



# **Predicción de errores de maquinaria.**

Jose Ramon Casas

# Problema a resolver

El problema principal es predecir cuando la máquina va a fallar, para anticiparnos y...

... minimizar costes de reparación, mantenimiento y producción.

... evitar inactividad.

... cumplir con compromisos de producción.



# Datos

- 10000 registros.
- Limpios, sin valores nulos
- Bien estructurados.
- No consistentes en los fallos y tipos de fallos

		DATOS UTILIZADOS PARA EL CALCULO DE NUESTROS MODELOS							OBJETIVOS A PREDECIR					
	UDI	Product ID	Type	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]	Machine failure	TWF	HDF	PWF	OSF	RNF
0	1	M14860	M	298.1	308.6	1551	42.8	0	0	0	0	0	0	0
1	2	L47181	L	298.2	308.7	1408	46.3	3	0	0	0	0	0	0
2	3	L47182	L	298.1	308.5	1498	49.4	5	0	0	0	0	0	0
3	4	L47183	L	298.2	308.6	1433	39.5	7	0	0	0	0	0	0
4	5	L47184	L	298.2	308.7	1408	40.0	9	0	0	0	0	0	0

# Fallo en máquina y distintos tipos fallos

Ajuste de cantidades

“Machine Failure”

→ 339

Suma Tipos de fallos TWF, HDF, PWF, OSF, RNF

→ 373

Creación de nueva variable

Múltiples tipos de error: Múltiple Fallos → “MF”

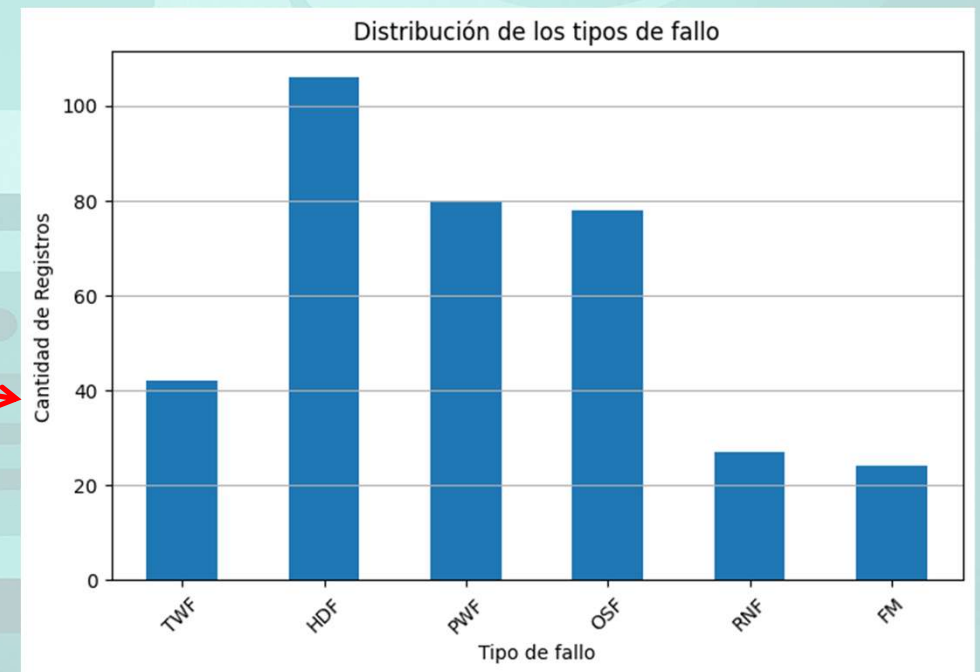
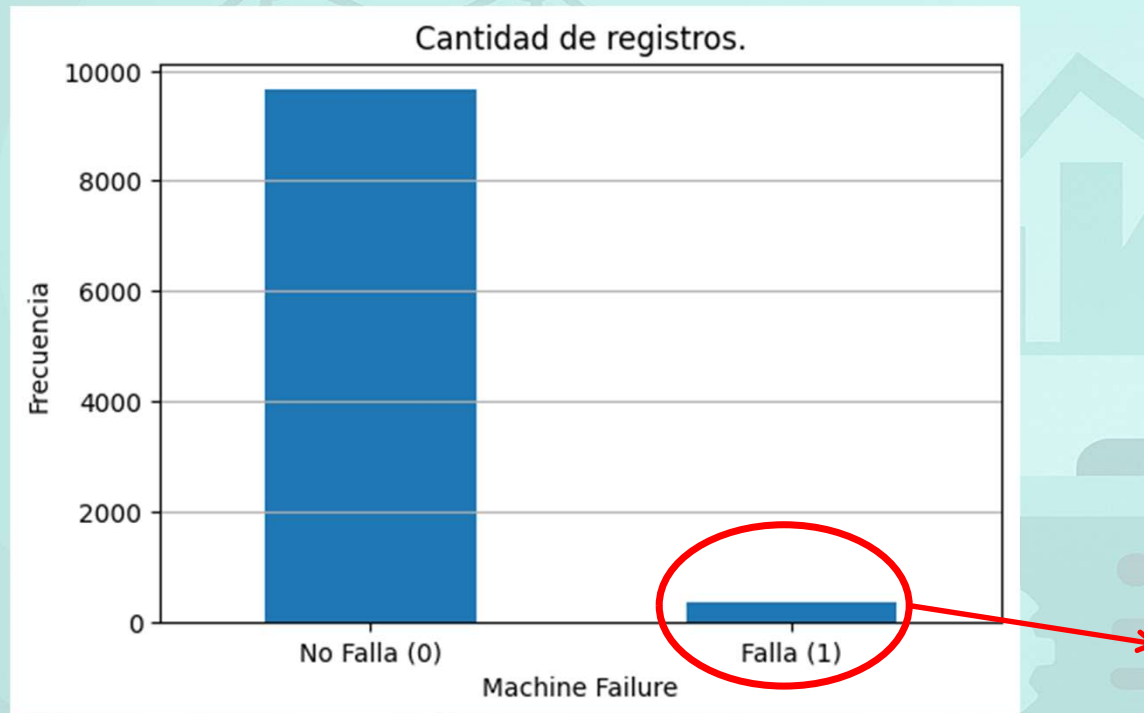
Después de hacer los ajustes tenemos:

Machine Failure”

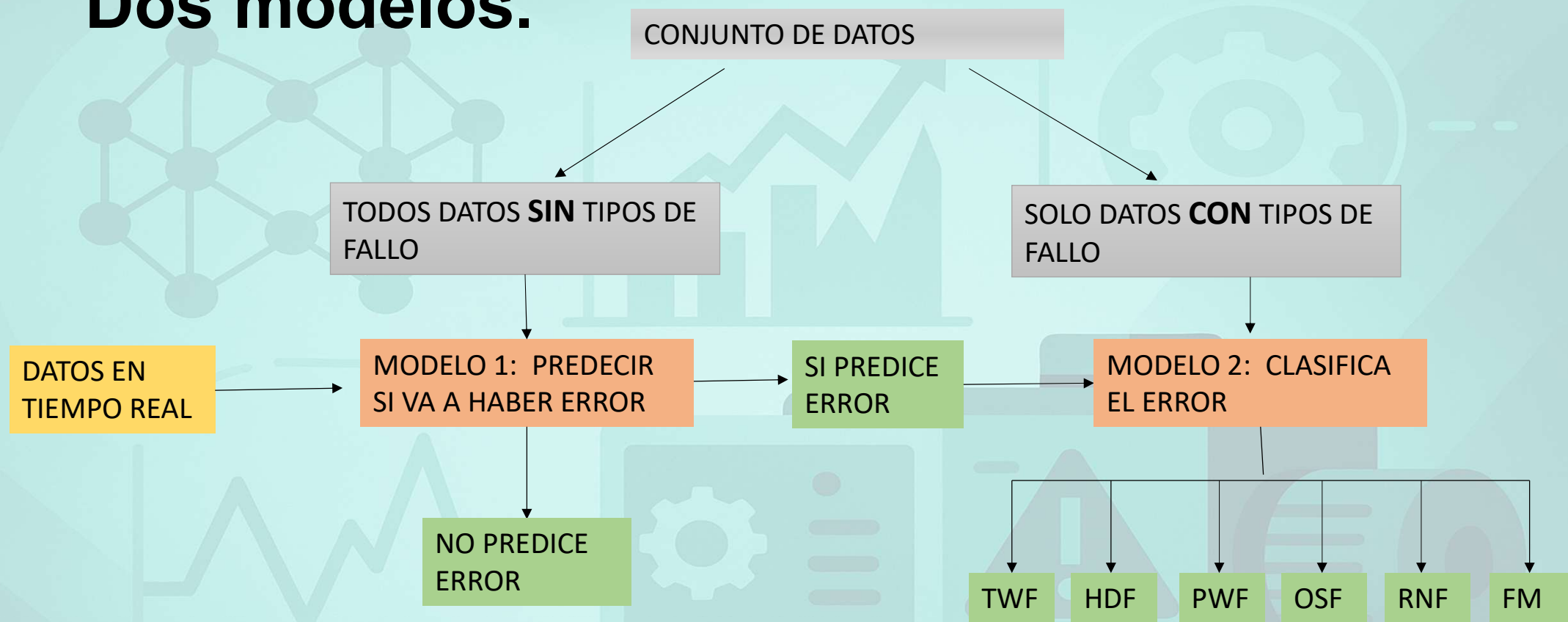
→ 357

Suma Tipos de fallos (TWF, HDF, PWF, OSF, RNF, MF) → 357

# Datos desbalanceados



# Dos modelos.



# Tratamiento de variables

Nuevas variables

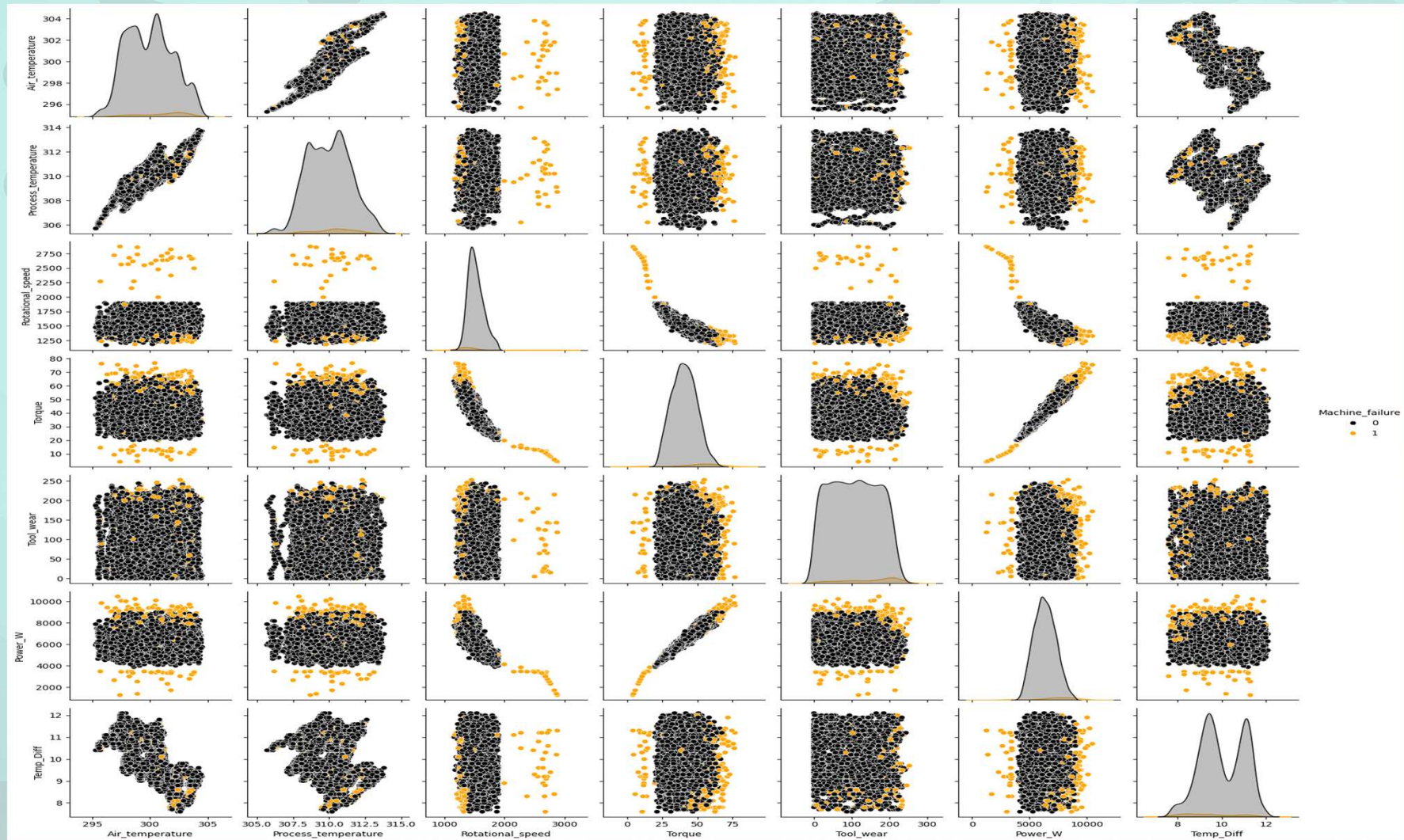
- $\text{Potencia} = (2\pi * \text{Torque} * \text{Velocidad de rotación})/60$
- $\text{Diferencia de temperatura} = \text{Temp. proceso} - \text{Temp. ambiente}$

Transformar la variable Type

- 3 columnas con valor 0,1 dependiendo del tipo H, M, L

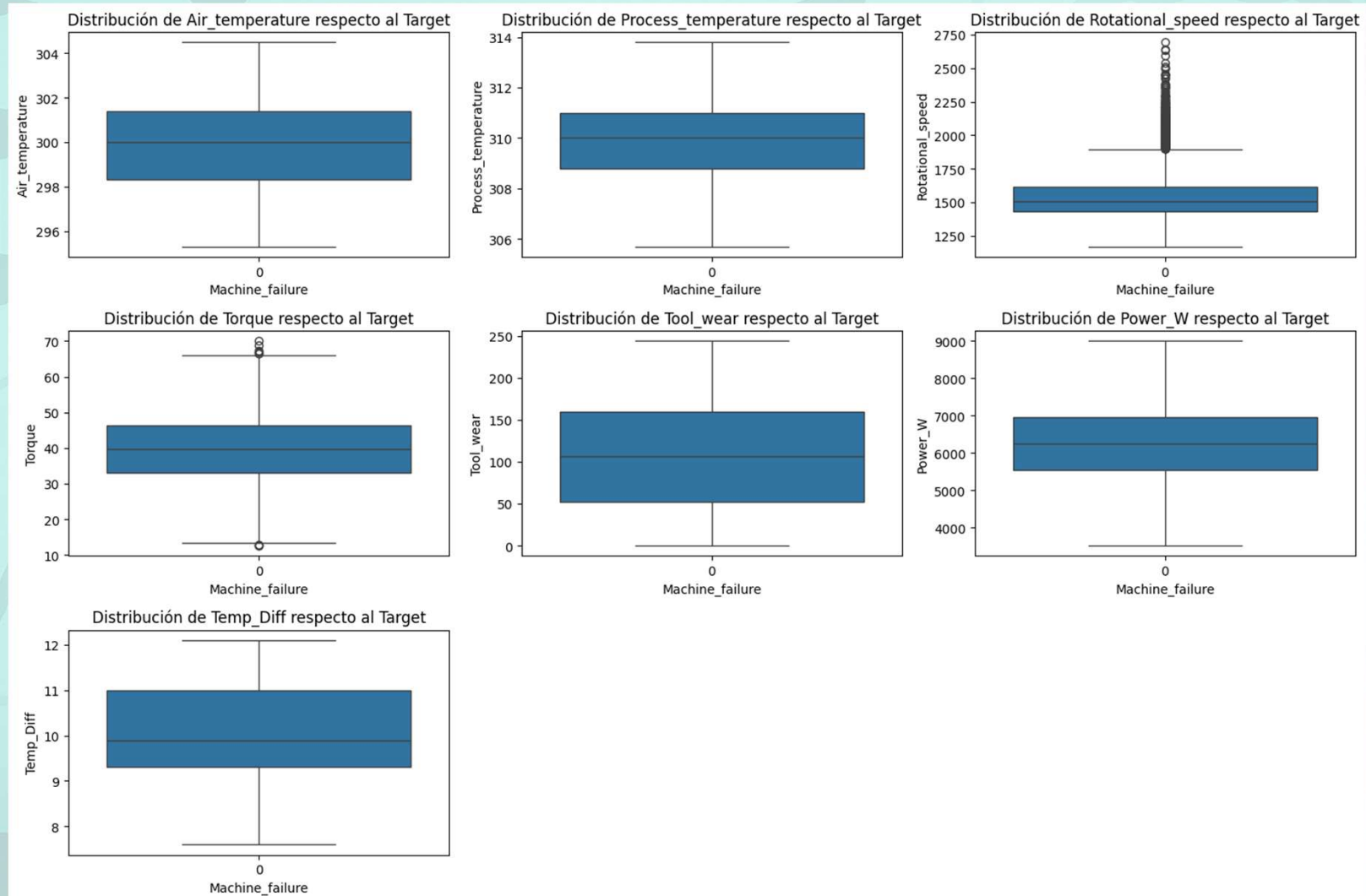


# Distribución





# Tratamiento de Outliers (solo registros sin fallo)



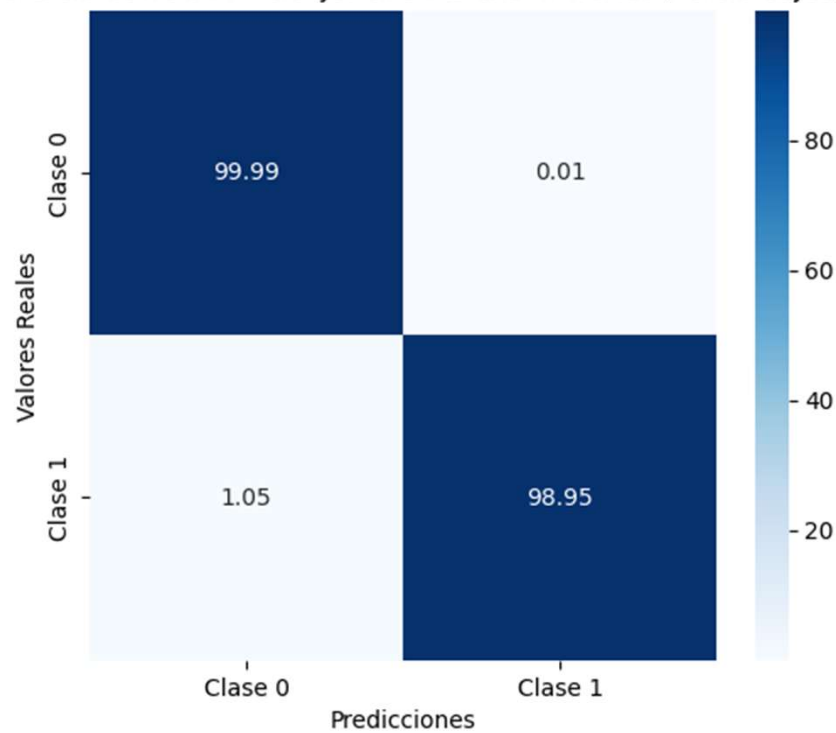
# Entrenamiento primero modelo

Balanceo mediante SMOTE

Random Forest Classifier

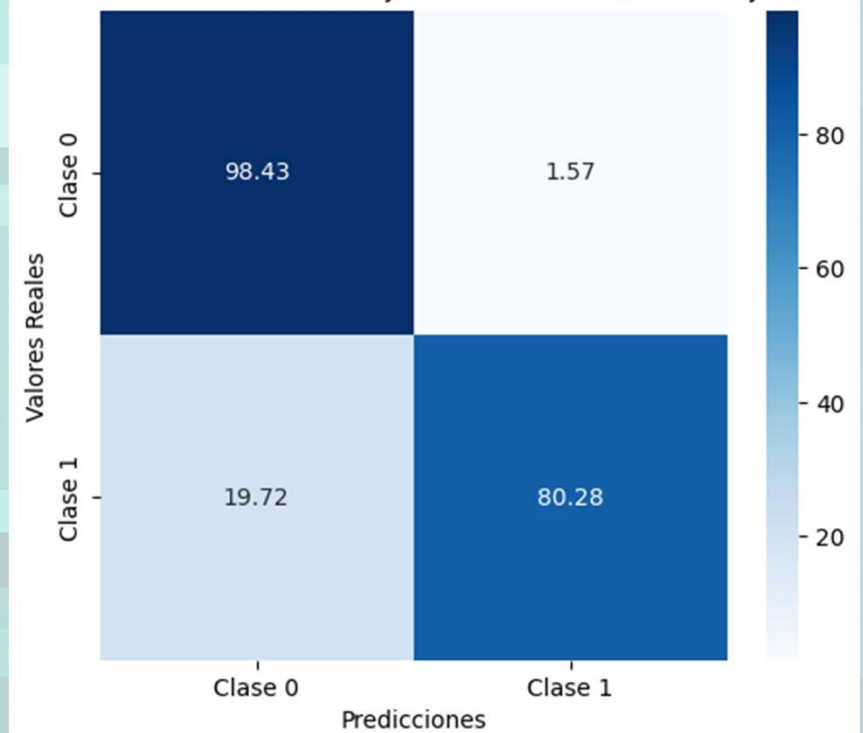
TEST

Matriz de Confusión - Conjunto de Entrenamiento (Porcentajes)



TRAIN

Matriz de Confusión - Conjunto de Prueba (Porcentajes)



# Ajustar predicción de fallos

Variable a clasificar (target):

0 → FM → FALLO MULTIPLE

1 → RNF → FALLO ALEATORIO

2 → OSF → FALLO SOBRESFUERZO

3 → PWF → FALLO DE POTENICA

4 → HDF → FALLO POR DISIPACIÓN DE CALOR

5 → TWF → FALLO POR DESGASTE DE HERRAMIENTA

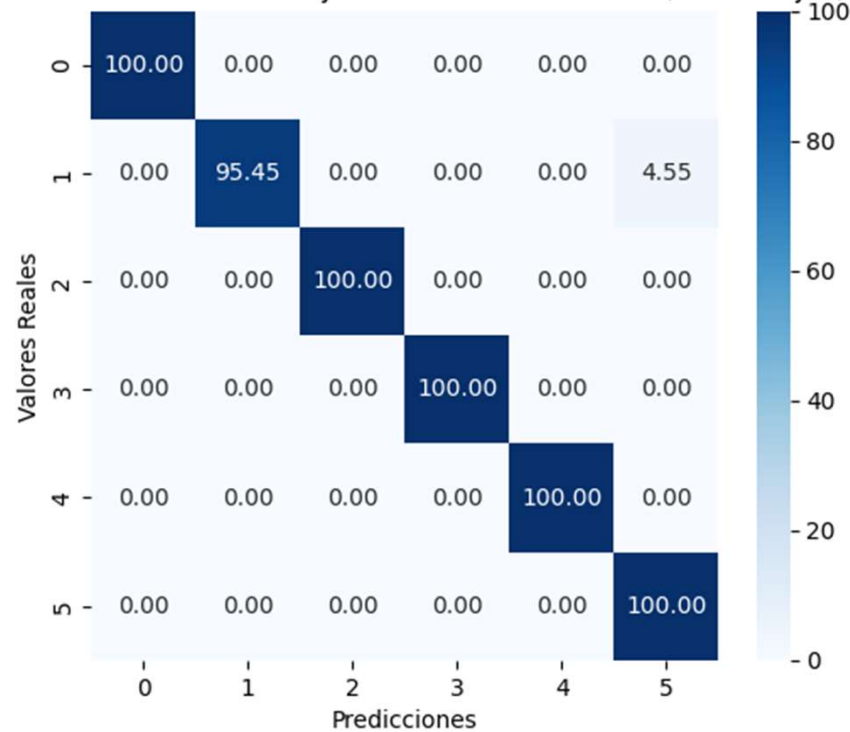
# Entrenamiento segundo modelo

Balanceo mediante SMOTE

XGBOOST

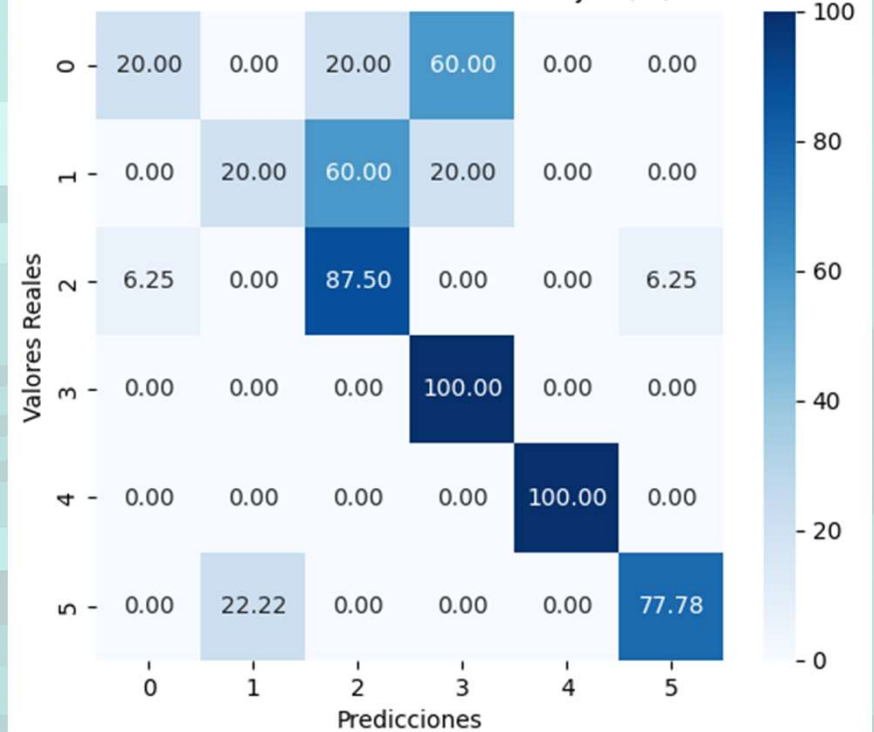
TEST

Matriz de Confusión - Conjunto de Entrenamiento (Porcentajes)



TRAIN

Matriz de Confusión - Porcentajes (%)



## Conclusiones

El primer modelo aprende bien, tiene una buena métrica de evaluación del 89% contra los datos de test.

Es capaz de predecir el 80 % verdaderos errores.

El segundo modelo no tiene tan buena métrica de evaluación un 67%,  
Le cuesta clasificar las categorías de “error múltiple” y los “fallos aleatorios”. Estas dos categorías penalizan el modelo, en cambio el resto la clasificación es óptima