

Predicciones de fallas en sistemas de producción

Elementos de Inteligencia Artificial
Informe

Katherine Rodríguez Rodríguez

Ciencia de Datos
Universidad de la Habana, MATCOM
2025

1. Introducción

En los entornos más avanzados de la industria manufacturera se producen fallas. En el sector industrial, los fallos pueden tener consecuencias devastadoras, desde pérdidas financieras hasta riesgos para la seguridad. La detección de fallos en sistemas industriales es vital para minimizar el tiempo de inactividad y mejorar la seguridad operativa.

El objetivo de este proyecto consta de un análisis predictivo utilizando algoritmos avanzados para predecir fallos antes de que se conviertan en desastres costosos.

2. Metodología

2.1. Análisis y preprocessamiento de la base de datos

La base de datos fue extraída del sitio <https://archive.ics.uci.edu/dataset/601/ai4i+2020+predictive+maintenance+dataset>. El conjunto de datos consta de 10 000 filas con 14 características en columnas.

UID: identificador único que va del 1 al 10000.

ID de producto: compuesto por una letra L, M o H para bajo, medio y alto.

Temperatura del aire [K]: generada mediante un proceso de caminata aleatoria.

Temperatura del proceso [K]: generada mediante un proceso de caminata aleatoria normalizada, sumada a la temperatura del aire.

Velocidad de rotación [RPM]: calculada a partir de una potencia de 2860 W.

Par [Nm]: los valores de par normalmente se distribuyen alrededor de 40 Nm y sin valores negativos.

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

Fallo de máquina: Variable que indica si la máquina ha fallado.

También se presentan cinco columnas que representan tipos específicos de fallos, en cuanto uno de estos se detecta la variable fallo de máquina se activa en 1.

TWF (Fallo por desgaste de la herramienta): La herramienta falla o se reemplaza cuando el desgaste alcanza entre 200-240 minutos.

HDF (Fallo de disipación de calor): Fallo por disipación de calor.

OSF (Fallo por sobreesfuerzo): Fallo por sobreesfuerzo, relacionado con la combinación de torque y desgaste de herramienta.

PWF (Fallo de potencia): el producto del par y la velocidad de rotación (en rad/s) es igual a la potencia requerida para el proceso.

RNF (fallos aleatorios): Otros tipos de fallos específicos del sistema.

Se llevó a cabo un análisis descriptivo para identificar patrones y tendencias. La base de datos no presenta valores faltantes, por lo que no hay necesidad de eliminar filas. Se utiliza un diccionario de mapeo para transformar las variables categóricas, en la columna de los tipos (Type), en valores numéricos, para su mejor interpretación por los algoritmos de aprendizaje. Los datos se dividen en conjuntos de entrenamiento y prueba, este conjunto de entrenamiento también se divide en un conjunto de validación. Se estandariza las características utilizando StandardScaler (una librería de sklearn en python), para garantizar que todas las características contribuyan equitativamente al proceso de entrenamiento.

Los fallos en el dataset representan el 3.4 % del total, como se puede apreciar en la Figura 1. Algunos tipos de fallos se presentan con mayor frecuencia que otros; por ejemplo, los fallos por disipación de calor son más comunes, mientras que los fallos aleatorios y los fallos por desgaste de herramientas son menos frecuentes (Figura 2).

Además, el dataset incluye información sobre los tipos de productos, lo que permite observar su distribución en relación con los diferentes tipos de fallos

(Figura 3). Se destaca que los productos de baja calidad son los más afectados por los fallos, mientras que los productos de alta calidad tienden a presentar más fallos relacionados con la disipación de calor.

2.2. Modelo de Bosques Aleatorios

El bosque aleatorio es un algoritmo de aprendizaje automático que se compone de varios árboles de decisión. Un árbol de decisión es un modelo compuesto por una colección de preguntas organizadas de forma jerárquica en forma de árbol. Cada nodo que no es hoja contiene una condición, y cada nodo hoja contiene una predicción. La inferencia de un modelo de árbol de decisión se calcula enrutiando un ejemplo de la raíz a uno de los nodos de hoja según las condiciones. El valor de la hoja alcanzada es la predicción del árbol de decisión. Mientras que los árboles de decisión consideran todas las posibles divisiones de características, los bosques aleatorios solo seleccionan un subconjunto de esas características, por lo que hay una baja correlación entre los árboles de decisión.

Se utiliza este algoritmo para la predicción de fallos en combinación con el clasificador MultiOutputClassifier para abordar un problema de clasificación multietiqueta, donde se le aplica a cada etiqueta por separado. Al establecer el parámetro "balanced", el modelo ajusta automáticamente los pesos de las clases.

2.3. Modelo de Red Neuronal MLP

Un perceptrón multicapa (MLP) es un tipo de red neuronal artificial formada por varias capas de neuronas. Las neuronas del MLP suelen utilizar funciones de activación no lineales, lo que permite a la red aprender patrones complejos en los datos. La red suele organizarse en capas, empezando por la capa de entrada, donde se introducen los datos. Le siguen las capas ocultas, donde se realizan los cálculos y, por último, la capa de salida, donde se hacen las predicciones o se toman las decisiones. Las neuronas de capas adyacentes están conectadas mediante conexiones ponderadas, que transmiten señales de una capa a la siguiente. La fuerza de estas conexiones, representada por los pesos, determina cuánta influencia tiene la salida de una neurona sobre la entrada de otra.

Cada neurona de las capas ocultas y de la capa de salida aplica una función de activación a su suma ponderada de entradas. En este caso para predecir los fallos se utiliza en la entrada la cantidad de características del conjunto de entrenamiento, que serían 6. Se crea un modelo secuencial, que es una forma de apilar capas de manera lineal. Esto significa que la salida de una capa se convierte en la entrada de la siguiente. Hay una capa densa con 128 neuronas y utiliza la función de activación ReLU (Unidad lineal rectificada), que ayuda a introducir no linealidades en el modelo. En otra capa se aplica la técnica de regularización conocida como "*dropout*", el 20% (0.2) de las neuronas se desactivan aleatoriamente para prevenir el sobreajuste, lo que ayuda que el modelo generalice mejor

a datos no vistos. Tiene otra capa densa con 64 neuronas y activación ReLU. La capa final del modelo 5 neuronas, donde cada neurona corresponde a una clase que el modelo debe predecir, que serian los 5 tipos de fallos. Esta capa final tiene como función de activación una función sigmoide, donde se genera una probabilidad independiente y se establece un umbral (0.5) para decidir si se considera que se va a producir el fallo o no.

2.4. Evaluación del modelo

La evaluación del modelo de Bosques Aleatorios se realiza mediante el uso del `classification_report`, que incluye las siguientes métricas:

- Precisión: Proporción de verdaderos positivos sobre el total de predicciones positivas.
- Recall (Sensibilidad): Proporción de verdaderos positivos sobre el total de verdaderos positivos y falsos negativos.
- F1-score: Media armónica entre la precisión y el recall, que proporciona una única métrica para evaluar el rendimiento del modelo.

También se utiliza una matriz de confusión, donde se desglosa el número real de cada etiqueta frente al número predicho por el modelo. Esto permite visualizar las predicciones correctas e incorrectas, facilitando su comprensión.

En el caso de la red neuronal se observa el valor *binary accuracy*. Es útil cuando las salidas del modelo son probabilidades (salida de una sigmoide) y se necesita una métrica que interprete estas como clases binarias. Para esto utiliza un umbral de 0.5, luego compara estas etiquetas binarias con las verdaderas.

3. Resultados

En el primer modelo se obtuvieron los siguientes resultados en las métricas de evaluación (Cuadro 1). El modelo funciona bien en HDF, PWF y OSF, pero falla completamente en TWF y RNF, que son las de menor muestra. El rendimiento en general es moderado (F1-score micro de 0.60), con un problema para generalizar a todas las clases por igual.

En el Cuadro 2 se muestran los resultados de la evaluación de las métricas mencionadas anteriormente, donde se puede ver un comportamiento bastante similar, dejando en claro el problema en las clases minoristas.

4. Conclusiones

El proyecto de detección de tipos de fallos ha logrado desarrollar un sistema capaz de identificar eficazmente los fallos más comunes en el proceso. Sin embargo, se han identificado limitaciones fundamentales que impiden su implementación completa en su estado actual. Se recomienda abordar el desbalance

de clases y realizar una calibración adecuada de los umbrales. Además, es importante analizar los errores de los modelos en relación con los diferentes tipos de fallos, utilizando la matriz de confusión que se presenta en el apéndice de figuras.

5. Apéndice de Figuras

Class	Precision	Recall	F1-Score	Support
TWF	0.00	0.00	0.00	14
HDF	0.94	0.50	0.65	34
PWF	0.84	0.55	0.67	29
OSF	0.94	0.59	0.72	29
RNF	0.00	0.00	0.00	6
micro avg	0.91	0.45	0.60	112
macro avg	0.55	0.33	0.41	112
weighted avg	0.75	0.45	0.56	112
samples avg	0.02	0.02	0.02	112

Cuadro 1: Métricas de rendimiento del modelo de Bosque Aleatorios

Class	Precision	Recall	F1-Score	Support
TWF	0.00	0.00	0.00	14
HDF	0.77	0.49	0.60	35
PWF	0.68	0.59	0.63	29
OSF	0.92	0.79	0.85	29
RNF	0.00	0.00	0.00	6
micro avg	0.79	0.50	0.62	113
macro avg	0.47	0.37	0.42	113
weighted avg	0.65	0.50	0.56	113
samples avg	0.02	0.02	0.02	113

Cuadro 2: Métricas de rendimiento del clasificador

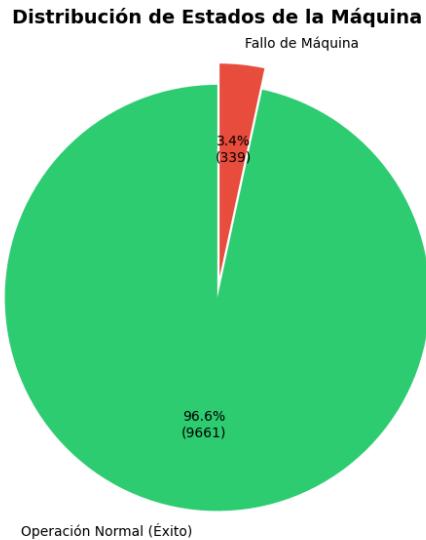


Figura 1: Distribución de los fallos de la máquina.

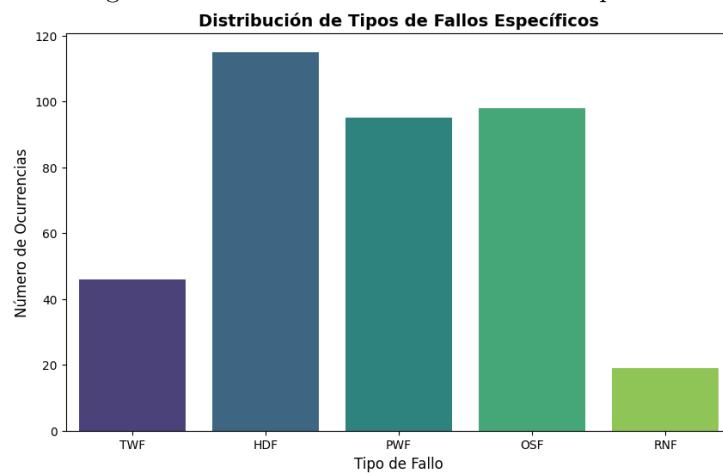


Figura 2: Distribución de los tipos de fallos.

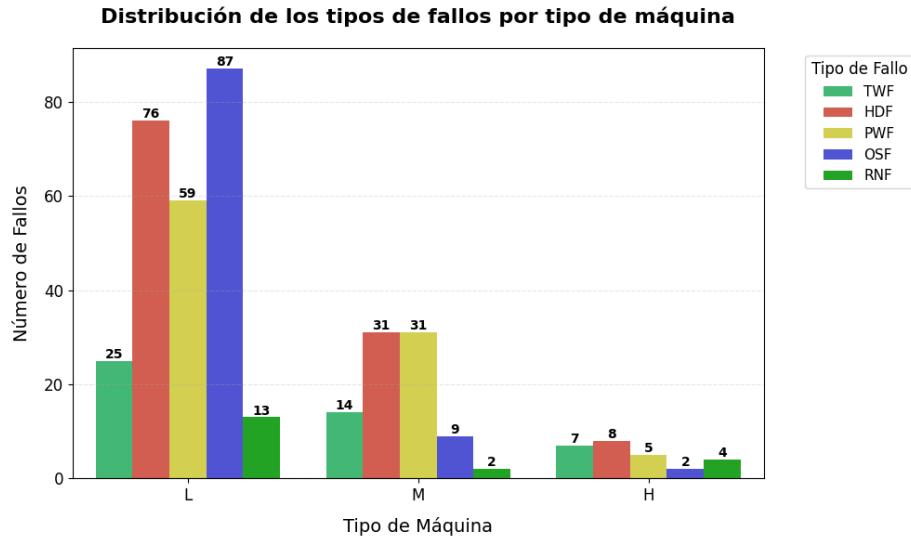


Figura 3: Distribución de los tipos de fallos por el tipo de máquina.

```
binary_accuracy: 0.9972 - f1_score: 0.7914 - loss: 0.0089 - precision: 0.8897 - recall: 0.7127
```

```
binary_accuracy: 0.9974 - f1_score: 0.8012 - loss: 0.0088 - precision: 0.9149 - recall: 0.7127
```

```
binary_accuracy: 0.9972 - f1_score: 0.7848 - loss: 0.0086 - precision: 0.9185 - recall: 0.6851
```

Figura 4: Métricas de evaluación de MLP en el entrenamiento.

```
val_binary_accuracy: 0.9954 - val_f1_score: 0.6364 - val_loss: 0.0160 - val_precision: 0.7925 - val_recall: 0.5316
```

```
val_binary_accuracy: 0.9956 - val_f1_score: 0.6567 - val_loss: 0.0157 - val_precision: 0.8000 - val_recall: 0.5570
```

```
val_binary_accuracy: 0.9957 - val_f1_score: 0.6763 - val_loss: 0.0160 - val_precision: 0.7833 - val_recall: 0.5949
```

Figura 5: Métricas de evaluación de MLP de la validación

Confusion matrix TWF

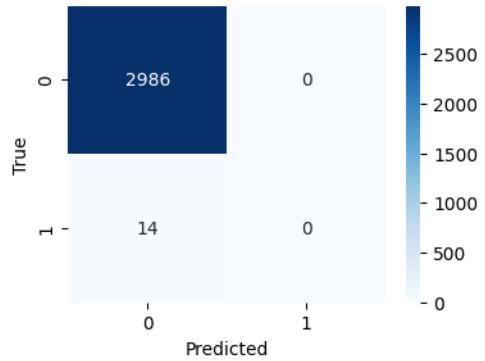


Figura 6: Matriz de confusión de TWF en MLP

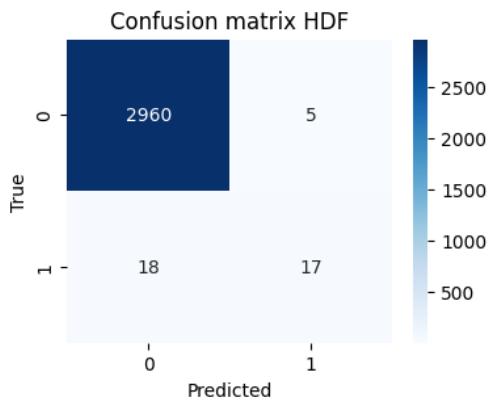


Figura 7: Matriz de confusión de HDF en MLP

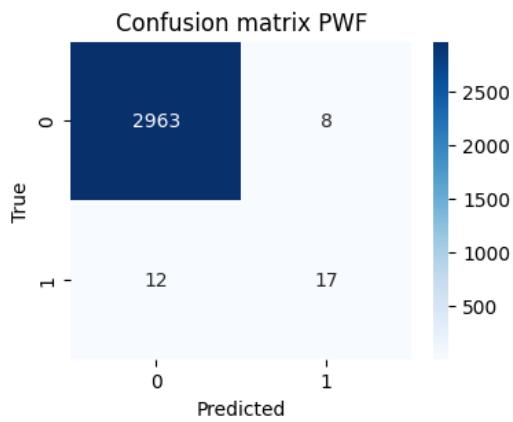


Figura 8: Matriz de confusión de PWF en MLP

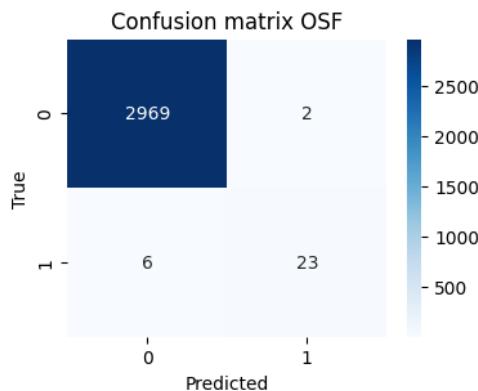


Figura 9: Matriz de confusión de OSF en MLP

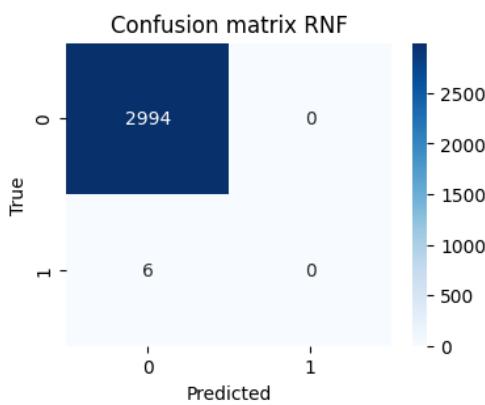


Figura 10: Matriz de confusión de RNF en MLP

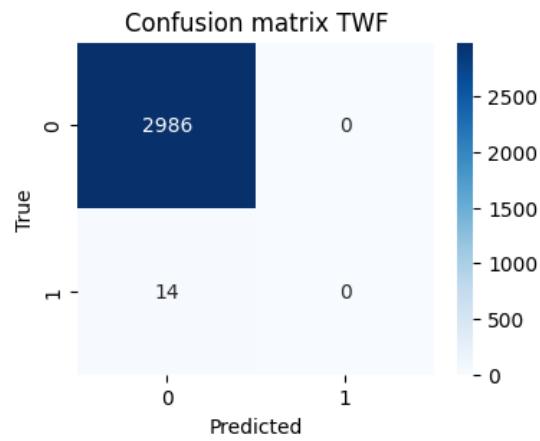


Figura 11: Matriz de confusión de TWF en Bosques Aleatorios

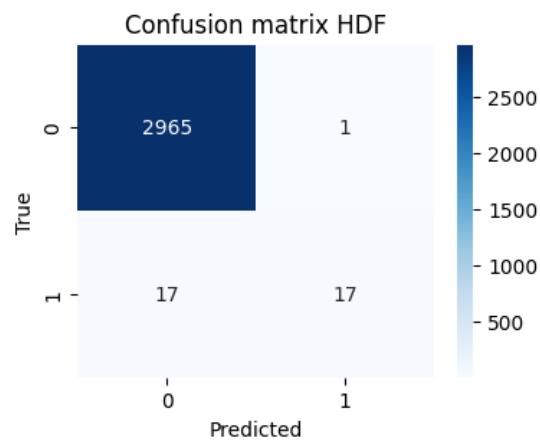


Figura 12: Matriz de confusión de HDF en Bosques Aleatorios

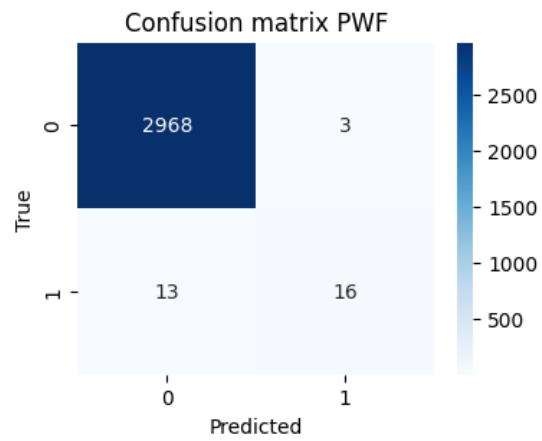


Figura 13: Matriz de confusión de PWF en Bosques Aleatorios

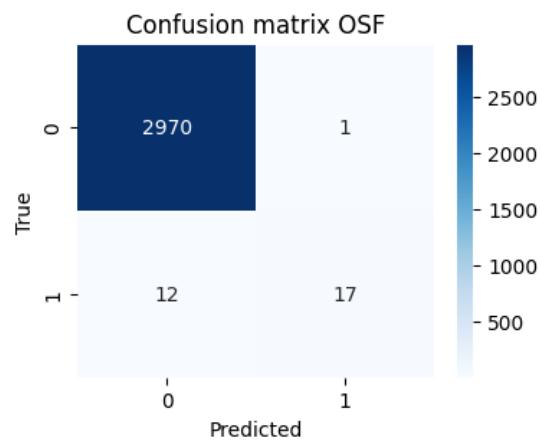


Figura 14: Matriz de confusión de OSF en Bosques Aleatorios

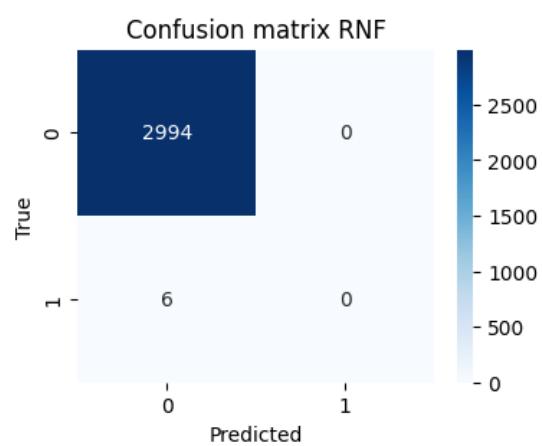


Figura 15: Matriz de confusión de RNF en Bosques Aleatorios