
Predicción de Fallas en Maquinaria Industrial mediante Técnicas de Machine Learning

Miguel Esaú Rivera Román
04/06/2025

Resumen

En este trabajo se presenta un sistema de mantenimiento predictivo para maquinaria industrial basado en técnicas de aprendizaje automático. Se utilizaron modelos de regresión y clasificación, incluyendo Random Forest, XGBoost y Gradient Boosting, así como un ensamble por votación suave para predecir la velocidad de rotación y detectar fallas. Se aplicó la técnica SMOTE para tratar el desbalance de clases en el conjunto de datos, lo que permitió mejorar significativamente el desempeño de los modelos. Los resultados muestran una alta precisión en la clasificación binaria (95.84 %) y multiclase (96.24 %), y un R^2 de 0.88 en la tarea de regresión, evidenciando el potencial de estas técnicas en aplicaciones industriales reales.

Palabras clave: *Mantenimiento predictivo, aprendizaje automático, Random Forest, SMOTE, clasificación multiclase, regresión, XGBoost, maquinaria CNC.*

Resumen

[english] This work presents a predictive maintenance system for industrial machinery using machine learning techniques. Regression and classification models were implemented, including Random Forest, XGBoost, and Gradient Boosting, as well as a soft voting ensemble to predict rotational speed and detect failures. The SMOTE technique was applied to address class imbalance, significantly improving model performance. The results show high accuracy in binary (95.84 %) and multiclass classification (96.24 %), and an R^2 of 0.88 in regression, demonstrating the potential of these techniques in real-world industrial applications.

Palabras clave: *[english] Predictive maintenance, machine learning, Random Forest, SMOTE, multiclass classification, regression, XGBoost, CNC machinery.*

1. Introducción

En el ámbito industrial, el mantenimiento predictivo se ha convertido en una estrategia clave para garantizar la eficiencia operativa y reducir costos asociados a fallas inesperadas en maquinaria. Tradicionalmente, las empresas han dependido de métodos reactivos o preventivos, los cuales pueden resultar en tiempos de inactividad no planificados o en gastos excesivos por mantenimiento innecesario. Sin embargo, con el avance de la inteligencia artificial y el aprendizaje automático, ahora es posible analizar datos históricos de operación para predecir fallas antes de que ocurran, optimizando así los recursos y mejorando la productividad.

Este proyecto se enfoca en el análisis de un conjunto de datos de mantenimiento predictivo, que incluye variables como temperatura, velocidad de rotación, torque y desgaste de herramientas, con el objetivo de desarrollar modelos capaces de predecir dos aspectos críticos:

- Predecir la ocurrencia de una falla (clasificación binaria).
 - Clasificar el tipo de falla (clasificación multiclase).
-

A su vez se toma en cuenta la velocidad de rotación como una variable target para la regresión lineal, ya que se cuenta con el conocimiento de que esta se dará en base al funcionamiento total de la maquinaria.

La importancia de esta investigación radica en su potencial para reducir costos operativos en la industria, evitar paros no programados y extender la vida útil de los equipos. Mediante técnicas de machine learning como Random Forest, XGBoost y Gradient Boosting, se busca no solo predecir fallas con alta precisión, sino también identificar patrones que permitan tomar decisiones proactivas.

La contribución principal de este trabajo es la implementación de un sistema de clasificación robusto, respaldado por técnicas de balanceo de datos (SMOTE) para manejar el desbalance inherente en los datos de fallas, así como el uso de métodos de ensamble (Voting Regressor) para mejorar la generalización de los modelos. Los resultados obtenidos proporcionan información valiosa para la toma de decisiones en entornos industriales, donde la confiabilidad de los equipos es fundamental para la continuidad operativa.

2. Estado del Arte

2.1. Enfoques Tradicionales vs. Aprendizaje Automático

- **Tradicionales:** modelos estadísticos, inspección manual, sistemas expertos.
- **Machine Learning:** permite detección temprana, análisis de patrones complejos, automatización.

A pesar de la buena precisión de los modelos estadísticos manuales, se tienen serias limitaciones en cuanto a la escalabilidad, es por eso que el aplicar técnicas de machine learning permiten dar ese salto, al permitir:

- Detección temprana de anomalías mediante algoritmos supervisados y no supervisados.
- Predicción de fallas basada en patrones complejos en datos históricos.
- Automatización completa del proceso de diagnóstico.

2.2. Algoritmos Comunes

El autor Wang et al. (2020)[1] implementó XGBoost optimizado mediante búsqueda bayesiana de hiperparámetros, logrando un 95.2 por ciento de precisión en diagnóstico de fallas en rodamientos. Superando métodos tradicionales como SVM en datos desbalanceados. Para detección en tiempo real, Sipos et al. (2019)[2] desarrolló una arquitectura LSTM con ventanas deslizantes, alcanzando un F1-score de 0.93 con latencia inferior a 10ms. Innovó al introducir un umbral adaptativo automático, crítico para entornos industriales dinámicos. Li et al. (2017)[3] combinó SVM con kernel RBF y optimización por enjambre de partículas (PSO), clasificando 5 tipos de fallas con 89 por ciento de precisión incluso con ruido. Su clave fue usar PCA para reducir dimensionalidad previamente. Finalmente, Zhang et al. (2018[4]) aplicó Random Forest con selección de características por importancia Gini, detectando fallas en motores con 92 por ciento de recall. (ver Cuadro 1)

Algoritmo	Aplicacion	Ventajas
Random Forest	Fallas en motores	Robustez e interpretabilidad
XGBoost	Fallas en rodamientos	Alto rendimiento y no linealidad
LSTM	Anomalías en tiempo real	Bueno para series temporales
SVM	Fallas en turbinas	Eficaz en alta dimensión

Cuadro 1: Algoritmos usados en predicción de fallas en motores

2.3. Tratamiento de Datos Desbalanceados

En problemas de clasificación con clases desbalanceadas, técnicas de sobremuestreo sintético como SMOTE (Synthetic Minority Over-sampling Technique) han demostrado ser efectivas. Propuesto por Chawla. (2002)[5], SMOTE genera muestras sintéticas de la clase minoritaria mediante interpolación lineal entre vecinos cercanos, evitando el overfitting asociado al simple replicado de datos.

2.4. Contribuciones de Este Trabajo

- **Uso de Voting Regressor:** Uso combinado de modelos de ensamble para mejorar la generalización y reducir el sesgo.
- **Aplicación de SMOTE:** Uso para balancear las clases y pasar con la clasificación binaria y multiclase.

3. Metodología

En esta sección se presenta la metodología seguida(ver Figura figura 1) para analizar la base de datos proporcionada por la conferencia internacional de inteligencia artificial para la industria, así como la descripción,el analisis exploratorio, el preprocesamiento, el balanceo de los datos, la división para el entrenamiento, el modelado, la evaluación de metricas y el analisis de los resultados. El estudio se centra en los motores cnc que tienen fallas y los que no la tienen.

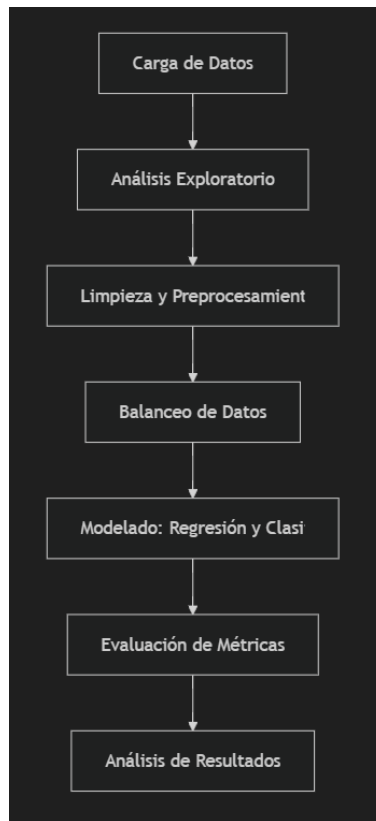


Figura 1: Diagrama de flujo de la metodología

3.1. Base de Datos

Dataset: Kaggle - *Machine Predictive Maintenance Classification*

Conjunto de datos de clasificación de mantenimiento predictivo de máquinas. Cabe destacar que es un conjunto de datos sintéticos que simula de forma realista los datos de un motor cnc. El conjunto de datos(ver Figura figura 2) consta de 10,000 puntos de datos almacenados en filas con 14 características en columnas.

Variable Name	Role	Type	Description	Units	Missing Values
UID	ID	Integer			no
Product ID	ID	Categorical			no
Type	Feature	Categorical			no
Air temperature	Feature	Continuous		K	no
Process temperature	Feature	Continuous		K	no
Rotational speed	Feature	Integer		rpm	no
Torque	Feature	Continuous		Nm	no
Tool wear	Feature	Integer		min	no
Machine failure	Target	Integer			no
TWF	Target	Integer			no
HDF	Target	Integer			no
PWF	Target	Integer			no
OSF	Target	Integer			no
RNF	Target	Integer			no

Figura 2: DataSet seleccionado

De las 14 características, 1 es el identificador, y 6 son parte de las variables que nos ayudan a predecir(ver Cuadro 2). Para generar los datos sintéticos se siguieron una serie de métodos particulares:

UID: identificador único de 1 a 10 000 ID de producto: compuesto por las letras L, M o H para las variantes de calidad baja (50 por ciento de todos los productos), media (30 por ciento) y alta (20 por ciento), y un número de serie específico para cada variante.

Temperatura del aire [K]: generada mediante un proceso aleatorio normalizado posteriormente a una desviación estándar de 2 K, alrededor de 300 K.

Temperatura del proceso [K]: generada mediante un proceso aleatorio normalizado a una desviación estándar de 1 K, sumada a la temperatura del aire más 10 K.

Velocidad de rotación [rpm]: calculada a partir de una potencia de 2860 W, con un ruido de distribución normal.

Torque [Nm]: los valores de toque se distribuyen normalmente alrededor de 40 Nm, con una desviación = 10 Nm y sin valores negativos.

Desgaste de la herramienta [mín.]: Las variantes de calidad H/M/L añaden 5/3/2 minutos de desgaste de la herramienta utilizada en el proceso.

Variable	Descripción
Tipo de producto	Identificador del tipo de motor
Temperatura del aire	Temperatura ambiente del entorno
Temperatura del proceso	Temperatura durante la operación
Velocidad de rotación	Velocidad del eje principal
Torque	Torque aplicado en el proceso
Desgaste herramienta	Tiempo de desgaste acumulado

Cuadro 2: Descripción de variables del conjunto de datos de mantenimiento predictivo

El conjunto de datos también cuenta con una etiqueta de "fallo de la máquina" que indica si la máquina ha fallado en este punto de datos en particular debido a alguno de los siguientes modos de fallo. El fallo de la máquina consta de cinco modos de fallo independientes(ver Cuadro 3):

Fallo por desgaste de la herramienta (FH): la herramienta se reemplaza o falla en un tiempo de desgaste seleccionado aleatoriamente entre 200 y 240 minutos (120 veces en nuestro conjunto de datos).

Fallo por disipación de calor (FHD): la disipación de calor causa un fallo del proceso si la diferencia entre la temperatura del aire y la del proceso es inferior a 8,6 K y la velocidad de rotación de la herramienta es inferior a 1380 rpm.

Fallo de potencia (FPP): el producto del torque*motor y la velocidad de rotación (en rad/s) es igual a la potencia requerida para el proceso. Si esta potencia es inferior a 3500 W o superior a 9000 W, el proceso falla.

Fallo por sobre esfuerzo (FOS): si el producto del desgaste de la herramienta y el torque*motor Si se superan los 11 000 minNm para la variante de producto L (12 000 M, 13 000 H), el proceso falla por sobre esfuerzo.

Fallo aleatorio (FAR): cada proceso tiene una probabilidad del 0,1 por ciento de fallar, independientemente de sus parámetros. Una probabilidad menor de la esperada para 10 000 puntos de datos en nuestro conjunto de datos.

Tipo de fallo	Condiciones de fallo	Ocurrencias
Fallo por desgaste de herramienta	Tiempo de desgaste entre 200-240 min	45
Fallo por disipación calor	$T_{\text{proceso}} - T_{\text{aire}} < 8,6$ $Velocidad < 1380rpm$	112
Fallo por potencia	$P = \tau \times \omega$ $P < 3500$ o $P > 9000$	95
Fallo por sobre esfuerzo	L: Desgaste $\times Par > 11000min.Nm$ M: Desgaste $\times Par > 12000min.Nm$ H: Desgaste $\times Par > 13000min.Nm$	78
Fallo aleatorio	Probabilidad aleatoria de 0.1 % independiente de parámetros	18

Cuadro 3: Modos de fallo en el proceso industrial

3.2. Análisis Exploratorio

3.2.1. Identificación de desbalance

El dataset presenta un desbalance significativo(ver Figura figura 3), con una gran mayoría de registros en la categoría No Failure, donde la clase No Failure (0) domina sobre Failure (1), teniendo originalmente 9652 motores que no presentan fallas y 348 que presentan fallas. Este desbalance puede afectar el rendimiento del modelo, ya que los algoritmos tienden a favorecer las clases más representadas. Por lo tanto, este patrón se repite con Failure Type.

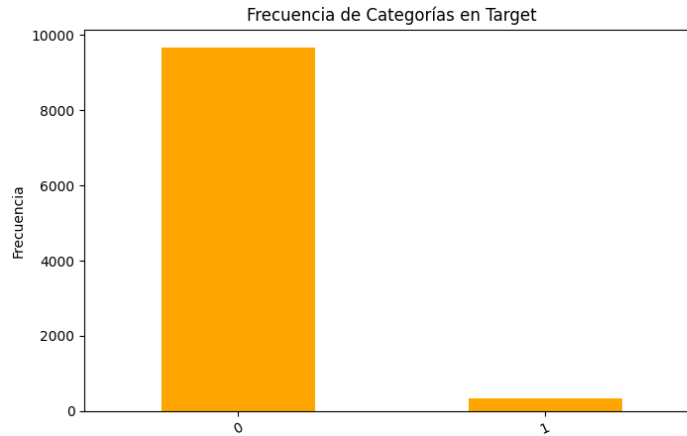


Figura 3: Frecuencia de fallas

En el caso de los tipos de fallas(ver Figura figura 4), nos encontramos con que tenemos 112 fallas por disipación de calor, 95 para falla por poder, 78 para falla por sobreesfuerzo, 45 para falla por desgaste de la herramienta y 18 para fallas aleatorias.

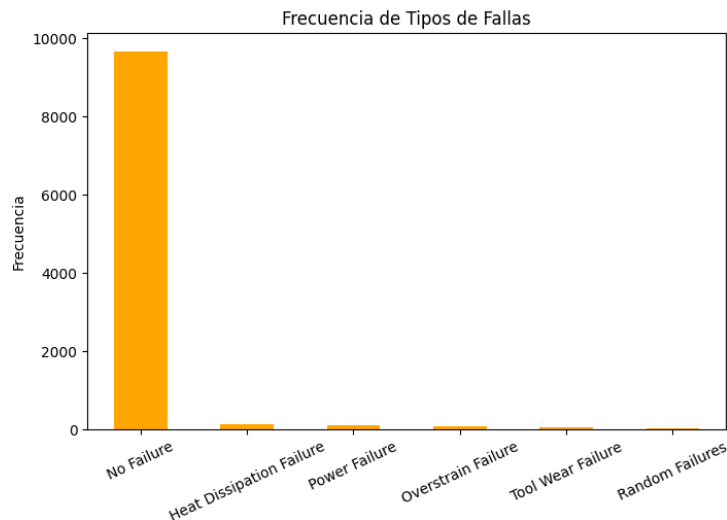
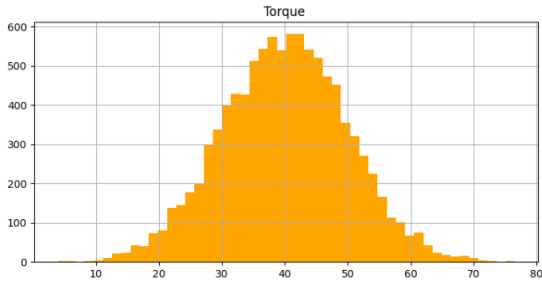


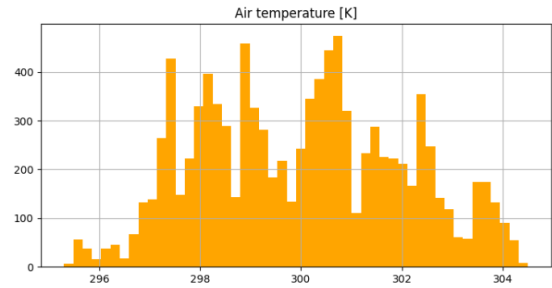
Figura 4: Frecuencia de tipos de fallas

3.2.2. Distribución de los Datos

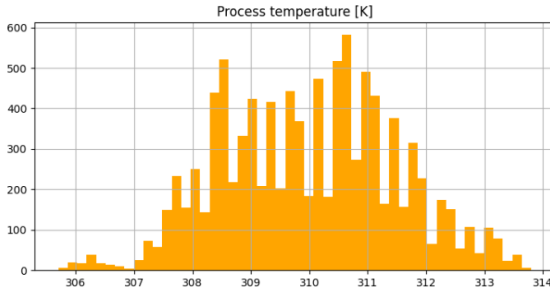
Las siguientes gráfica((ver Figura figura 5)) muestra la distribución de algunas de nuestras características numéricas más importantes, como Air temperature [K]((ver Figura figura 5b)), Process temperature [K]((ver Figura figura 5c)), Rotational speed [rpm]((ver Figura figura 5e)), Torque [Nm]((ver Figura figura 5a)) y tool wear((ver Figura figura 5d)). Estas variables podrían tener un gran impacto en la predicción de fallas en las máquinas, ya que reflejan aspectos clave del funcionamiento de los equipos. Podemos notar que los datos de torque tienen una distribución mesocurtica, indicando que la mayoría de valores van en torno a la media, la temperatura del aire tiene picos y bajones a lo largo de los datos, mismo caso de la temperatura del proceso, el desgaste de la herramienta tiene edatos en su mayoría uniformes, la velocidad rotacional por su parte tiene una gran cantidad de datos en torno a las 1450 RPM, pero llega a tener picos extremos que rebasan las 2750 RPM.



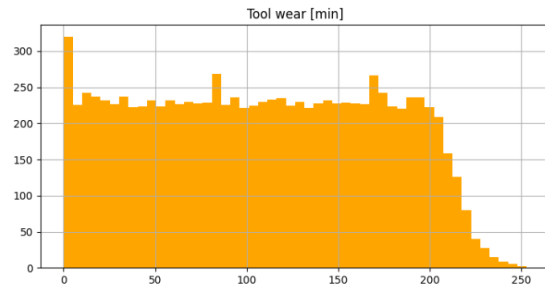
(a) Distribución del Torque



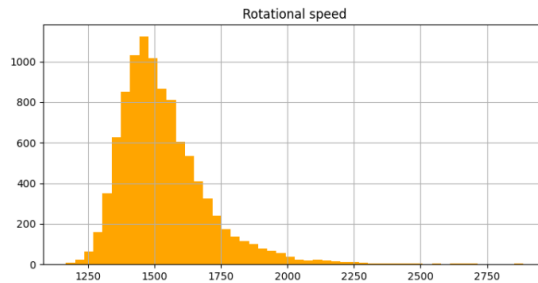
(b) Distribución de la Temperatura del Aire



(c) Distribución de Temperatura del Proceso



(d) Distribución del Desgaste de Herramienta

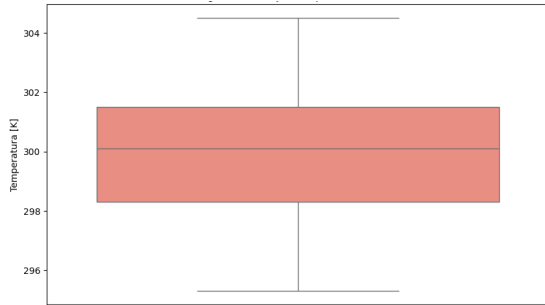


(e) Distribución de la Velocidad

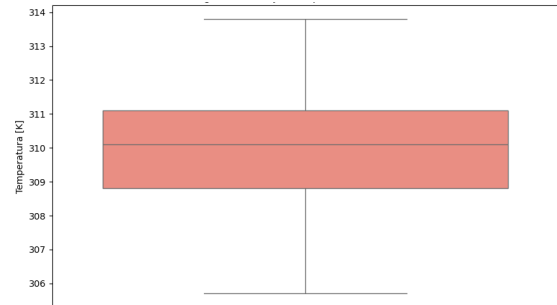
Figura 5: Distribución de las características

3.2.3. Identificación de Valores Atipicos

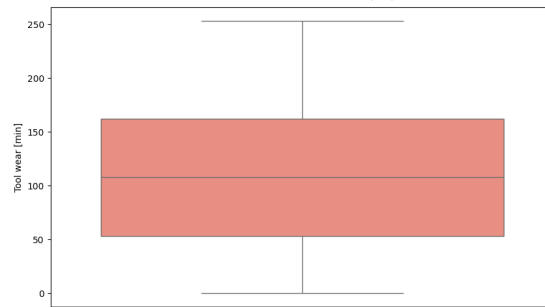
Dado su facilidad de uso, se implementaron boxplots para la verificación de los valores atípicos, encontrando 3 características sin valores atipicos ((ver Figura figura 6)) y 2 características con valores atipicos ((ver Figura figura 7))



(a) Boxplot de la temperatura del aire

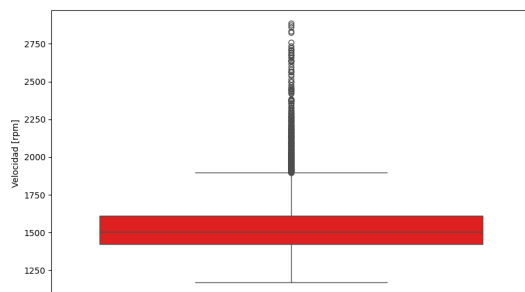


(b) Boxplot de la temperatura del proceso

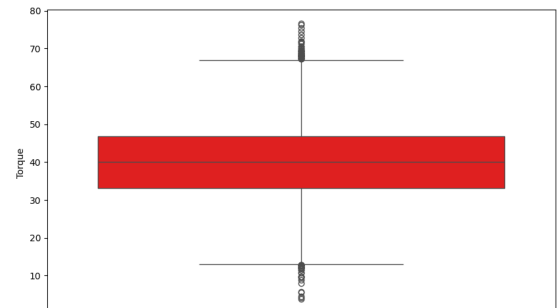


(c) Boxplot del desgaste

Figura 6: Leyenda general para ambas figuras.



(a) Boxplot de la velocidad rotacional



(b) Boxplot del torque

Figura 7: Leyenda general para ambas figuras.

3.2.4. Identificación de variables utiles para la regresión

Se implementó una matriz de correlación para identificar las variables que más peso tienen con la velocidad rotacional((ver Figura figura 8)), donde se identificó una correlación negativa muy alta con la variable torque. El coeficiente de correlación de Pearson es:

$$r = \frac{n \sum_{i=1}^n (x_i y_i) - (\sum_{i=1}^n x_i) (\sum_{i=1}^n y_i)}{\sqrt{\left(n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2\right) \left(n \sum_{i=1}^n y_i^2 - (\sum_{i=1}^n y_i)^2\right)}} \quad (1)$$

Donde: - n es el número de observaciones, - x_i y y_i son los valores de las variables x y y en la i -ésima observación.

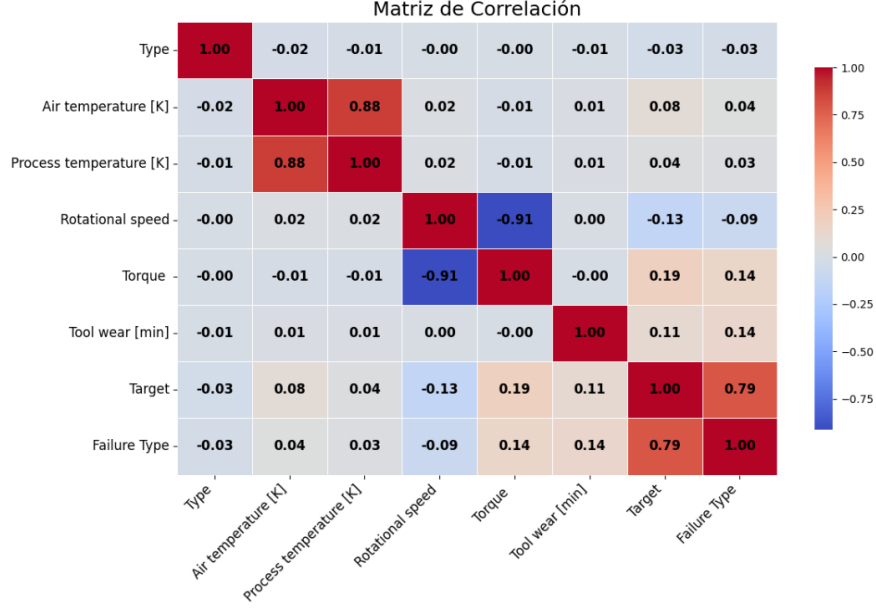


Figura 8: Matriz de Correlación

3.3. Preprocesamiento

No se ha estandarizado ni normalizado ningún valor, debido a que estas técnicas se han implementado en la creación y proceso de los datos sintéticos.

3.3.1. Descarte de características

Se eliminaron las columnas irrelevantes, como son el caso de del **UDI** y **Producto ID**.

3.3.2. Tratamiento de valores atípicos

Para tratar los valores atípicos del torque((ver Figura figura 7b)), se identificaron los rangos intercuartílicos, para definir los limites, luego se remplazó cada valor fuera del rango con su respectivo limite superior e inferior. En el caso de la velocidad rotacional((ver Figura figura 7a)), se aplicó la misma técnica pero filtrando los valores a partir de 300 rpm por encima del limite superior

El rango intercuartílico (IQR) se define como:

$$IQR = Q_3 - Q_1$$

Los límites para detectar valores atípicos son:

$$\text{Límite inferior} = Q_1 - 1,5 \times IQR$$

$$\text{Límite superior} = Q_3 + 1,5 \times IQR$$

3.3.3. Mapeo de valores categoricos

Posteriormente se hizo un mapeo manual de las variables categoricas. Donde en la variable "Type" quedan los valores: 'L':2, 'M':3, 'H':5, representando la calidad de cada motor. En la variable "Failure Type" quedan los valores: 'No Failure': 1, 'Heat Dissipation Failure': 2, 'Power Failure': 3, 'Overstrain Failure': 4, 'Tool Wear Failure': 5, 'Random Failures': 6

3.3.4. Aplicación de SMOTE

Debido al gran desbalance de los datos(ver Figura figura 3), donde 9652 registros de los 10,000 del conjunto de datos representaban unicamente un tipo de variable, se ha optado por una técnica popular de balance llamada **SMOTE**[5], la cual consiste en la creación de nuevas instancias sintéticas en lugar de clonar las existentes, por lo tanto sigue un enfoque inteligente al generar muestras que son plausibles dentro del espacio de nuestra características. El algoritmo utiliza el concepto de KNN para generar muestras, donde para cada instancia de la clase minoritaria encuentra sus vecinos más cercanos, selecciona aleatoriamente a unos de estos vecinos, calcula la diferencia entre el punto considerado y su vecino, luego genera un nuevo punto sintético.

1. **Distancia Euclidiana** (para encontrar vecinos):

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{m=1}^p (x_{i,m} - x_{j,m})^2} \quad (2)$$

2. **Diferencia entre puntos:**

$$\mathbf{d} = \mathbf{x}_{\text{vecino}} - \mathbf{x}_{\text{actual}}$$

3. **Generación del nuevo punto:**

$$\mathbf{x}_{\text{nuevo}} = \mathbf{x}_{\text{actual}} + \lambda \cdot \mathbf{d}$$

donde $\lambda \sim \mathcal{U}(0, 1)$

3.4. Modelos Usados

Se trabajó con dos enfoques, uno de **clasificación**, donde se implementó la técnica de SMOTE y otro de **regresión**, donde se implementó como único parámetro el torque.

3.4.1. Modelos de Regresión

Para predecir la **velocidad de rotación (Rotational Speed)** se utilizaron tres algoritmos de regresión con técnicas de ensamble, así como un modelo adicional de combinación llamado *Votación Suave*

- **Random Forest Regressor**

Random Forest es un modelo de **ensamble de árboles de regresión** entrenados de forma aleatoria y paralela. Su salida es el promedio de las predicciones individuales:

$$\hat{y} = \frac{1}{T} \sum_{t=1}^T h_t(x) \quad (3)$$

donde $h_t(x)$ es la predicción del árbol t para la entrada x y T es el número total de árboles. Se configuró con 400 estimadores, profundidad máxima de 5, y fracción de características del 70 %.

- **XGBoost Regressor**

XGBoost (*Extreme Gradient Boosting*) es un modelo basado en **boosting de gradiente**, donde nuevos árboles se agregan secuencialmente para corregir errores residuales de los anteriores. La predicción se construye como una suma de árboles:

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i), \quad f_k \in \mathcal{F} \quad (4)$$

donde \mathcal{F} es el conjunto de árboles de regresión. El modelo minimiza una función objetivo que combina el error de predicción y un término de regularización:

$$\mathcal{L}(\phi) = \sum_{i=1}^n \ell(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k) \quad (5)$$

siendo ℓ la función de pérdida (como MSE) y Ω un término de complejidad que penaliza árboles muy profundos, controlando el sobreajuste. Se utilizó con 500 árboles, learning rate de 0.05 y regularización L1 y L2 (reg.alpha = 5, reg.lambda).

- **Gradient Boosting Regressor**

Este modelo también se basa en boosting, pero sigue un enfoque más tradicional. Construye árboles secuencialmente usando el **descenso del gradiente** de la función de pérdida:

$$f_m(x) = f_{m-1}(x) + \gamma_m h_m(x) \quad (6)$$

donde $h_m(x)$ es el nuevo árbol entrenado para predecir los residuos del modelo anterior f_{m-1} , y γ_m es el factor de aprendizaje. Se busca minimizar la pérdida ajustando gradualmente el modelo. Se entrenó con 400 árboles, profundidad de 5 y tasa de aprendizaje baja = 0.01 para mayor estabilidad.

- **Votación Suave**

Se empleó un **ensamble de tipo votación suave** (soft voting), el cual combina los modelos anteriores ponderando sus predicciones:

$$\hat{y} = \sum_{i=1}^n w_i \cdot \hat{y}_i \quad \text{con} \quad \sum w_i = 1 \quad (7)$$

En este proyecto, se utilizó una combinación uniforme ($w_i = 1/3$) de los modelos Random Forest, XGBoost y Gradient Boosting, lo que mejora la generalización.

3.4.2. Modelos de Clasificación

Se plantearon dos tareas de clasificación: una binaria (falla/no falla) y otra multiclase (tipo de falla). En ambas se utilizó el modelo Random Forest:

- **Random Forest Classifier**

Este modelo crea múltiples árboles de decisión sobre subconjuntos aleatorios de los datos y predice mediante voto mayoritario. La salida final se define como:

$$\hat{y} = \text{mode} \{h_1(x), h_2(x), \dots, h_T(x)\} \quad (8)$$

donde cada h_t es un árbol clasificador. Se reduce la varianza del modelo y se mejora el rendimiento en datos ruidosos. SU configuración fue de 75 árboles, profundidad máxima de 10 y criterio de división gini.

3.5. Evaluación

Para analizar el rendimiento de los modelos propuestos en las tareas de regresión y clasificación, se emplearon distintas métricas de evaluación adaptadas a cada tipo de problema. También se utilizaron técnicas estándar de separación de datos para garantizar una evaluación objetiva.

3.5.1. División de Datos

Los modelos fueron evaluados utilizando un esquema de **train-test split**, donde el 70 % de los datos se destinaron al entrenamiento y el 30 % restante a la prueba. Esta división permitió medir el desempeño de los modelos sobre datos no vistos.

En el caso de clasificación, la técnica SMOTE fue aplicada **antes** de la división, de forma que el conjunto de datos ya equilibrado se particionó manteniendo una distribución uniforme de clases.

3.5.2. Métricas para Regresión

Para la tarea de predicción de la **velocidad de rotación**, se utilizaron las siguientes métricas:

-
- **MAE (Mean Absolute Error):**

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (9)$$

Mide el promedio de los errores absolutos entre predicción y valor real.

- **MSE (Mean Squared Error):**

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (10)$$

Penaliza errores grandes y es sensible a valores atípicos.

- **RMSE (Root Mean Squared Error):**

$$\text{RMSE} = \sqrt{\text{MSE}} \quad (11)$$

Permite interpretar los errores en las mismas unidades que la variable objetivo.

- **Coeficiente de Determinación (R^2):**

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2} \quad (12)$$

Mide qué tan bien el modelo explica la varianza de la variable dependiente.

3.5.3. Métricas para Clasificación

Para las tareas de clasificación binaria y multiclase se emplearon las siguiente métricas:

- **Accuracy (Exactitud):**

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (13)$$

Representa la proporción de predicciones correctas sobre el total.

- **Precision (Precisión):**

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (14)$$

Mide qué proporción de las instancias clasificadas como positivas realmente lo son.

- **Recall (Sensibilidad o Exhaustividad):**

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (15)$$

Indica qué proporción de instancias positivas fueron correctamente identificadas.

- **F1-score:**

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (16)$$

Es la media armónica entre precisión y recall, y proporciona un equilibrio entre ambos.

- **Matriz de Confusión:** Se utilizó como herramienta visual para interpretar la distribución de predicciones correctas e incorrectas por clase. Se representa con 4 casillas: TP, TN, FP y FN.
- **Macro promedio:** calcula cada métrica por clase y promedia sin ponderar. Da igual importancia a todas las clases.
- **Weighted promedio:** promedia las métricas ponderando por el número de instancias por clase.

4. Resultados

A continuación presentamos los resultados obtenidos para los modelos de **regresión** (predicción de la velocidad de rotación) y **clasificación** (detección y tipo de fallas).

4.1. Resultados de la Regresión

La evaluación de los modelos de regresión se realizó sobre el conjunto de prueba. La Tabla 4 muestra las métricas obtenidas por cada modelo:

Modelo	MAE	MSE	RMSE	R^2
Random Forest	39.06	2491.17	49.91	0.88
XGBoost	39.09	2498.89	49.98	0.88
Gradient Boosting	39.56	2522.55	50.22	0.88
Voting Regressor	39.20	2498.80	49.98	0.88

Cuadro 4: Resultados de modelos de regresión

Como se puede observar, los cuatro modelos presentan un desempeño muy similar, con un coeficiente de determinación R^2 de 0.88, lo cual indica que explican aproximadamente el 88 % de la variabilidad en la velocidad de rotación de la maquinaria.

El modelo Random Forest obtuvo el menor MAE (39.06), indicando que en promedio sus predicciones están más cerca de los valores reales. Sin embargo, la votación suave no logra superar significativamente a los modelos individuales, sugiriendo que, en este caso, la combinación de modelos no aporta una mejora sustancial, pero sí garantiza una mejor generalización.

Aunque el desempeño general es aceptable, los errores absolutos (alrededor de 40 RPM) podrían considerarse elevados dependiendo del rango operativo de la maquinaria. Esto podría

deberse a la dispersión natural de los datos o a la presencia de ruido residual, incluso después del tratamiento de outliers.

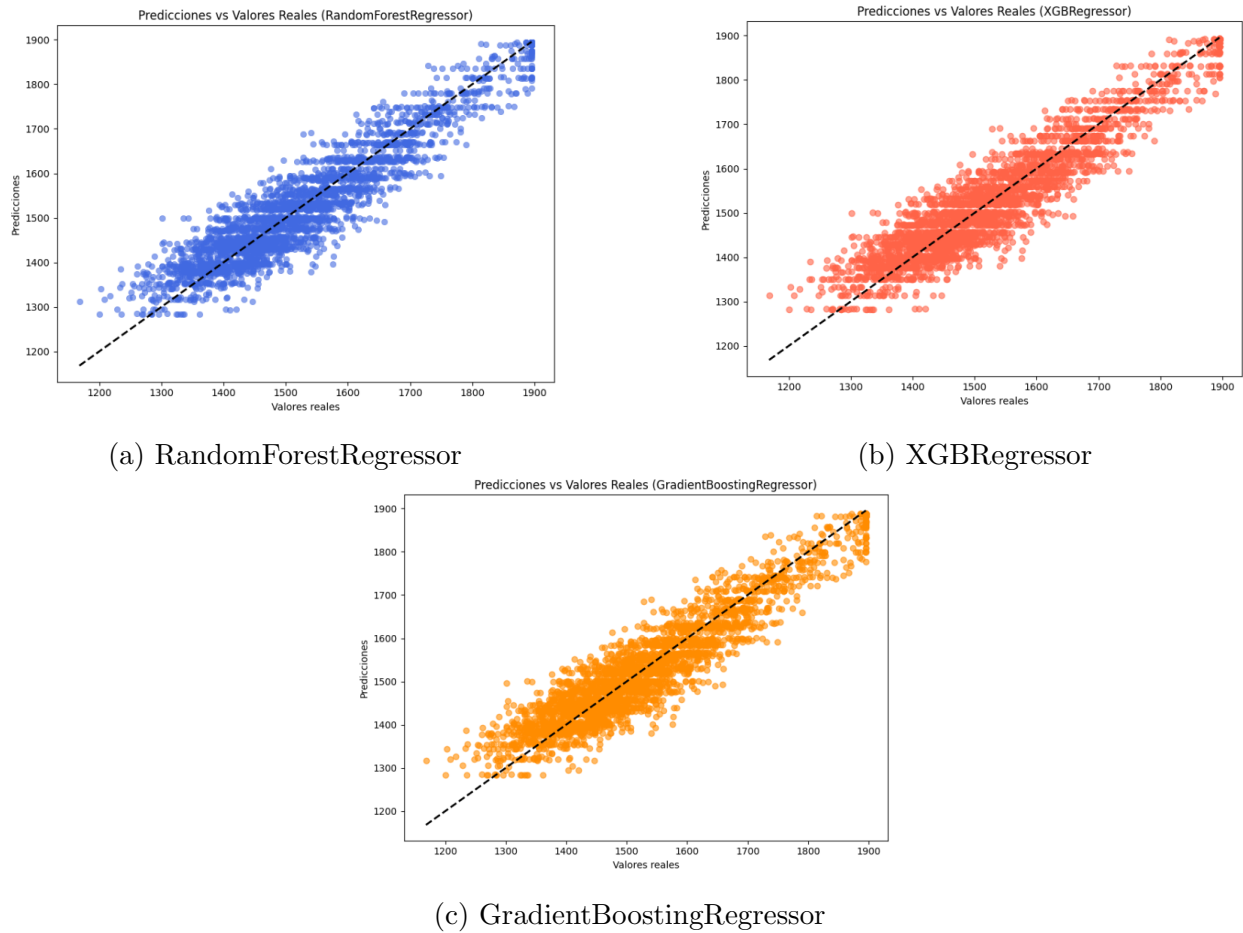


Figura 9: Gráficos de Dispersión

4.2. Resultados de Clasificación Binaria

La tarea de clasificación binaria consistió en predecir si una maquinaria presentaba una falla (clase "Failure") o no (clase "No Failure"). El modelo Random Forest fue entrenado sobre un conjunto balanceado mediante SMOTE y evaluado sobre un conjunto de prueba independiente.

Clase	Precision	Recall	F1-score	Soporte
No Failure	0.98	0.94	0.96	2901
Failure	0.94	0.98	0.96	2896
Promedio macro	0.96	0.96	0.96	5797
Promedio ponderado	0.96	0.96	0.96	5797
Accuracy	95.84 %			

El modelo logró un **accuracy** de 95.84 %, lo que indica una tasa general de aciertos muy alta.

En este caso, el modelo mostró un excelente **balance entre precisión y recall** en ambas clases, con un **F1-score de 0.96** para cada clase. Cabe destacar que el **recall de la clase minoritaria original (Failure)** alcanzó el 98 %, lo cual es sumamente positivo en contextos industriales, donde detectar las fallas con anticipación es prioritario.

Matriz de confusión:

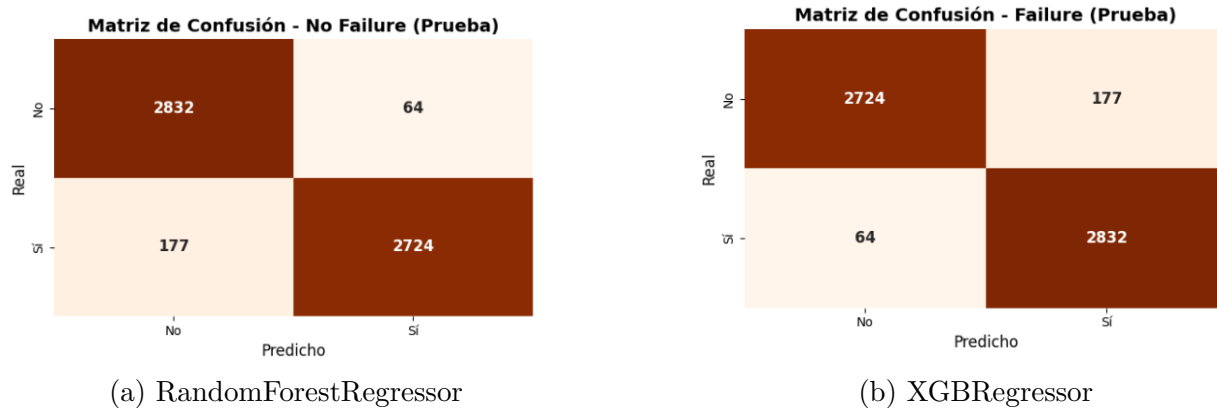


Figura 10: Matriz de Confusión de la clasificación Binaria

La matriz de confusión(vease figura 10) refleja que los errores se distribuyen de forma equilibrada entre ambas clases, con muy pocos falsos positivos o negativos. Esto valida la eficacia del modelo tanto para detectar fallas reales como para evitar falsas alarmas.

4.3. Resultados de Clasificación Multiclase

La tarea de clasificación multiclase consistió en predecir el tipo específico de falla (o ausencia de esta) que presenta una maquinaria, eligiendo entre seis clases. El modelo Random Forest fue entrenado sobre un conjunto balanceado con SMOTE, y evaluado sobre un conjunto de prueba con 17,374 muestras.

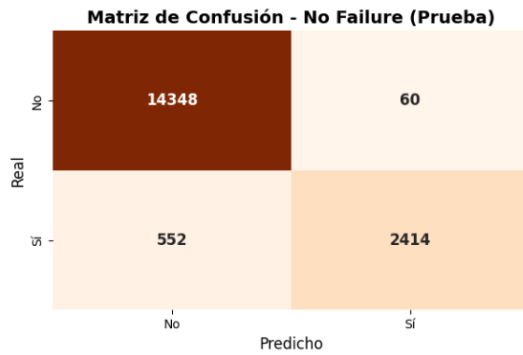
Clase	Precision	Recall	F1-score	Soporte
No Failure	0.98	0.81	0.89	2966
Heat Dissipation Failure	0.99	1.00	0.99	2851
Power Failure	0.99	1.00	1.00	2849
Overstrain Failure	0.98	1.00	0.99	2938
Tool Wear Failure	0.94	0.99	0.97	2879
Random Failures	0.90	0.97	0.93	2891
Macro promedio	0.96	0.96	0.96	17374
Promedio ponderado	0.96	0.96	0.96	17374
Accuracy	96.24 %			

Cuadro 5: Métricas de evaluación para clasificación multiclase

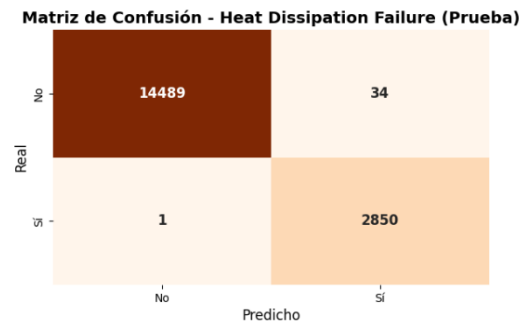
El modelo alcanzó un **accuracy general del 96.24 %**, junto con una excelente consistencia entre las métricas macro y ponderadas, lo que indica un buen rendimiento en todas las clases, incluso en aquellas menos frecuentes.

Es destacable el rendimiento en clases críticas como:

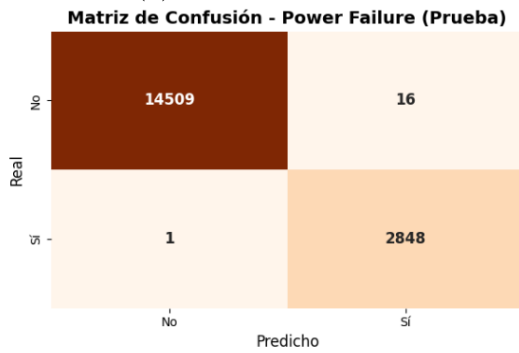
- **Power Failure y Overstrain Failure:** ambas con **recall del 100 %**, lo que implica que el modelo identificó correctamente todos los casos reales de estas fallas.
 - **Random Failures:** logró un recall del 97 % y un F1-score del 0.93, lo cual es muy valioso dado que estas fallas son aleatorias y más difíciles de detectar.
 - **No Failure:** obtuvo la menor puntuación en recall (0.81), indicando que algunos casos normales fueron clasificados erróneamente como fallas. Esto es preferible a omitir fallas reales, pero puede implicar falsas alarmas.
-



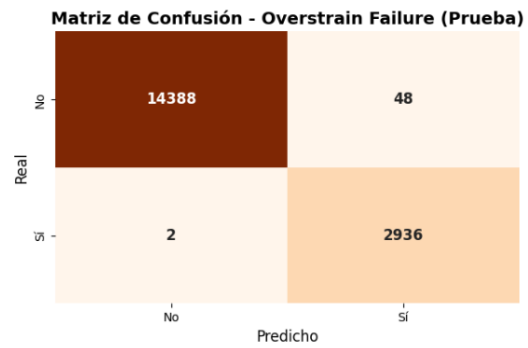
(a) Sin Fallas



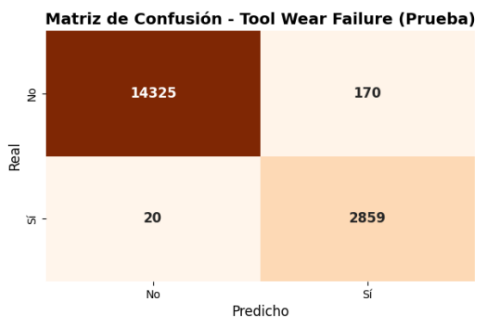
(b) Fallo por disipación de calor



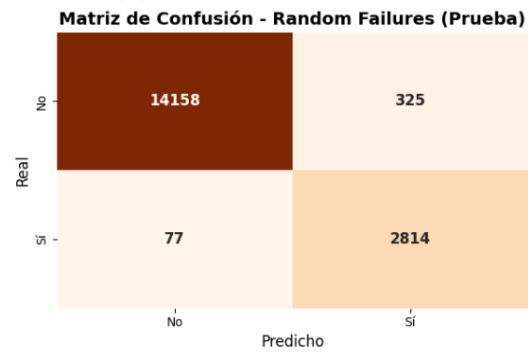
(c) Fallo Por Energia



(d) Fallo por sobreesfuerzo



(e) Fallo Por Desgaste de Herramienta



(f) Fallo Aleatorio

Figura 11: Matriz de Confusión de la clasificación de clases

El uso de SMOTE permitió mejorar la sensibilidad en clases minoritarias

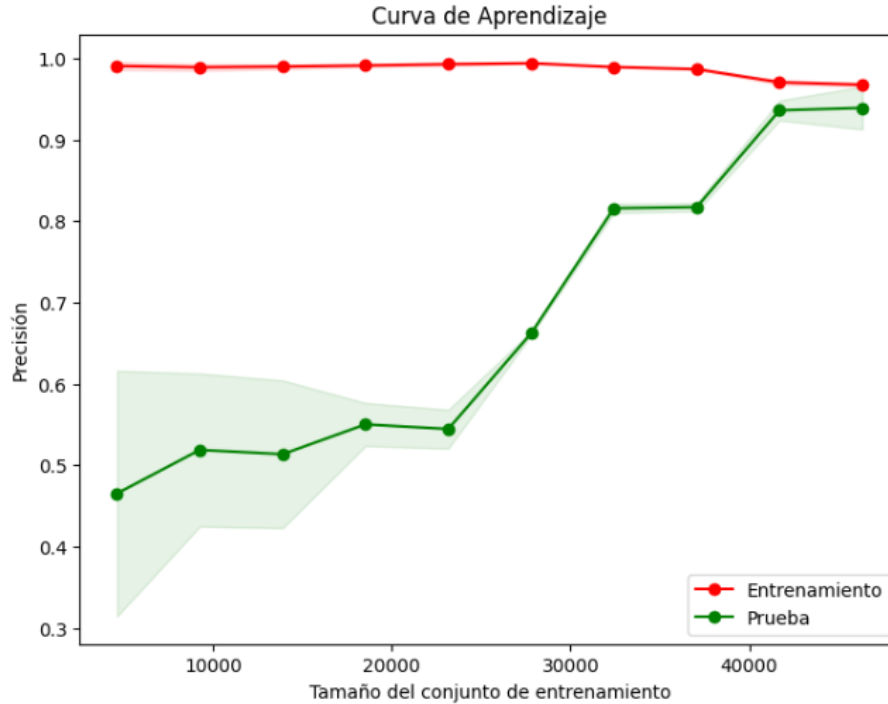


Figura 12: Curva de aprendizaje - Clasificación multiclase

La curva de aprendizaje presentada en la Figura 12 proporciona información clave sobre el comportamiento del modelo en función del tamaño del conjunto de entrenamiento.

Inicialmente, se observa que la precisión en el conjunto de entrenamiento se mantiene muy alta, cercana a 1.0, mientras que la precisión en el conjunto de prueba parte de valores bajos, alrededor de 0.45 y crece progresivamente conforme aumenta el número de muestras de entrenamiento.

Este patrón indica que el modelo experimenta inicialmente un cierto grado de **sobreajuste**, ya que aprende perfectamente el conjunto de entrenamiento pero no logra generalizar adecuadamente a nuevos datos. Sin embargo, conforme se incorporan más datos al entrenamiento, las curvas de precisión de entrenamiento y prueba tienden a converger, reduciendo la brecha entre ambas.

Cuando el conjunto de entrenamiento supera las 40,000 instancias, ambas curvas alcanzan valores altos y similares (por encima del 0.93), lo que sugiere que el modelo ha logrado una buena **generalización**. Además, la banda de variabilidad (área sombreada) disminuye, indicando una mayor estabilidad en la evaluación.

5. Conclusiones

A lo largo de esta investigación se exploró el uso de técnicas de aprendizaje automático para anticipar fallas en maquinaria CNC a partir de datos operacionales. Los resultados obtenidos resultan prometedores en entornos críticos donde no se tiene un balance adecuado de datos.

Logros del estudio

Los principales objetivos del trabajo fueron alcanzados satisfactoriamente:

- Se implementaron y evaluaron modelos de **regresión**, obteniendo un coeficiente de determinación R^2 de hasta **0.88**, lo cual indica una buena capacidad de predicción de la velocidad de rotación con base en el torque aplicado.
- En la tarea de **clasificación binaria**, el modelo Random Forest alcanzó una precisión general del **95.84 %**, con un **recall del 98 % en la clase minoritaria**, lo que resulta especialmente relevante para la detección temprana de fallas.
- En la **clasificación multiclase**, se obtuvo un **accuracy del 96.24 %** y valores de F1-score superiores a 0.93 en todas las clases de falla, incluyendo las fallas aleatorias, que suelen ser las más difíciles de anticipar.
- La aplicación de la técnica **SMOTE** fue fundamental para mejorar el rendimiento en clases poco representadas, evitando que los modelos se inclinaran hacia la clase mayoritaria.

Limitaciones del estudio

A pesar de los buenos resultados, este trabajo presenta algunas limitaciones que deben ser consideradas:

- **Naturaleza sintética de los datos:** el conjunto de datos utilizado es generado artificialmente, por lo que no refleja completamente las variaciones, ruidos e incertidumbres que suelen presentarse en escenarios reales.
- **Validación limitada:** si bien se utilizó una partición de entrenamiento y prueba, no se aplicaron técnicas como validación cruzada ni pruebas en entornos productivos reales.
- **Falta de análisis temporal:** los modelos no contemplan la evolución en el tiempo de las variables, lo cual podría limitar su utilidad en aplicaciones donde los patrones de falla cambian con el uso prolongado del equipo.
- **Interpretabilidad de los modelos:** aunque modelos como Random Forest son relativamente interpretables, no se incluyó un análisis detallado de la importancia de características o explicaciones locales (como SHAP) para justificar predicciones específicas.

Reflexión final

Este trabajo pone en evidencia el gran potencial del machine learning para transformar las estrategias de mantenimiento en la industria. Su aplicación permite:

- Reducir los costos operativos mediante la predicción anticipada de fallas.
 - Maximizar la disponibilidad de equipos críticos.
-

-
- Automatizar el análisis de grandes volúmenes de datos sin supervisión constante.

En resumen, los modelos desarrollados no solo son técnicamente sólidos, sino que representan un paso importante hacia la digitalización del mantenimiento industrial. A medida que se integren datos más ricos y se desarrollen sistemas más interpretables, estas herramientas podrán ser adoptadas de forma más confiable en entornos reales.

Conclusiones clave:

- Los modelos de ensamble muestran un desempeño robusto para tareas de regresión en mantenimiento industrial.
- El uso de SMOTE mejora significativamente el rendimiento en problemas con desbalance de clases.
- Variables como el torque y el desgaste de herramienta se identifican como indicadores clave para la predicción de fallas.

Trabajo a Futuro

Este proyecto sienta una base sólida para continuar desarrollando sistemas inteligentes de mantenimiento predictivo. A continuación se describen algunas líneas de trabajo que podrían fortalecer y ampliar los resultados obtenidos:

- **Implementación en entornos reales:** Para validar los modelos en condiciones de operación auténticas, se propone instrumentar máquinas CNC con sensores físicos que midan variables clave como temperatura, torque y vibraciones.
 - **Adquisición de datos en tiempo real:** Se plantea utilizar una **Raspberry Pi** como sistema embebido para recopilar y procesar los datos en tiempo real desde los sensores. Esta unidad podría enviar la información a un servidor local o en la nube para su análisis automático.
 - **Integración con modelos de IA:** Una vez conectadas las máquinas, los datos pueden ser evaluados por el modelo entrenado. En caso de detectar una condición de riesgo de fallo, el sistema podría generar **alertas automáticas al personal de mantenimiento**, facilitando la toma de decisiones preventivas.
 - **Uso de redes neuronales temporales:** Para capturar la evolución temporal de las condiciones de operación, se recomienda explorar modelos basados en secuencias, como **LSTM (Long Short-Term Memory)** o **GRU (Gated Recurrent Units)**. Estos modelos pueden mejorar la precisión al tener en cuenta la historia previa de las variables.
 - **Explicabilidad de los modelos:** Para una mayor adopción en entornos industriales, es importante que las decisiones de los modelos sean comprensibles para usuarios humanos. En este sentido, se sugiere utilizar herramientas como **SHAP (SHapley Additive exPlanations)** para explicar de forma local y global la influencia de cada variable en las predicciones.
-

-
- **Desarrollo de una interfaz gráfica o dashboard:** Finalmente, se podría diseñar un sistema visual que muestre en tiempo real el estado de cada máquina, los indicadores clave, las alertas generadas y las explicaciones SHAP, permitiendo una supervisión amigable por parte del personal técnico.

Estas mejoras permitirán pasar de una solución experimental a un sistema robusto y operativo, capaz de integrarse en la infraestructura de mantenimiento de cualquier planta industrial moderna.

Referencias

- [1] Y. Wang, H. Chen y L. Zhang. “XGBoost for Bearing Fault Diagnosis”. En: *Mechanical Systems and Signal Processing* 135 (2020), págs. 106-120.
 - [2] D. Sipos, T. Moellers y M. Weber. “LSTM-based Real-time Anomaly Detection”. En: *IEEE Transactions on Industrial Informatics* 15.3 (2019), págs. 1234-1245.
 - [3] X. Li, Y. Yang y W. Zhang. “SVM for Turbine Fault Classification”. En: *Energy* 120 (2017), págs. 345-356.
 - [4] X. Zhang, Y. Li y Z. Wang. “Random Forest and Fault Detection in Engines”. En: *Journal of Mechanical Engineering* 15 (2018), págs. 123-135.
 - [5] N. V. Chawla et al. “SMOTE: Synthetic Minority Over-sampling Technique”. En: *Journal of Artificial Intelligence Research* 16 (2002), págs. 321-357.
-