introduction to predictive maintenance* (2nd ed.). Elsevier. https://www.sciencedirect.com/book/9780750675314/an-introduction-to-predictive-maintenance

Pang, J. L. (2023). Adaptive fault prediction and maintenance in production lines using deep learning. *International Journal of Simulation Modelling, 22*(4), 734–745. https://doi.org/10.2507/IJSIMM22-4-CO20

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., VanderPlas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830. https://jmlr.org/papers/v12/pedregosa11a.html

Susto, G. A., Schirru, A., Pampuri, S., McLoone, S., & Beghi, A. (2015). Machine learning for predictive maintenance: A multiple classifier approach. *IEEE Transactions on Industrial Informatics, 11*(3), 812–820. https://doi.org/10.1109/TII.2014.2349359

Zhang, W., Yang, D., & Wang, H. (2019). Data-driven methods for predictive maintenance of industrial equipment: A survey. *IEEE Systems Journal, 13*(3), 2213–2227. https://doi.org/10.1109/JSYST.2019.2905565