

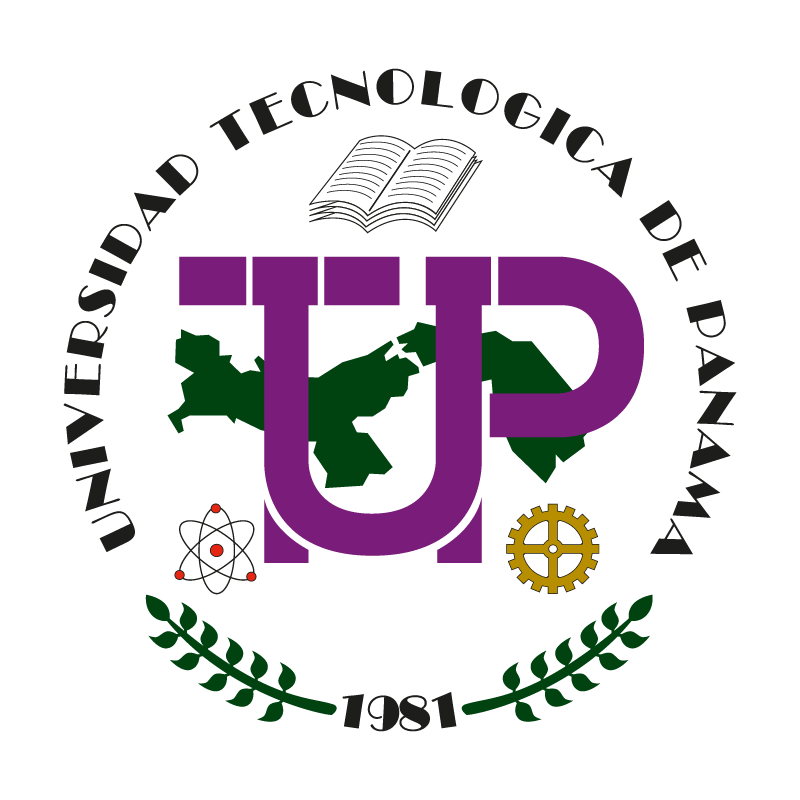
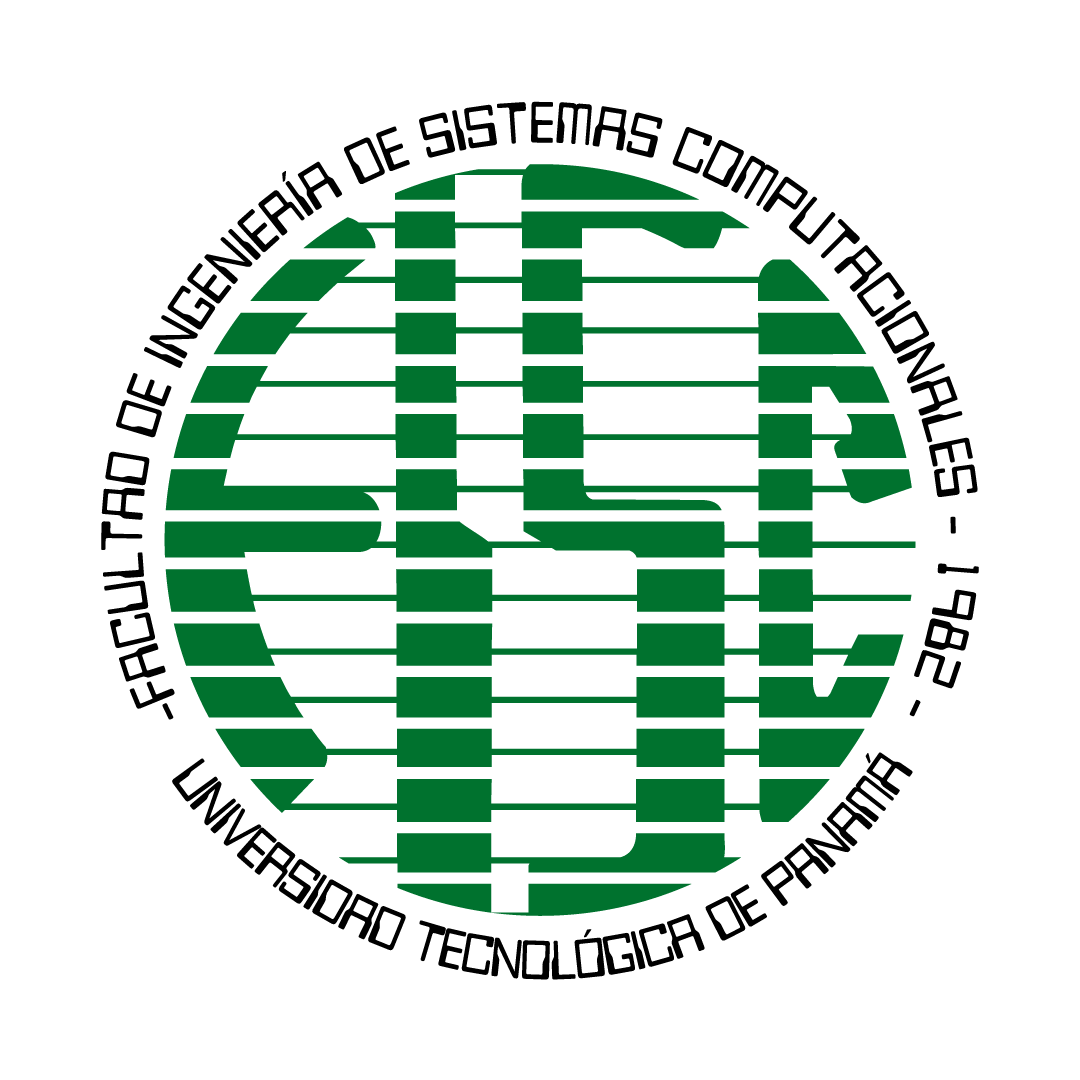
2025

**Detección de anomalías y predicción de fallos en el Sistema de producción de aire (APU)**

Angel Reyes

8-930-845

Universidad Tecnológica de Panamá



Universidad Tecnológica de Panamá

Facultad de Ingeniería en Sistemas y Computación

Maestría en analítica de datos

Asignatura

Modelos predictivos

Proyecto Final

Detección de anomalías y predicción de fallos en el Sistema de producción de aire (APU)

Estudiante

Angel Reyes

Cedula

8-930-845

Profesor

PhD Juan Marcos Castillo

Año

2025

Tabla de contenidos

[Introducción 4](#_Toc204180816)

[Justificación 5](#_Toc204180817)

[Antecedentes 6](#_Toc204180818)

[Definición del problema 7](#_Toc204180819)

[Análisis Predictivo 7](#_Toc204180820)

[I. Determinación de la base de datos 7](#_Toc204180821)

[II. Pre-procesamiento y limpieza 10](#_Toc204180822)

[III. Análisis descriptivo 12](#_Toc204180823)

[IV. Selección de variables 15](#_Toc204180824)

[V. Selección de modelos 15](#_Toc204180825)

[Conclusiones 24](#_Toc204180826)

[Recomendaciones y futuros estudios 24](#_Toc204180827)

[Bibliografía 25](#_Toc204180828)

# Introducción

Las fallas en trenes durante su operación generan múltiples problemas, especialmente cuando provocan la interrupción del servicio. Esto afecta tanto a la empresa operadora como a los usuarios, quienes pierden confianza en el sistema. Por lo que detectar estas fallas a tiempo puede evitar desalojos, detenciones y retiradas de servicio, lo que representa un gran valor.

En los trenes del metro, la Unidad de Producción de Aire (APU) alimenta sistemas clave como la suspensión y el sistema de frenos. La APU trabaja continuamente y, al no tener redundancia, cualquier falla implica retirar el tren de circulación. Además, estas fallas no suelen detectarse con los métodos tradicionales de mantenimiento basado en condiciones.

El mantenimiento predictivo busca cambiar el enfoque reactivo por uno preventivo, reduciendo detenciones imprevistas y tiempos de inactividad.

Por lo que el análisis se enfocará en aplicar modelos para la detección y predicción de fallos en base a las señales capturadas por los sensores instalados en los trenes, con el fin de contribuir a la mejora del mantenimiento ferroviario desde un enfoque proactivo y basado en datos.

# Justificación

La detección temprana de fallos en el sistema ferroviario es importante para garantizar la seguridad operativa, reducir costos y mejorar la experiencia del usuario. Las fallas en unidades como la APU pueden interrumpir el servicio comercial y obligar al retiro del tren de operación, como ocurrió ha ocurrido en múltiples ocasiones.

Este proyecto busca aplicar técnicas aprendidas en clase para identificar patrones anómalos en los datos y predecir fallos antes de que se vuelvan críticos. La base de datos nos ofrece trabajar con datos reales, sin valores perdidos y con estándares técnicos validados, lo que permite explorar soluciones predictivas con impacto potencial en la planificación de mantenimiento, la eficiencia operativa y la seguridad ferroviaria.

Asi mismo, esta investigación puede servir como base para la creación de sistemas inteligentes de monitoreo, que alerten automáticamente al personal de mantenimiento sobre posibles anomalías. Adicionalmente, este proyecto me ayuda a entender mejor el funcionamiento del APU y como realizar el análisis predictivo de los fallos, ya que trabajo en el sector ferroviario.

# Antecedentes

Durante los últimos años, el mantenimiento predictivo ha aparecido como una estrategia para mejorar la confiabilidad y disponibilidad de los sistemas ferroviarios, en especial cuando los fallos no planificados pueden comprometer la seguridad y la continuidad del servicio (Yahiaoui & Mohammedi, 2025). A diferencia de los enfoques correctivos o preventivos, el mantenimiento predictivo se basa en el monitoreo continuo y el análisis inteligente de datos operativos, lo cual permite anticipar fallas antes de que se manifiesten y optimizar el uso de recursos.

La digitalización de infraestructuras ferroviarias ha habilitado el uso de sensores analógicos y digitales en componentes críticos, como las unidades de producción de aire (APU), responsables del funcionamiento de frenos, puertas automáticas y sistemas neumáticos. Esta creciente complejidad ha hecho necesarios métodos más sofisticados que superen las limitaciones de los modelos estadísticos tradicionales. En respuesta, el aprendizaje profundo ha demostrado ser especialmente efectivo al detectar patrones no lineales y variaciones sutiles en las señales de sensores.

Estudios recientes, como el de Davari et al. (2021), han mostrado cómo los autoencoders disperses (SAE) pueden identificar comportamientos anómalos en las APU de trenes urbanos mediante el análisis de datos en tiempo real, alcanzando mejoras sustanciales en precisión, sensibilidad y reducción de falsas alarmas frente a modelos como los autoencoders variacionales (VAE).

# Definición del problema

Como mencionado previamente el mantenimiento predictivo se ha consolidado como una estrategia eficaz para anticipar fallos y reducir tiempos de inactividad. Sin embargo, la implementación práctica de PdM en entornos reales presenta varios desafíos, particularmente en la capacidad de identificar patrones anómalos a partir de grandes volúmenes de datos generados por sensores.

Aunque las técnicas de machine learning, como regresión logística, random forest o k-NN, han demostrado ser útiles en tareas de clasificación de estados operativos, aún enfrentan limitaciones cuando se aplican a entornos industriales con datos ruidosos, variables no lineales y clases desbalanceadas. Además, la selección de variables relevantes, la validación cruzada adecuada y el manejo del sobreajuste siguen siendo retos comunes.

Por lo tanto, el problema central que aborda este proyecto es el siguiente:

*¿Cómo aplicar técnicas de clasificación en machine learning para detectar de manera anticipada fallas en unidades de producción de aire (APU) en sistemas ferroviarios, mejorando la precisión y reduciendo los falsos positivos en el mantenimiento predictivo?*

En otras palabras, el objetivo es desarrollar un modelo de clasificación utilizando técnicas de *machine learning* que sea capaz de predecir fallas en unidades de producción de aire con la mayor precisión posible, pero sin hacer el sistema demasiado complejo. La meta es encontrar un equilibrio: lograr buenos resultados en métricas como RMSE, MAE y MAPE, pero con un modelo lo suficientemente simple como para que pueda implementarse fácilmente en la práctica, dentro de un entorno ferroviario real.

# Análisis Predictivo

## **Determinación de la base de datos**

Este conjunto de datos fue elegido de UC Irvine machne learning repository debido a su aplicabilidad en contextos reales de mantenimiento ferroviario, ya que permite entrenar modelos que anticipan fallos y optimizan la operación de sistemas críticos como los compresores ferroviarios.

Consiste en series de tiempo obtenidas a partir de diversos sensores analógicos y digitales instalados en el compresor de un tren. Los datos abarcan el período comprendido entre febrero y agosto de 2020 e incluyen 15 variables: presión, corriente del motor, temperatura del aceite y señales eléctricas de las válvulas de admisión de aire, etc. Este data set mantiene 1,516,948 valores.

Estas variables son:

1. **Presión del compresor (TP2, bar):**medida de la presión en el compresor.
2. **Presión del panel neumático (TP3, bar):** medida de la presión generada en el panel neumático.
3. **Presión tras el filtro ciclónico (H1, bar):** medida de la presión generada debido a la caída de presión cuando ocurre la descarga del filtro separador ciclónico.
4. **Presión en válvulas de descarga (DV pressure, bar):** medida de la caída de presión generada cuando las torres descargan los secadores de aire; una lectura en cero indica que el compresor está funcionando bajo carga.
5. **Presión en los reservorios (Reservoirs, bar):**medida de la presión aguas abajo de los reservorios, la cual debería estar cerca de la presión del panel neumático (TP3).
6. **Corriente del motor (Motor Current, A):** medida de la corriente en una fase del motor trifásico; presenta valores cercanos a 0A cuando está apagado, 4A cuando funciona sin carga, 7A cuando opera bajo carga y 9A al arrancar.
7. **Temperatura del aceite (Oil Temperature, ºC):** medida de la temperatura del aceite en el compresor.
8. **Válvula de admisión – señal eléctrica (COMP):** señal eléctrica de la válvula de admisión de aire del compresor; está activa cuando no hay entrada de aire, lo que indica que el compresor está apagado o funcionando sin carga.
9. **Válvula de salida – señal eléctrica (DV electric):** señal eléctrica que controla la válvula de salida del compresor; está activa cuando el compresor funciona bajo carga, e inactiva cuando está apagado o sin carga.
10. **Control de torres de secado (TOWERS):** señal eléctrica que define qué torre se encarga del secado del aire y cuál elimina la humedad extraída; cuando está inactiva, indica que la torre uno está funcionando; cuando está activa, indica que está operando la torre dos.
11. **Activación bajo carga (MPG):** señal eléctrica responsable de iniciar el compresor bajo carga al activar la válvula de admisión cuando la presión en la unidad de producción de aire (APU) cae por debajo de 8.2 bar; activa el sensor COMP, que se comporta igual que MPG.
12. **Baja presión (LPS):** señal eléctrica que se activa cuando la presión cae por debajo de 7 bares.
13. **Interruptor de presión (Pressure Switch):** señal eléctrica que detecta la descarga en las torres de secado de aire.
14. **Nivel de aceite bajo (Oil Level):** señal eléctrica que detecta el nivel de aceite en el compresor; está activa cuando el nivel está por debajo del valor esperado.
15. **Impulso de caudal (Caudal Impulse):** señal eléctrica que cuenta los pulsos generados por la cantidad absoluta de aire que fluye desde la APU hacia los reservorios.

El conjunto de datos no está etiquetado, pero los reportes de fallos proporcionados por el dataset están disponibles en la siguiente tabla. Esto permite evaluar la efectividad de los modelos de detección de anomalías y predicción de fallos.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nr. | Start Time | End Time | Failure | Severity | Report |
| #1 | 04/18/20 0:00 | 04/18/20 23:59 | Air leak | High stress |  |
| #2 | 5/29/202023:30 | 05/30/20 6:00 | Air Leak | High stress | Maintenance on 30 Apr at 12:00 |
| #3 | 06/05/20 10:00 | 06/07/20 14:30 | Air Leak | High stress | Maintenance on 8 Jun at 16:00 |
| #4 | 07/15/20 14:30 | 07/15/20 19:00 | Air Leak | High stress | Maintenance on 16 Jul at 00:00 |

## **Pre-procesamiento y limpieza**

1. Primero se cargó el dataset y se observaron las columnas y las primeras filas, para entender mejor los datos:

A computer code with green text

AI-generated content may be incorrect.

1. Se incluye la columna Failure y se colocan todos los valores como 0

A white rectangular object with a black border

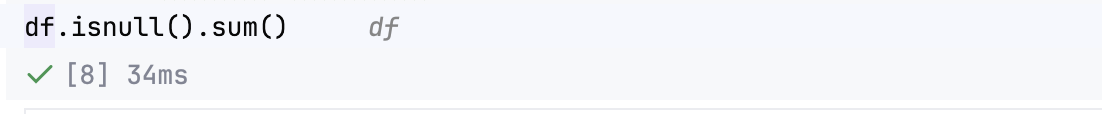
AI-generated content may be incorrect.

1. El Problema nos menciona unas fechas y unos tiempos donde hubo fallos en la realidad, por lo que se define el dataframe y los intervalos de tiempo, Luego se colocan las condiciones para marcarlo como un fallo (1).

A screenshot of a computer code

AI-generated content may be incorrect.

1. Se verifican los valores nulos. Aunque no fue encontrado ningún valor nulo.



## **Análisis descriptivo**

**Análisis estadístico**

Se realizo un análisis estadístico de las variables del dataset. Donde observamos los siguientes puntos relevantes:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | count | mean | min | 25% | 50% | 75% | max | std |
|  | 1516948 | 7584735 | 0 | 3792368 | 7584735 | 11377100 | 15169470 | 4379053 |
| timestamp | 1516948 | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| TP2 | 1516948 | 1.37 | -0.03 | -0.01 | -0.01 | -0.01 | 10.68 | 3.25 |
| TP3 | 1516948 | 8.98 | 0.73 | 8.49 | 8.96 | 9.49 | 10.30 | 0.64 |
| H1 | 1516948 | 7.57 | -0.04 | 8.25 | 8.78 | 9.37 | 10.29 | 3.33 |
| DV\_pressure | 1516948 | 0.06 | -0.03 | -0.02 | -0.02 | -0.02 | 9.84 | 0.38 |
| Reservoirs | 1516948 | 8.99 | 0.71 | 8.49 | 8.96 | 9.49 | 10.30 | 0.64 |
| Oil\_temperature | 1516948 | 62.64 | 15.40 | 57.78 | 62.70 | 67.25 | 89.05 | 6.52 |
| Motor\_current | 1516948 | 2.05 | 0.02 | 0.04 | 0.05 | 3.81 | 9.30 | 2.30 |
| COMP | 1516948 | 0.84 | 0.00 | 1.00 | 1.00 | 1.00 | 1.00 | 0.37 |
| DV\_eletric | 1516948 | 0.16 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.37 |
| Towers | 1516948 | 0.92 | 0.00 | 1.00 | 1.00 | 1.00 | 1.00 | 0.27 |
| MPG | 1516948 | 0.83 | 0.00 | 1.00 | 1.00 | 1.00 | 1.00 | 0.37 |
| LPS | 1516948 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.06 |
| Pressure\_switch | 1516948 | 0.99 | 0.00 | 1.00 | 1.00 | 1.00 | 1.00 | 0.09 |
| Oil\_level | 1516948 | 0.90 | 0.00 | 1.00 | 1.00 | 1.00 | 1.00 | 0.29 |
| Caudal\_impulses | 1516948 | 0.94 | 0.00 | 1.00 | 1.00 | 1.00 | 1.00 | 0.24 |
| failure | 1516948 | 0.02 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.14 |

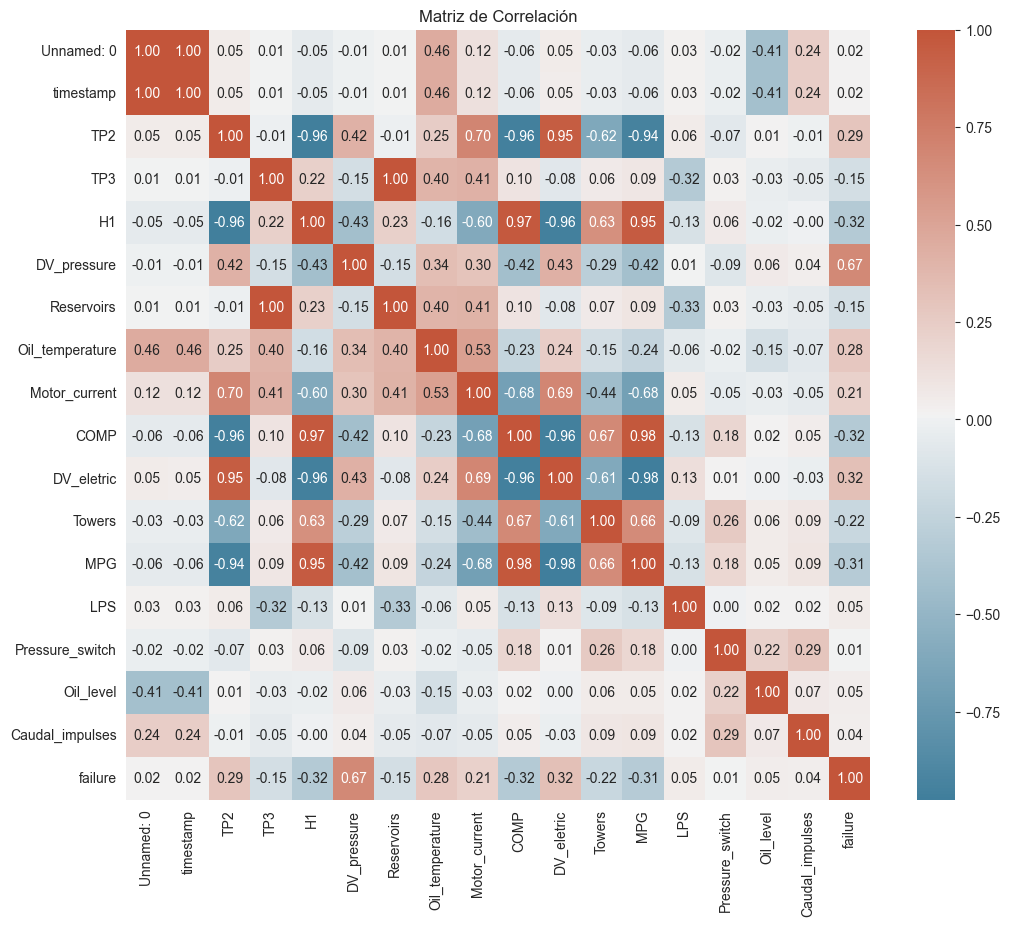
**TP2:** El rango va de -0.03 a 10.68, con una media de 1.37 y una desviación de 3.25.

**Dv Pressure:** El rango va de -0.03 a 9,84, con una media de 0,06 y una desviacion estándar de 0.38. Por lo que podemos ver que la media es baja y los puntos están bastante concentrados, por lo que un valor de 9.84 parece un punto extremo.

Tambien observamos que las variables: **COMP, DV\_eletric, Towers, MPG, LPS, Pressure\_switch, Oil\_level, Caudal\_impulses**; son variables binarias que representan estados (valores de 1 o 0).

**Matriz de correlación**

Se realizo la matriz de correlación de las variables, obteniendo el siguiente grafico:

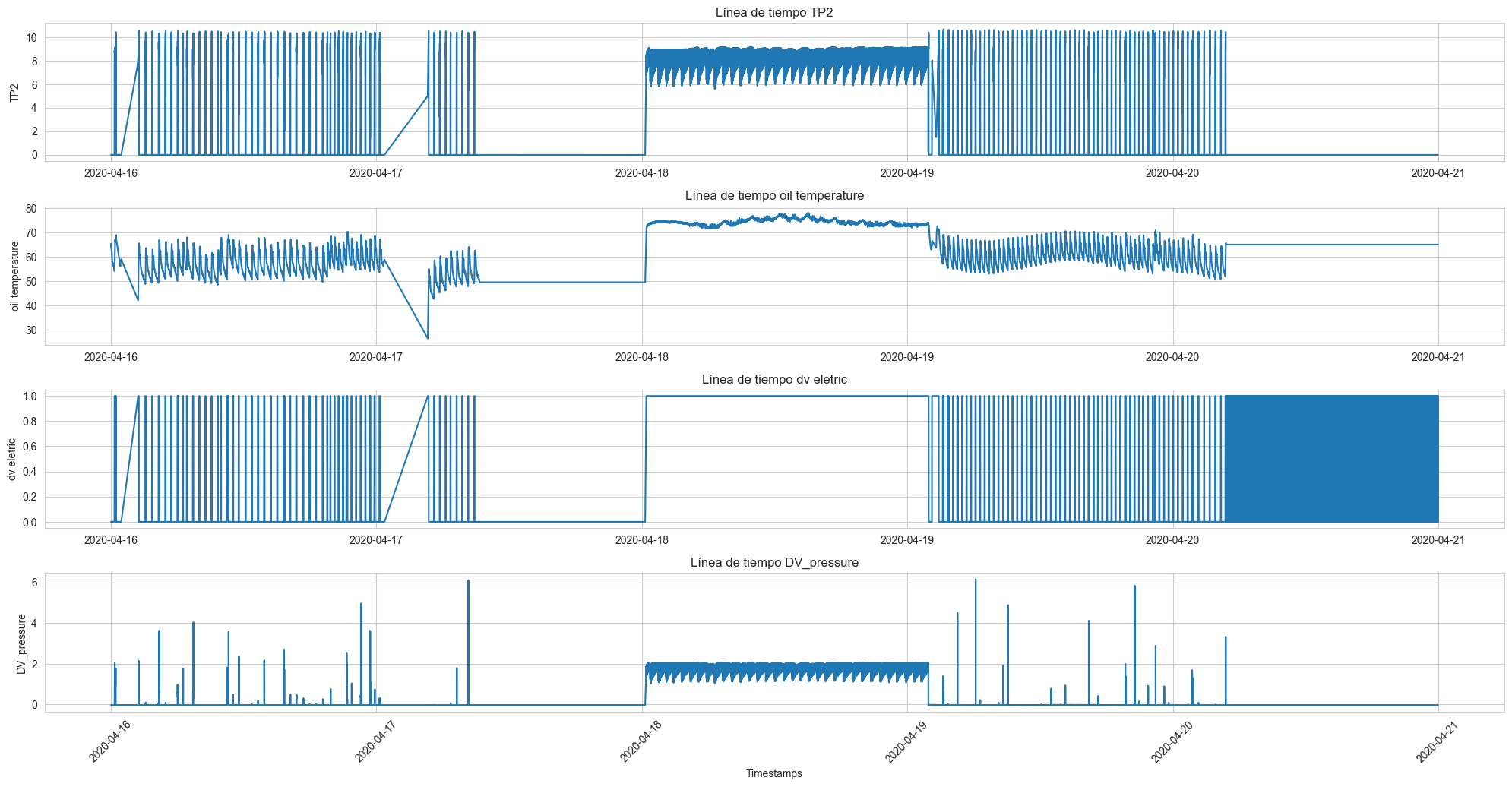


Podemos observar en el grafico una correlación negativa entre TP2 y DV\_eletric (-0.96), así como entre TP2 y COMP (-0.96), lo que sugiere que cuando TP2 aumenta, estos componentes probablemente estan apagados.

En cuanto a la variable objetivo (Failure) muestra una correlación moderada con DV\_pressure (0.67), lo que signigica que una presión diferencial elevada podría asociarse a fallas. Adicionalmente muestra la Correlación más débil pero positiva con TP2 (0.29), Oil\_temperature (0.28) y Dv\_electric (0.32).

**Exploración de la data**

Exploramos la data de las variables con mayor correlación tratamos de observar las tendencias durante el funcionamiento normal en comparación con las tendencias cuando el sistema ha fallado. Notamos que la diferencia en las tendencias es muy marcada, lo que indica que podemos aprovechar esta diferencia para predecir fallos.



En la imagen anterior, se a intentado mostrar cómo se ve la secuencia de señales en un escenario de normalidad a fallo y regreso a la normalidad. El contorno rojo indica el período de tiempo en el que ocurre la falla. Esto demuestra que, incluso antes de que se produzca el fallo, ya hay anomalías en los datos que, si se detectan, pueden ser útiles para predecir el fallo con anticipación.

## **Selección de variables**

La selección adecuada de variables es crucial para desarrollar un modelo predictivo efectivo y evitar errores. Por lo que hemos dividido el dataset en variables de X (predictoras) que representan todo lo que el sistema mide y que potencialmente podría ayudar a anticipar una falla.

Las variables utilizadas en X son: TP2, TP3, H1, DV\_pressure, Reservoirs, Oil\_temperature, Motor\_current, COMP, DV\_eletric, Towers, MPG, LPS, Pressure\_switch, Oil\_level y Caudal\_impulses.

Por otro lado, hemos elegido la variable Failure para Y que representa el objetivo que queremos predecir. En este caso, es un indicador binario que señala si hubo una falla (1) o no hubo falla (0) en el sistema del compresor.

Estas variables se han dividido en data de entrenamiento y de prueba con un division de 80% de entrenamiento y 20% de prueba.



## **Selección de modelos**

Para abordar el problema de predicción de fallos se adopto una técnica de clasificación. Dado que los datos son una serie temporal con registros que llegan cada 10 segundos, tenemos seis registros por minuto, cada uno etiquetado como éxito o falla. Se experimentó con varios modelos, incluyendo Regresion logistica, Random Forest y K-Nearest Neighbors.

Consideración de los modelos:

* **Regresión Logística:** Un modelo simple y fácilmente interpretable, adecuado como línea base, pero limitado para capturar relaciones no lineales entre variables.
* **Random Forest:** Un algoritmo de conjunto que combina múltiples árboles de decisión entrenados con diferentes subconjuntos de datos, ofreciendo robustez, buen manejo del ruido y alta precisión.
* **K-Nearest Neighbors (KNN):** Un modelo basado en la similitud entre observaciones, sencillo de implementar, pero sensible al ruido, a la escala de los datos y con alto costo computacional en la predicción.

**Regresión Logística**

Iniciamos evaluando el modelo de regresión logística, lo entrenamos con un número máximo de iteraciones de 1000, con los datos de entrenamiento previamente definidos

A screenshot of a computer code

AI-generated content may be incorrect.

Tras correr el modelo observamos los siguientes resultados:

* Reporte de clasificacion

Reporte de clasificación:

precision recall f1-score support

0 1.00 0.99 1.00 297344

1 0.75 0.81 0.78 6046

accuracy 0.99 303390

macro avg 0.87 0.90 0.89 303390

weighted avg 0.99 0.99 0.99 303390

ROC AUC: 0.9944

RMSE: 0.0959

MAE: 0.0092

MAPE: 24077387506493.1094

MAPE seguro: 0.1930

* Matriz de confusión Regresión logarítmica

A graph with numbers and a number

AI-generated content may be incorrect.

* Gráfico Regresión logarítmica – Fallos reales Vs Predichos

A graph of a graph with red dots

AI-generated content may be incorrect.

* Análisis de los resultados Regresión Logística
* El accuracy en general el modelo acierta en el 99% de los casos, aunque este valor está influenciado por la gran cantidad de ejemplos de la clase 0, por lo que no refleja del todo el rendimiento sobre la clase 1.

Por ejemplo, si analizamos el desempeño en la clase 1 (fallos) vemos que tiene una precisión del 75% y un recall del 81%, lo que significa que identifica correctamente la mayoría de los casos positivos, pero es significativamente menor la precision y el recall comparado con el de la clase 0(sin fallos).

* El ROC AUC es de 0.9944, lo que es muy alto, indica que el modelo tiene una capacidad excelente para distinguir entre las clases 0 y 1, aunque puede estar influenciado por el desbalance.
* Errores RMSE (0.0959) y MAE (0.0092) son muy bajos, aunque estas métricas son normales de regresión, muestran que el modelo rara vez se equivoca al predecir, nuevamente menciono que puede estar influenciado por el desbalance.
* El MAPE da un valor alto debido a que es binario y al ser dividido por 0 es invalido, por lo que se generó un MAPE seguro que excluye los valores en cero y se obtiene un error porcentual de 19.3% lo que es razonable sobre los fallos.
* En la matriz de confusión se observa que el modelo clasifica correctamente 4,879 de los 6,046 positivos reales, con 1,167 falsos negativos y 1,622 falsos positivos, lo cual es aceptable considerando el desbalance.

**Random Forest Classifier**

El modelo siguiente evaluado el de Random Forest Classifier, lo entrenamos con un random state de 23, con los datos de entrenamiento previamente definidos.

A screenshot of a computer code

AI-generated content may be incorrect.

Tras correr el modelo observamos los siguientes resultados:

* Reporte de clasificacion

Reporte de clasificación:

precision recall f1-score support

0 1.00 1.00 1.00 297344

1 0.99 0.99 0.99 6046

accuracy 1.00 303390

macro avg 0.99 0.99 0.99 303390

weighted avg 1.00 1.00 1.00 303390

ROC AUC: 0.9999

RMSE: 0.0226

MAE: 0.0005

MAPE: 1276606242637.7358

MAPE seguro: 0.0114

* Matriz de confusión Random Forest Classifier

**A graph with numbers and a number in a row

AI-generated content may be incorrect.**

* Gráfico Random Forest Classifier – Fallos reales Vs Predichos

**A graph with red dots and blue lines

AI-generated content may be incorrect.**

* Análisis de los resultados Random Forest Classifier
* El Accuracy en general, el modelo acierta en el 100% de los casos, lo que refleja un rendimiento muy bueno. A diferencia del caso anterior, esta vez el desempeño es también muy alto en la clase 1 (fallos), con una precisión del 99% y un recall del 99%. Esto indica que el modelo identifica correctamente casi todos los casos de fallos, con apenas una diferencia mínima respecto a la clase 0 (sin fallos).
* El ROC AUC es de 0.9999, lo cual es prácticamente perfecto. Esto muestra que el modelo tiene una capacidad sobresaliente para distinguir entre ambas clases, y en este caso, el desbalance no parece afectar negativamente esa capacidad.
* Errores RMSE (0.0226) y MAE (0.0005) son extremadamente bajos. Lo que confirma que el modelo comete muy pocos errores incluso al tratar las clases como valores numéricos. Esto refuerza la idea de que el modelo tiene un rendimiento consistente.
* El MAPE nuevamente da un valor muy alto debido a la división por cero, lo cual lo vuelve inválido en problemas de clasificación binaria. Por ello, se utilizó un MAPE seguro, que excluye los ceros, arrojando un error porcentual de solo 1.14%, lo cual es muy bajo y muestra una gran precisión en los casos positivos.
* En la matriz de confusión se observa que el modelo clasifica correctamente 5,977 de los 6,046 positivos reales, con 69 falsos negativos y 86 falsos positivos, Esto indica que el modelo tiene una alta sensibilidad y especificidad, siendo capaz de detectar casi todos los fallos sin generar muchas falsas alarmas.

**K-Nearest Neighbors**

El ultimo modelo evaluado es el de K-Nearest Neighbors, lo entrenamos nuevamente con los datos de entrenamiento previamente definidos.

A screenshot of a computer program

AI-generated content may be incorrect.

Tras correr el modelo observamos los siguientes resultados:

* Reporte de clasificacion

Reporte de clasificación:

precision recall f1-score support

0 1.00 1.00 1.00 297344

1 0.97 0.97 0.97 6046

accuracy 1.00 303390

macro avg 0.98 0.98 0.98 303390

weighted avg 1.00 1.00 1.00 303390

ROC AUC: 0.9963

RMSE: 0.0355

MAE: 0.0013

MAPE: 2746187847534.6646

MAPE seguro: 0.0326

* Matriz de confusión K-Nearest Neighbors

**A graph with numbers and a number in blue squares

AI-generated content may be incorrect.**

* Gráfico K-Nearest Neighbors – Fallos reales Vs Predichos

**A graph of a graph

AI-generated content may be incorrect.**

* Análisis de los resultados K-Nearest Neighbors
* El accuracy en el modelo acierta en prácticamente el 100% de los casos igual que en el random forest. Aunque este valor está influenciado por la mayoría de ejemplos pertenecientes a la clase 0, el rendimiento sobre la clase 1 (fallos) es también muy alto, con una precisión del 97% y un recall del 97%, lo que significa que identifica correctamente la gran mayoría de los fallos.
* El ROC AUC es de 0.9963. El modelo tiene una gran capacidad de distinguir entre clases, aunque un poco menor que la del modelo Random Forest. Aún así, ofrece un rendimiento muy bueno.
* Errores RMSE (0.0355) y MAE (0.0013) son bajos, lo cual indica que el modelo comete muy pocos errores al clasificar. Por lo que respaldan la calidad de las predicciones.
* El MAPE nuevamente da un valor inválido debido a divisiones por cero en la clase 0. Por eso, se calculó un MAPE seguro, que excluye estos casos y arrojó un error porcentual de 3.26%.
* En la matriz de confusión se observa que el modelo clasifica correctamente 5,849 de los 6,046 positivos reales, con 197 falsos negativos y 185 falsos positivos. Siendo capaz de detectar la mayoría de los fallos, aunque con un poco más de errores que Random Forest.

# Conclusiones

Tras aplicar distintos modelos de clasificación al conjunto de datos del compresor APU, los resultados evidencian un desempeño bastante alto, especialmente en la detección de fallos, lo cual es crucial para tareas de mantenimiento predictivo.

Obteniendo los siguientes resultados:

**Resumen resultados**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Modelo | Accuracy | Precision (clase 1) | Recall (Clase 1) | ROC | RMSE | MAE | MAPE  Seguro |
| Regresion Logistica | 99% | 0.75 | 0.81 | 0.994 | 0.0959 | 0.0092 | 19.30% |
| Random Forest Classifier | 100% | 0.99 | 0.99 | 0.999 | 0.0226 | 0.0005 | 1.14% |
| K-Nearest Neighbors | 100% | 0.97 | 0.97 | 0.996 | 0.035 | 0.0013 | 3.26% |

Si bien tanto Random Forest como K-Nearest Neighbors (KNN) y Regresion logistica alcanzan una accuracy de casi del 100%

El modelo Random Forest se posiciona como la mejor opcion, mostrando un buen equilibrio entre sensibilidad y precisión para la clase minoritaria (Fallos). Sin incurrir en un alto número de falsos positivos, lo convierte en una herramienta altamente confiable. Adicionalmente, sus métricas de error, tanto en términos absolutos como relativos, son las más bajas entre los modelos evaluados, lo que sugiere una alta consistencia.

# Recomendaciones y futuros estudios

Observando las limitaciones del enfoque actual se plantean las siguientes recomendaciones para avanzar hacia un mejor sistema predictivo:

* Reformular el objetivo del modelado para anticipar fallos antes de que ocurran, ya que los modelos actuales clasifican condiciones posteriores al fallo, lo cual limita su utilidad operativa.
* Profundizar en la predicción del tiempo hasta el fallo, utilizando agregaciones temporales (como medias y desviaciones de los últimos minutos) y explorando modelos más sofisticados.
* Explorar autoencoders para detección temprana de anomalías, permitiendo identificar desviaciones sutiles del comportamiento normal sin depender de etiquetas explícitas de fallo, y activando alertas con mayor anticipación.
* Incluir variables contextuales y operativas (como ubicación, condiciones externas o carga del sistema) para enriquecer el modelo y mejorar su capacidad de generalización.

Estas líneas contribuirán a construir un sistema de mantenimiento predictivo más proactivo y robusto, alineado con los objetivos reales del proyecto.

# Bibliografía

1. **Yahiaoui, L., & Mohammedi, M.** (2025). Efficient and Reliable Predictive Maintenance in Trains based on BiLSTM Model. Wireless Communications and Mobile Computing.  
   <https://ieeexplore.ieee.org/document/11059667>
2. **Davari, N., Veloso, B., & Ribeiro, R. P.** (2021). Predictive maintenance based on anomaly detection using deep learning for air production unit in the railway industry. IEEE International Conference on Data Science and Advanced Analytics (DSAA).  
   <https://ieeexplore.ieee.org/document/9564181>

# Anexo

<https://github.com/arrch98/Proyecto-predicci-n-de-fallos/tree/00327462026d4fafadf3f0b6c9018a6cc882ffaf>