

# MANTENIMIENTO PREDICTIVO MOTORES DE AVIÓN:

## MODELO MACHINE LEARNING

PRESENTACIÓN TÉCNICA

Leire Alonso Polanco

7 de Julio 2025



# ÍNDICE

- 1- INTRODUCCIÓN
- 2- EXPLORACIÓN DE LOS DATOS (EDA)
- 3- DIVISIÓN DEL DATASET
- 4- MODELOS ENTRENADOS
- 5- EVALUACIÓN DE LOS MODELOS
- 6- MODELO FINAL SELECCIONADO
- 7- VALIDACIÓN Y PRUEBAS FINALES
- 8- CONCLUSIONES
- 9- PRÓXIMOS PASOS / MEJORAS



# *1- INTRODUCCIÓN*

## MANTENIMIENTO CORRECTIVO

- Después de que ocurre un fallo.
- Es el tipo más costoso a largo plazo, debido a paradas imprevistas y daños colaterales.

## MANTENIMIENTO PREVENTIVO

- Programado, con base en el tiempo o uso.
- Evitar fallos realizando revisiones o sustituciones antes de que algo se dañe.

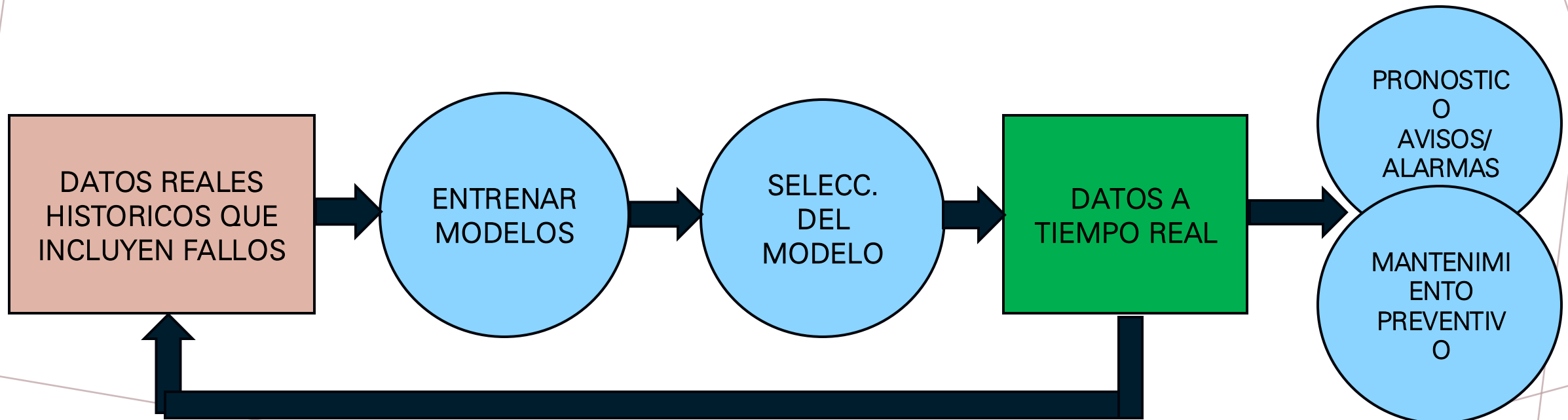
## MANTENIMIENTO PREDICTIVO

- Datos en tiempo real y modelos predictivos para anticipar fallos.
- Sensores y Machine Learning para analizar el estado real del equipo.
- Solo se interviene cuando hay señales reales de fallo.

# *1- INTRODUCCIÓN*

## OBJETIVO DEL PROYECTO

Desarrollar un modelo predictivo mediante técnicas **Machine Learning** de aprendizaje automático que permita anticipar **posibles fallos en motores de avión**, con el fin de implementar soluciones proactivas a través de un sistema de **mantenimiento predictivo**. Este sistema facilitará la sustitución o reparación oportuna de componentes antes de que se produzca una avería.





# *1- INTRODUCCIÓN*

## OBJETIVO DEL PROYECTO

Desarrollar un modelo predictivo mediante técnicas **Machine Learning** de aprendizaje automático que permita anticipar **posibles fallos en motores de avión**, con el fin de implementar soluciones proactivas a través de un sistema de **mantenimiento predictivo**. Este sistema facilitará la sustitución o reparación oportuna de componentes antes de que se produzca una avería.

MENOR  
TIEMPO DE  
INACTIVIDAD

OPTIMIZ. DE  
RECURSOS

EXTENSIÓN  
DE LA VIDA  
UTIL

MEJOR  
PLANIF.  
OPERATIVA

DECISIONES  
BASADA EN  
DATOS

MEJORA EN  
LA CALIDAD  
DEL SERVICIO

REDUCCIÓN  
DE RIEGOS  
LABORALES

CUMPLIMIEN  
TO  
NORMATIVO

# *2- EXPLORACIÓN DE DATOS*

## *EDA*

### *DATASETS*

**DATASET\_1:** Parámetros de motores de aviones. Para simplificar, los nombres de las variables están codificados con una letra y un número.

Los datos son de 100 motores de avion y para cada uno se dan una serie de ciclos hasta que falla.

Cada fila representa un ciclo de operación de un motor, con un total de 26 columnas numéricas (variables).

20631 filas × 26 columnas

**DATASET\_2:** En que ciclo se da el fallo en cada motor.

100 filas × 2 columnas



# 2- EXPLORACIÓN DE DATOS

## EDA

### *DATASET\_1*

Variable	Inglés	Español
id	Engine ID	ID del motor
cycle	Time cycle / operating cycle	Ciclo de operación
setting1	Operational setting 1	Parámetro operativo 1
setting2	Operational setting 2	Parámetro operativo 2
setting3	Operational setting 3	Parámetro operativo 3
s1	Total temperature at fan inlet	Temperatura total en la entrada del ventilador
s2	LPC outlet temperature	Temperatura a la salida del compresor de baja presión
s3	HPC outlet temperature	Temperatura a la salida del compresor de alta presión
s4	LPT outlet temperature	Temperatura a la salida de la turbina de baja presión





# 2- EXPLORACIÓN DE DATOS

## EDA

### *DATASET\_1*

<b>s5</b>	Fan inlet pressure	Presión en la entrada del ventilador
<b>s6</b>	Bypass-duct pressure	Presión en el conducto de derivación (bypass)
<b>s7</b>	HPC outlet pressure	Presión a la salida del compresor de alta presión
<b>s8</b>	Physical fan speed	Velocidad física del ventilador
<b>s9</b>	Physical core speed	Velocidad física del núcleo
<b>s10</b>	Engine pressure ratio	Relación de presión del motor
<b>s11</b>	Static pressure at HPC outlet	Presión estática a la salida del compresor de alta presión
<b>s12</b>	Fuel flow / Ps30	Flujo de combustible relativo a Ps30
<b>s13</b>	Corrected fan speed	Velocidad corregida del ventilador
<b>s14</b>	Corrected core speed	Velocidad corregida del núcleo





# *2- EXPLORACIÓN DE DATOS*

## *EDA*

### *DATASET\_1*

<b>s15</b>	Bypass ratio	Relación de bypass
<b>s16</b>	Burner fuel-air ratio	Relación combustible-aire del quemador
<b>s17</b>	Bleed enthalpy	Entalpía del sangrado
<b>s18</b>	Demanded fan speed	Velocidad deseada del ventilador
<b>s19</b>	Demanded corrected core speed	Velocidad corregida deseada del núcleo
<b>s20</b>	HPT coolant bleed	Flujo de refrigerante en la turbina de alta presión
<b>s21</b>	LPT coolant bleed	Flujo de refrigerante en la turbina de baja presión

# *2- EXPLORACIÓN DE DATOS*

## *EDA*

### *DATOS*

- UNIFICACIÓN DE LAS DOS TABLAS

Se renombra la columna cycle de la segunda tabla para no confundirlas cuando se unan las dos tablas.

```
df_fallos = df_fallos.rename(columns={'cycle': 'failure_cycle'})
```

```
df = df.merge(df_fallos, on='id', how='left')
```

- CREACIÓN DE UNA COLUMNA NUEVA que dice si en ese ciclo hay fallo o no (0/1)

```
df['is_failure_cycle'] = (df['cycle'] == df['failure_cycle']).astype(int)
```



# *2- EXPLORACIÓN DE DATOS*

## *EDA*

### *DATOS*

- No existen valores nulos
- Necesidad de crear una target (se crean 3):

¿Cuántos ciclos quedan para el fallo?

se crea la variable RUL: Remaining Useful Life para cada ciclo.

Se crean una nueva variable "label 1" que da 1 desde el valor elegido para  $w_1$ . además se crea otra variable "label 2" donde se definen 3 niveles de operación.

label2: Variable multiclase, indica el estado del motor. Hay tres clases:

- "Ok" para motores con RUL mayor que  $w_1$ , da 0
- "Moderado" para motores con RUL menor o igual que  $w_1$  pero mayor que  $w_0$ , da 1
- "Critico" para motores con RUL menor o igual que  $w_0$ , da 2

En este caso se elige  $w_0 = 15$  y  $w_1 = 30$





# 2- EXPLORACIÓN DE DATOS

## EDA

### TABLA COMPLETA

#	Column	Non-Null Count	Dtype
0	id	20631 non-null	int64
1	cycle	20631 non-null	int64
2	setting1	20631 non-null	float64
3	setting2	20631 non-null	float64
4	setting3	20631 non-null	float64
5	s1	20631 non-null	float64
6	s2	20631 non-null	float64
7	s3	20631 non-null	float64
8	s4	20631 non-null	float64
9	s5	20631 non-null	float64
10	s6	20631 non-null	float64
11	s7	20631 non-null	float64
12	s8	20631 non-null	float64
13	s9	20631 non-null	float64
14	s10	20631 non-null	float64
15	s11	20631 non-null	float64

16	s12	20631 non-null	float64
17	s13	20631 non-null	float64
18	s14	20631 non-null	float64
19	s15	20631 non-null	float64
20	s16	20631 non-null	float64
21	s17	20631 non-null	int64
22	s18	20631 non-null	int64
23	s19	20631 non-null	float64
24	s20	20631 non-null	float64
25	s21	20631 non-null	float64
26	failure_cycle	20631 non-null	int64
27	is_failure_cycle	20631 non-null	int64
28	RUL	20631 non-null	float64
29	label1	20631 non-null	int64
30	label2	20631 non-null	int64





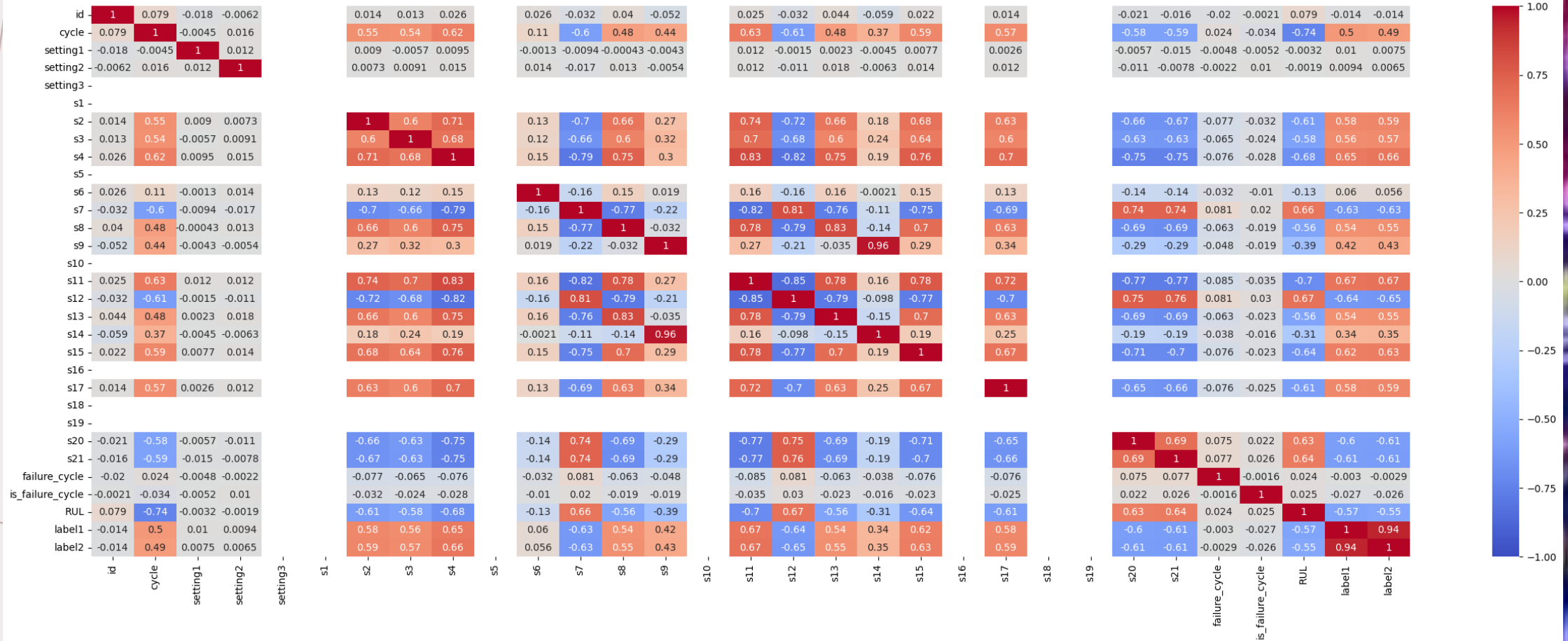
## *2- EXPLORACIÓN DE DATOS*

### *EDA*

### *NORMALIZACIÓN DE LOS DATOS*

- Se excluyen las columnas 'id', 'cycle', 'failure\_cycle', 'is\_failure\_cycle', 'RUL', 'label1', 'label2'.
- Se normaliza con MinMaxScaler



[illegible]

# *2- EXPLORACIÓN DE DATOS*

## *EDA*

### *LIMPIEZA DE DATOS*

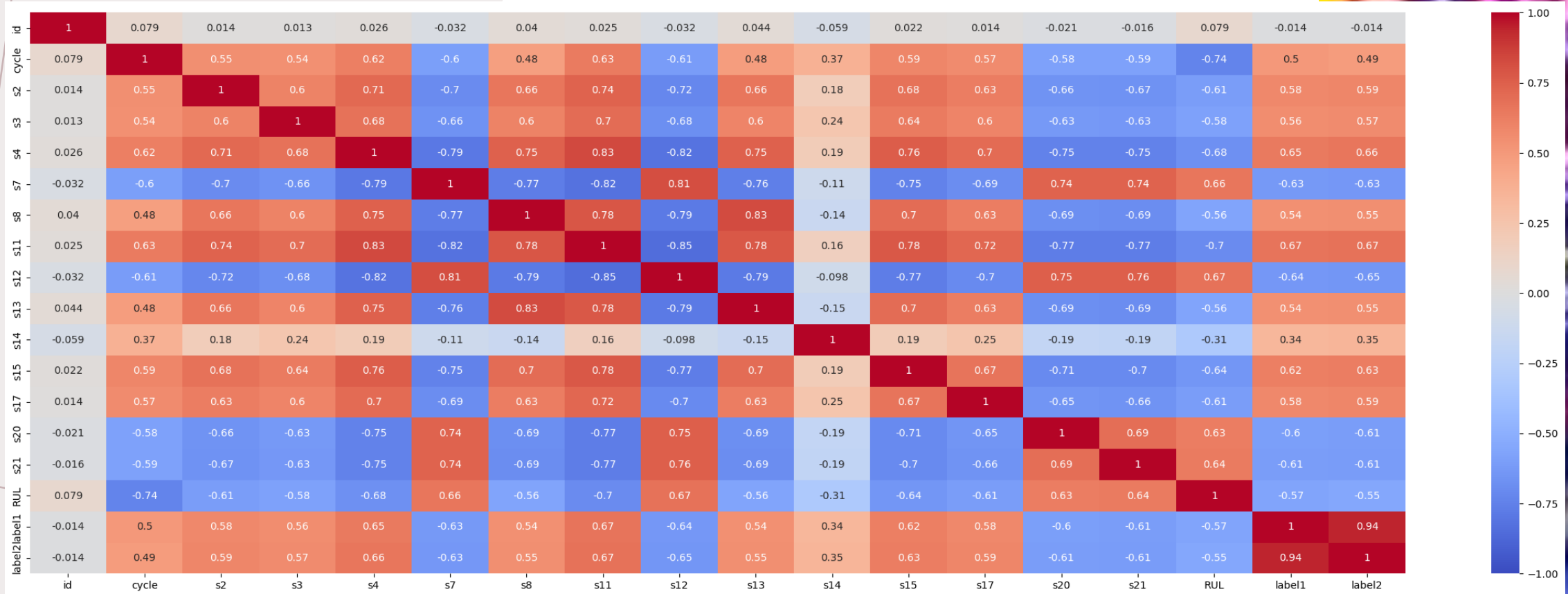
- No existen valores nulos
- Se eliminan variables:
  - o que tienen valores constantes: setting3, s1,s5,s10,s16,s18,s19.
  - que tienen una correlación muy alta entre sí: s9 y se deja s14.
  - variables que ya no son necesarias: failure\_cycle, is\_failure\_cycle.
  - variable con muy baja correlación con las posibles Target: s6



# 2- EXPLORACIÓN DE DATOS

## EDA

### HEATMAP CON DATOS LIMPIOS

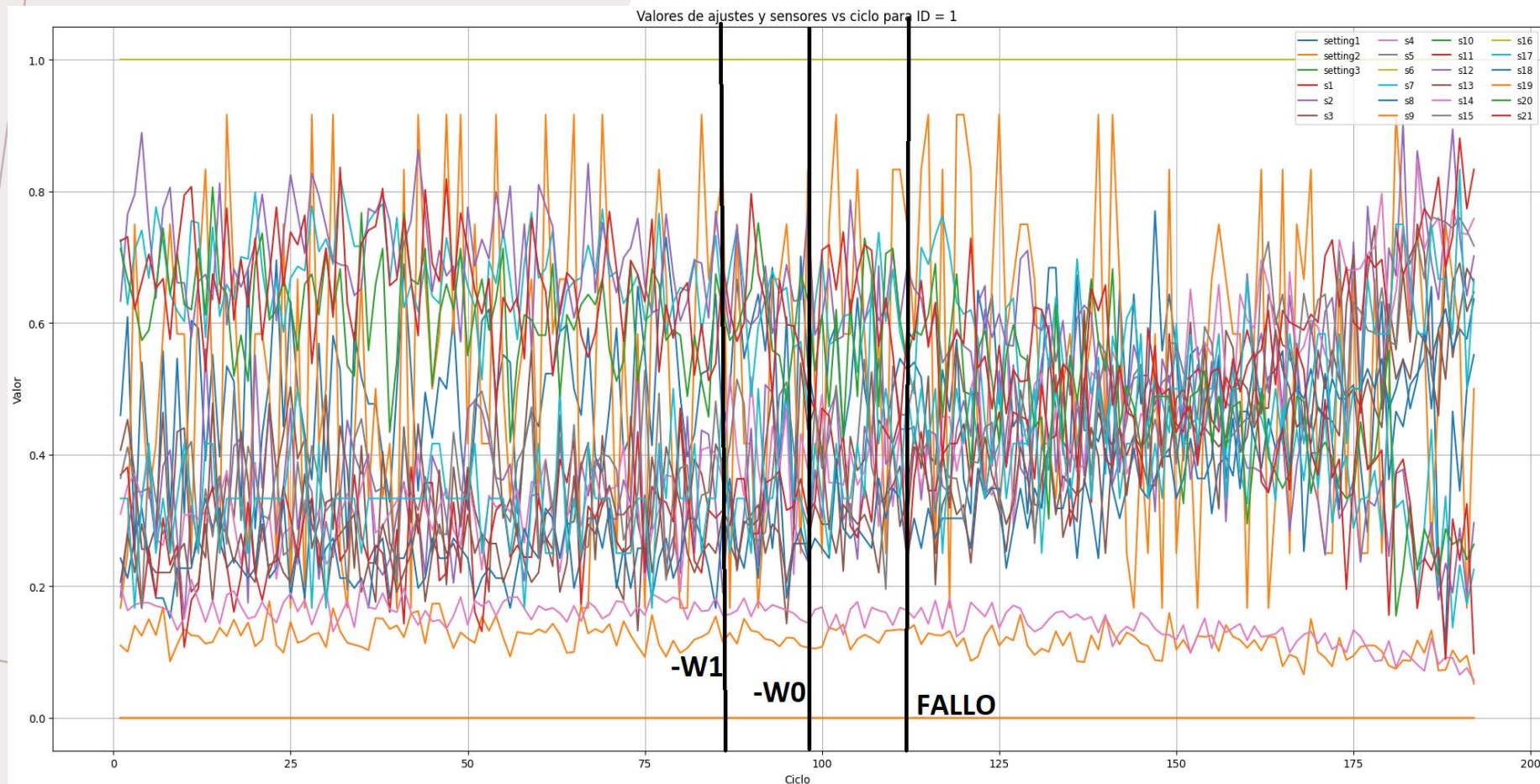




# 2- EXPLORACIÓN DE DATOS

## EDA

### VISUALIZACIONES



# *2- EXPLORACIÓN DE DATOS*

## *EDA*

*MANTENIMIENTO PREDICTIVO:  
MOTORES DE AVIÓN*

### *REDUCCIÓN DE VALORES OK*

Al revisar la distribución de los datos se ve que hay una mayoría de ciclos OK, **label2=0**

Los datos disponibles son 17531 en modo OK, 1500 en modo MODERADO y 1600 en modo CRITICO.

```
label2
0 17531
1 1500
2 1600
```

Con el objetivo de mejorar las predicciones se probará a reducir en diferentes porcentajes el numero de ciclos OK, quitando los primeros ciclos de cada ID o motor.

Se hará para cada modelo para comprobar que reducción es la optima.

```
label2
0 10231
1 1500
2 1600
```

## *3-DIVISIÓN DEL DATASET*

### SEPARACIÓN DE DATOS TRAIN /TEST

Para X se toman todas la variables menos 'id', 'cycle', 'label1', 'label2', 'RUL'  
Para y, la Target se toma 'label2'

```
X = df_filtered.drop(columns=['id', 'cycle', 'label1', 'label2', 'RUL'])  
y = df_filtered['label2']
```

Se toma el 20% de los datos para la parte de test y por tanto el 80% para entrenar los modelos:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=10)
```

Train: (10664, 13)

Test: (2667, 13)

## *4-MODELOS ENTRENADOS*

- MODELOS SUPERVISADOS:
  - LOGISTIC REGRESION
  - RANDOM FOREST
  - PIPELINE
  - SVM
  - RED NEURONAL
- MODELO NO SUPERVISADO:
  - K-MEANS





## *4- MODELOS ENTRENADOS*

Regresión Logística	Interpretabilidad + costos computacionales bajos + predicción probabilística
Random Forest	Robusto, preciso, mide importancia de sensores
SVM	Ideal para alta dimensionalidad y fronteras complejas
Red Neuronal	Capacidad de capturar no linealidades profundas
Pipeline	Orden y reproducibilidad del proceso completo
K-Means (no supervisado)	Para descubrir agrupaciones naturales y validar patrones relacionados con Label2

## *4- MODELOS ENTRENADOS*

### Modelos Supervisados

#### 1. Regresión Logística

- Ideal para problemas de clasificación multinomial como tu Label2 (clases 0, 1 y 2).
- Produce probabilidades interpretables y gestiona eficientemente datos linealmente separables.
- Es rápido y fácil de entender, lo que facilita explicar qué variables influyen en la predicción

#### 2. Random Forest

- Combina múltiples árboles para reducir el sobreajuste y mejorar precisión en datos tabulares .
- Proporciona medidas de importancia de variables, útiles para entender qué sensores o parámetros predicen mejor Label2.
- Robusto frente a datos faltantes o ruido

#### 3. Pipeline

- Encapsula preprocesamiento (por ejemplo, escalado) y selección de modelo de forma ordenada y reproducible.
- Facilita experimentar con múltiples modelos (SVM, Logistic, Random Forest, KNN...), asegurando que el mismo escalado se aplica consistentemente
- Mejora eficiencia en validación cruzada y evita fugas de datos entre fases.

## *4- MODELOS ENTRENADOS*

### 4. Support Vector Machine (SVM)

- Potente tanto para clasificaciones lineales como no lineales mediante núcleos (kernels)
- Funciona bien con alta dimensionalidad de tus variables de sensores
- Suele ofrecer buena generalización (margen máximo) aunque es más lento de entrenar

### 5. Red Neuronal

- Útil para aprender relaciones complejas y no lineales entre sensores y la probabilidad de fallo.
- Capaz de extraer automáticamente características relevantes cuando los datos son abundantes.
- Puede alcanzar alta precisión aunque requiere más datos, cómputo y es más difícil de interpretar

### Modelo No Supervisado

### 6. K-Means Clustering

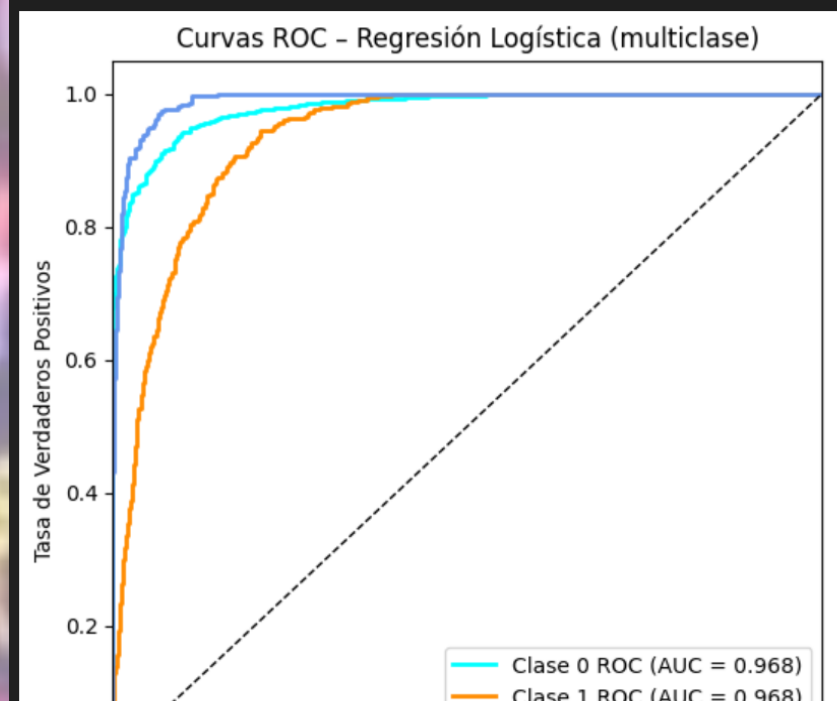
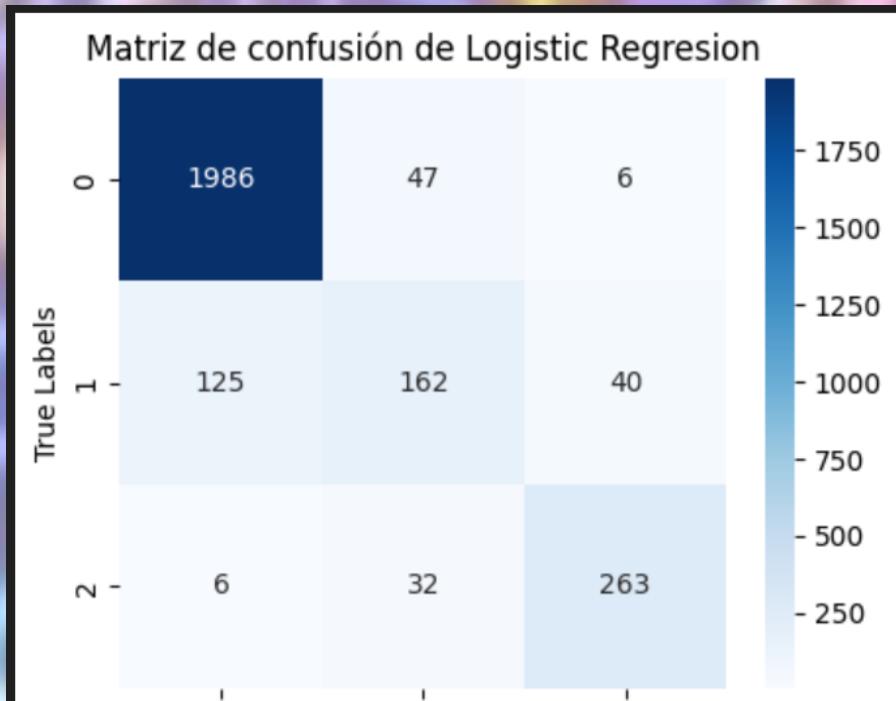
- Agrupa motores sin usar Label2, identificando patrones similares en su comportamiento.
- Útil como análisis exploratorio para validar si los grupos reflejan realmente la probabilidad de fallo (Label2).
- Puede usarse luego para diseñar alertas o estrategias de mantenimiento preventivo basadas en la pertenencia al cluster.
- **K-Means**, aunque no predice Label2 directamente, te ayuda a encontrar estructuras subyacentes en los datos que pueden correlacionar con el deterioro del motor y anticipar patrones de fallo.

# 5- EVALUACIÓN DE LOS MODELOS

## LOGISTIC REGRESION

0.9040119985001874

	precision	recall	f1-score	support
0	0.94	0.97	0.96	2039
1	0.67	0.50	0.57	327
2	0.85	0.87	0.86	301
accuracy			0.90	2667
macro avg	0.82	0.78	0.80	2667
weighted avg	0.90	0.90	0.90	2667





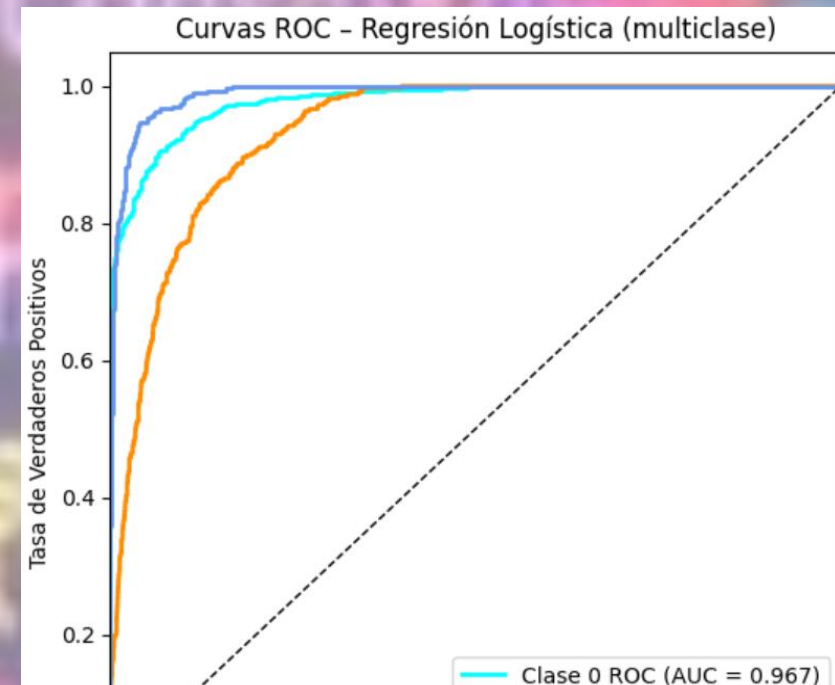
## 5- EVALUACIÓN DE LOS MODELOS

### RANDOM FOREST

Mejor combinación de parámetros:  
{'bootstrap': True, 'max\_depth': 10,  
'min\_samples\_leaf': 2, 'min\_samples\_split': 2,  
'n\_estimators': 300}

	precision	recall	f1-score	support
0	0.95	0.97	0.96	2039
1	0.68	0.54	0.60	327
2	0.84	0.85	0.85	301
accuracy			0.91	2667
macro avg	0.82	0.79	0.80	2667
weighted avg	0.90	0.91	0.90	2667

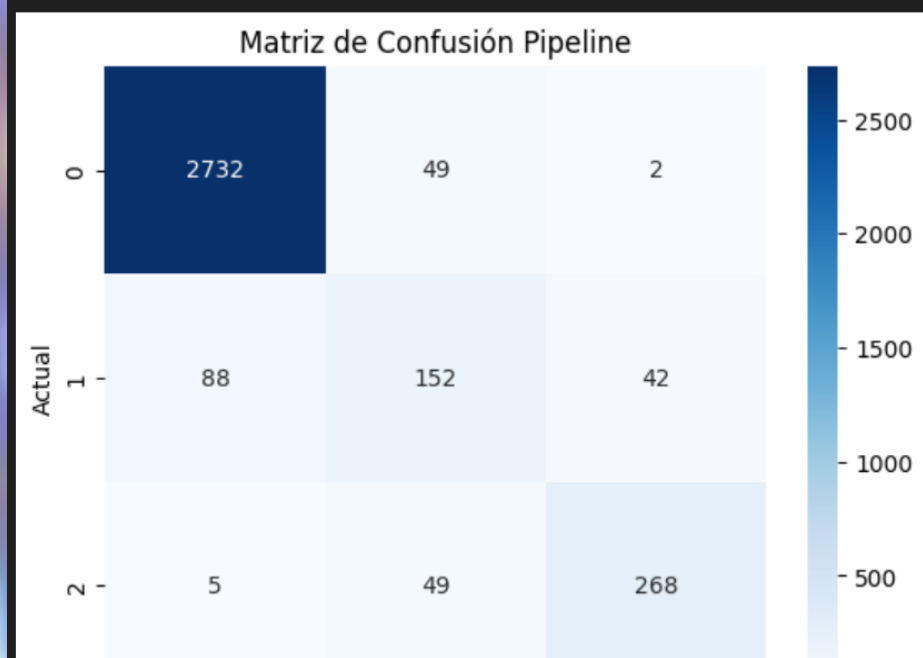
Matriz de Confusión Random Forest			
Reales	0	1	2
	1987	107	8
	48	177	37
	4	43	256



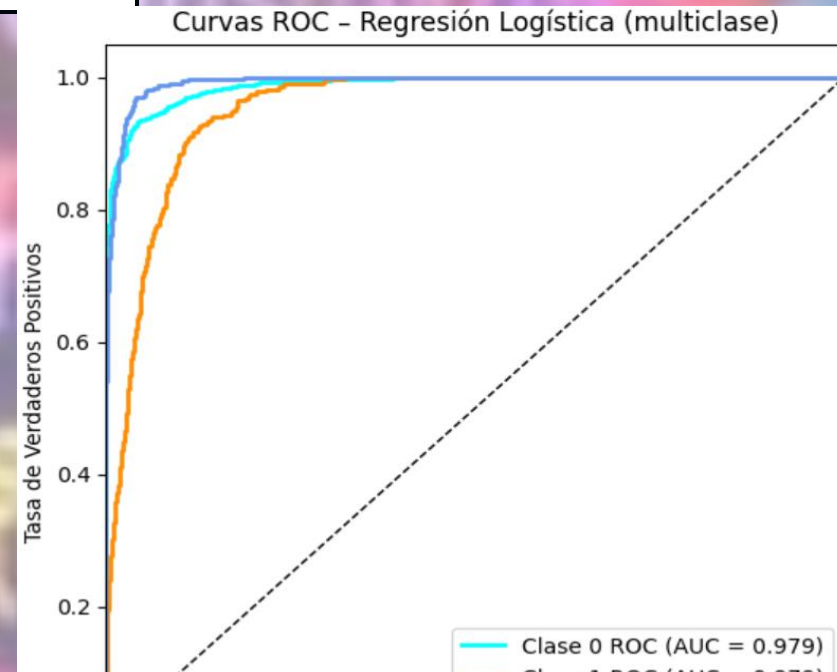
## 5- EVALUACIÓN DE LOS MODELOS

### PIPELINE

```
Mejor modelo: Pipeline(steps=[('scaler',  
StandardScaler()), ('clf',  
RandomForestClassifier(max_depth=10,  
n_estimators=300, random_state=0))])
```



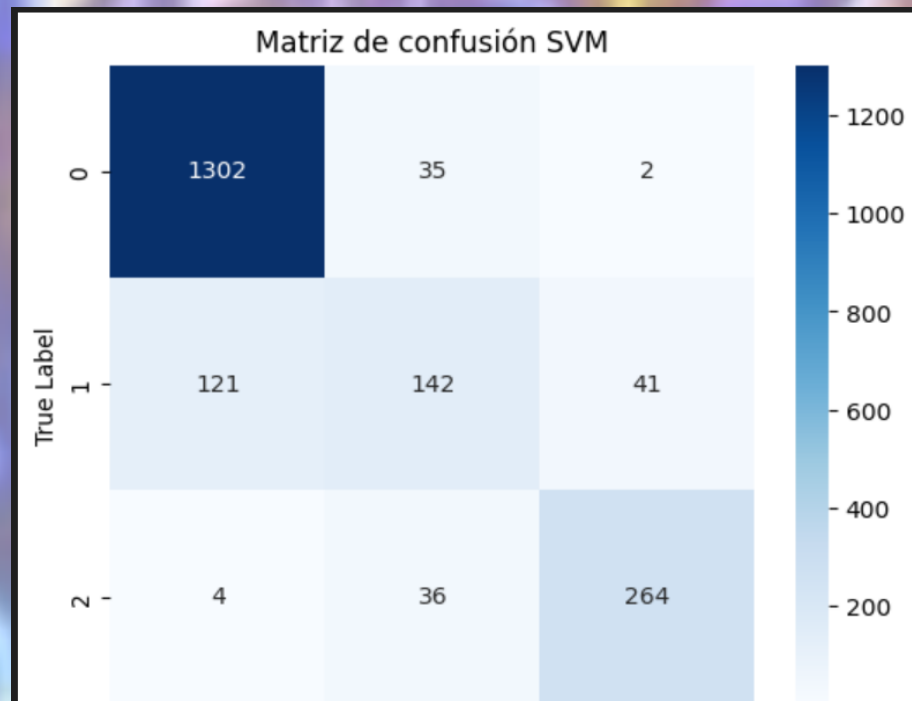
	precision	recall	f1-score	support
0	0.97	0.98	0.97	2783
1	0.61	0.54	0.57	282
2	0.86	0.83	0.85	322
accuracy			0.93	3387
macro avg	0.81	0.78	0.80	3387
weighted avg	0.93	0.93	0.93	3387



## 5- EVALUACIÓN DE LOS MODELOS

SVM

	precision	recall	f1-score	support
0	0.912	0.972	0.941	1339
1	0.667	0.467	0.549	304
2	0.860	0.868	0.864	304
accuracy			0.877	1947
macro avg	0.813	0.769	0.785	1947
weighted avg	0.866	0.877	0.868	1947

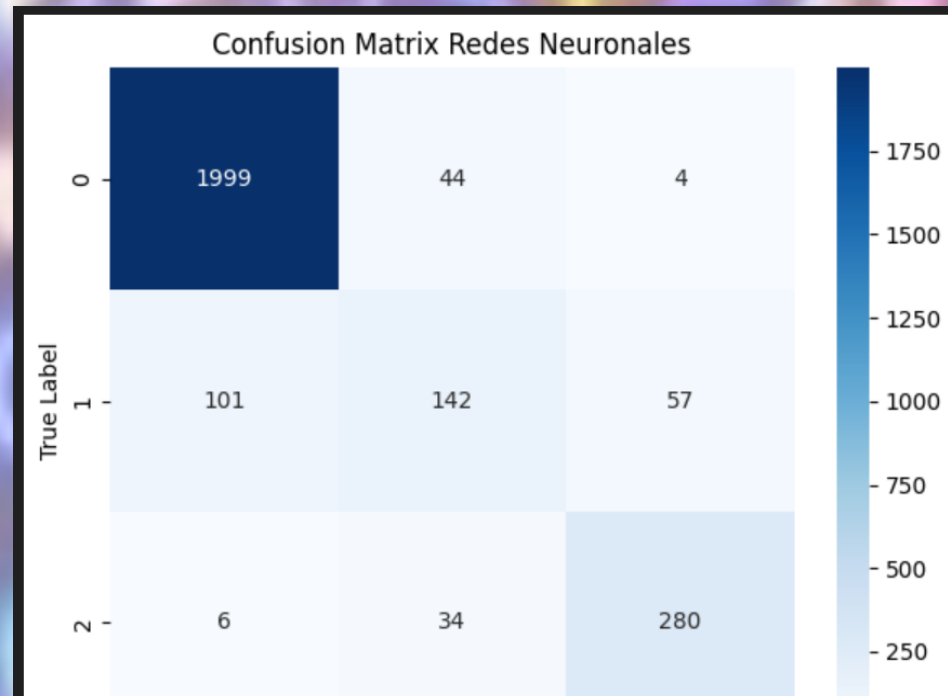




## 5- EVALUACIÓN DE LOS MODELOS

REDES NEURONALES\_ KERAS

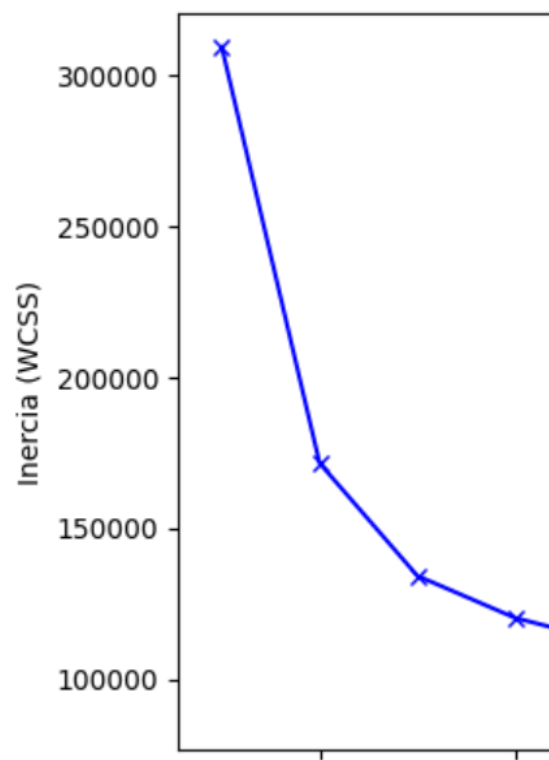
	precision	recall	f1-score	support
0	0.95	0.98	0.96	2047
1	0.65	0.47	0.55	300
2	0.82	0.88	0.85	320
accuracy			0.91	2667
macro avg	0.81	0.77	0.79	2667
weighted avg	0.90	0.91	0.90	2667



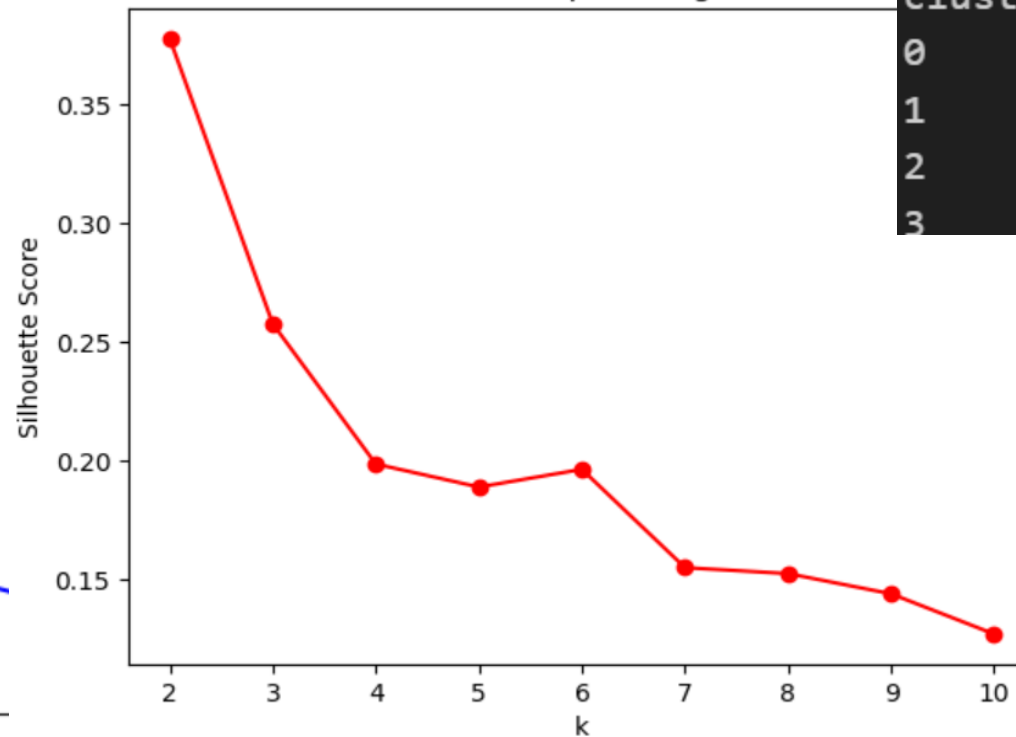
# 5- EVALUACIÓN DE LOS MODELOS

NO SUPERVISADO\_K-MEANS

Elbow Method para determinar k



Silhouette para elegir k



label2	0	1	2
cluster			
0	7071	0	0
1	379	871	1578
2	6009	0	0
3	4072	629	22

label2	0	1	2
cluster			
0	1.000000	0.000000	0.000000
1	0.134017	0.307992	0.557992
2	1.000000	0.000000	0.000000
3	0.862164	0.133178	0.004658

## *6-MODELO FINAL SELECCIONADO*

**Random Forest** ya que da los mejores resultados para los valores No OK de la Target Label 2. Lo mas importante de esta predicción no es que detecte lo posible fallos lo mejor posible, esto es, los 1 y 2.





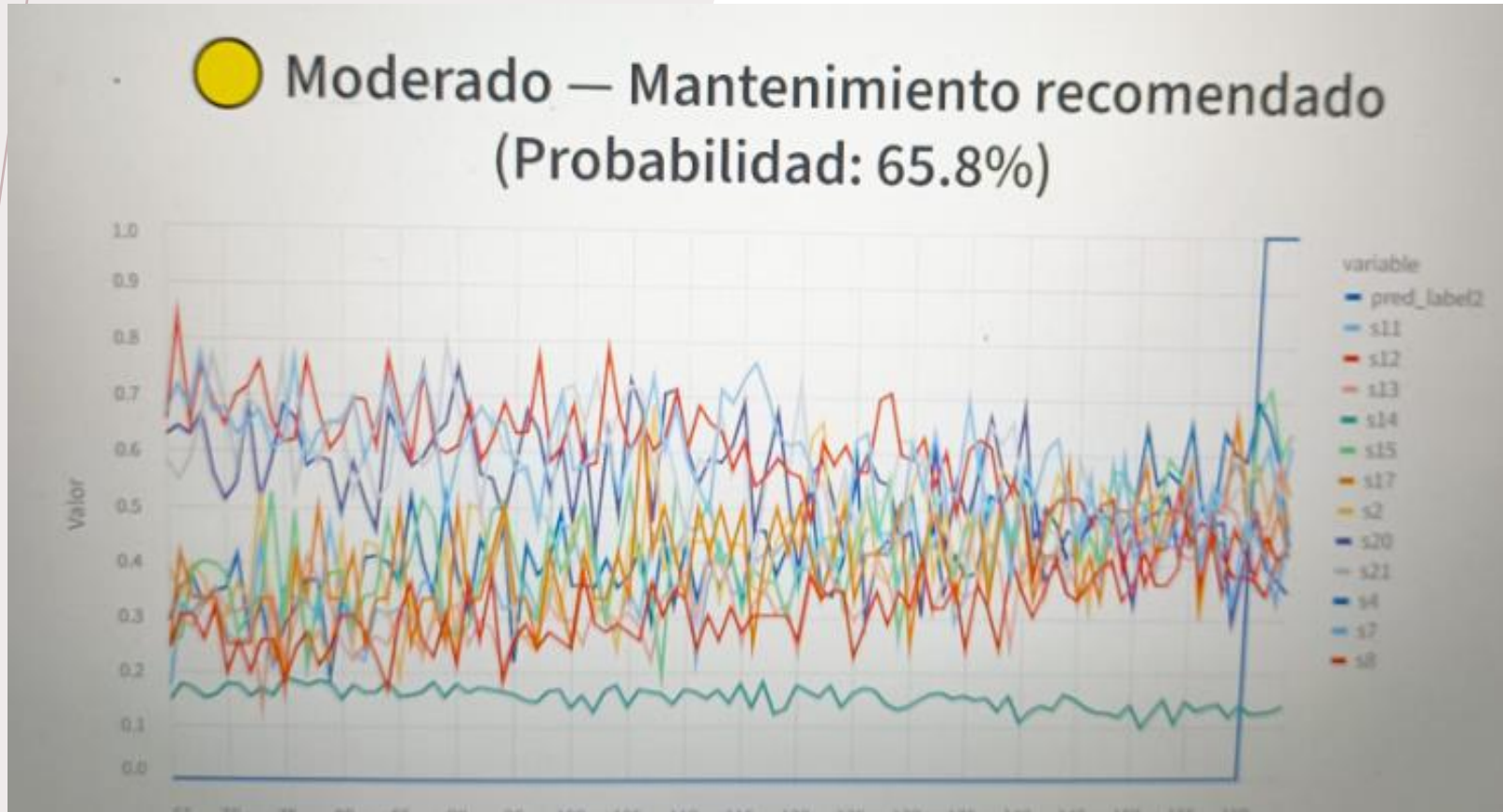
# 7-PRUEBAS FINALES



*MANTENIMIENTO  
PREDICTIVO: MOTORES  
DE AVIÓN*

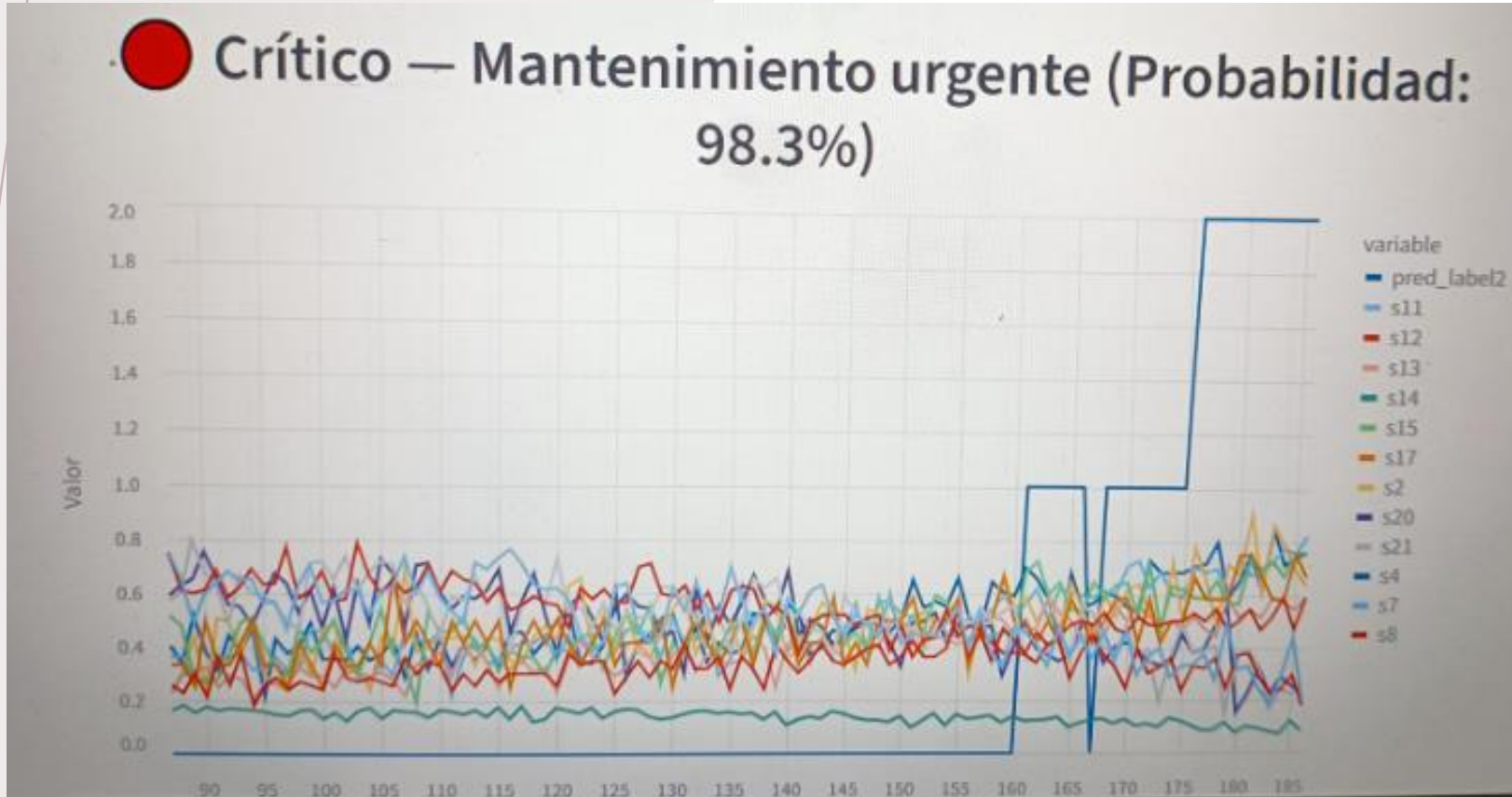


# 7-PRUEBAS FINALES



MANTENIMIENTO  
PREDICTIVO: MOTORES  
DE AVIÓN

## 7- PRUEBAS FINALES



*MANTENIMIENTO  
PREDICTIVO: MOTORES  
DE AVIÓN*





# 8- CONCLUSIONES

## Resultados y Fortalezas

- **Objetivo:** Predecir estado del motor con label2 (0 = OK, 1 = fallo en  $\leq 30$  ciclos, 2 = fallo en  $\leq 15$  ciclos).
- **Modelos probados:** Regresión logística, SVM, red neuronal, Pipeline y Random Forest.
- **Elección final: Random Forest, por:**
  - Alta efectividad en **detección de fallos** (clases 1 y 2).
  - Capacidad de gestionar datos desbalanceados
  - Interpretabilidad: ofrece **importancia de variables**, clave para entender qué sensores influyen en fallos.
- **Limitación conocida:**
  - Pocos datos reales de fallos → resultados en clases minoritarias limitados.
  - Aun así, detecta casos 1 y 2 mejor que otros modelos supervisados.





## 9- *PROXIMOS PASOS/MEJORAS*

- Recolectar más datos reales de fallos (clases 1 y 2).
  - Probar otros modelos
  - Evaluar técnicas balanceadas
- 
- **Conclusión:**
  - Proyecto consistente, bien estructurado y exitoso en detección de fallos dentro de las limitaciones.
  - Random Forest es una base sólida, ampliable y explicable.
  - Camino claro hacia una mejora continua con más datos y técnicas adaptadas al desequilibrio.







# *GRACIAS POR LA ATENCIÓN*

*MANTENIMIENTO PREDICTIVO  
MOTORES DE AVIÓN:*

*MODELO MACHINE LEARNING*

*PRESENTACIÓN TÉCNICA*

Leire Alonso Polanco

7 de Julio 2025