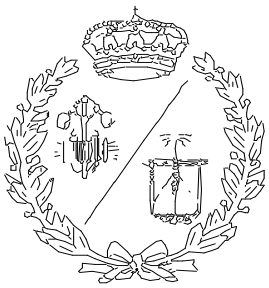
ESCUELA TÉCNICA SUPERIOR DE INGENIEROS

INDUSTRIALES Y DE TELECOMUNICACIÓN

UNIVERSIDAD DE CANTABRIA



Proyecto Fin de Máster

Modelo para el mantenimiento predictivo de segmentos especiales de vía

(Model for predictive maintenance of railway switches and crossings)

Para acceder al Título de

MÁSTER UNIVERSITARIO EN

INGENIERIA INDUSTRIAL

Autor: Pablo Infante Gómez

Julio – 2023

RESUMEN

En las últimas décadas, se ha producido un crecimiento exponencial en la generación, recopilación y almacenamiento de datos impulsado por el avance de la tecnología y la digitalización en diversos ámbitos. La invención de dispositivos móviles inteligentes, el avance de la instrumentación electrónica, el desarrollo de la tecnología IoT, todo ello sumado a la evolución de la capacidad de almacenamiento y de procesamiento computacional, han sido algunos factores destacados en el crecimiento vertiginoso de los datos.

La importancia de la evolución de los datos radica en su capacidad para proporcionar información y conocimientos que antes eran inimaginables. Los datos se han convertido en un activo invaluable para las organizaciones, permitiéndoles comprender mejor a sus clientes, identificar tendencias, detectar patrones, automatizar procesos y optimizar sus operaciones.

La necesidad de gestionar y obtener valor de esta generación masiva de datos ha impulsado el desarrollo de técnicas de análisis de datos avanzadas, como el aprendizaje automático (Machine Learning (ML)), cuya función principal es extraer información valiosa de los datos con el objetivo de generar nuevos conocimientos que permitan la innovación, ayudar en la toma de decisiones, mejorar la eficiencia y generar nuevos servicios y modelos de negocio.

A su vez, como consecuencia del aumento de la preocupación global por la emisión de gases contaminantes y el cambio climático, el sistema ferroviario ha adquirido en los últimos años mayor importancia debido a su eficiencia energética, bajas emisiones, menor impacto ambiental y uso eficiente del espacio. Este auge de la importancia del ferrocarril ha provocado que se adopten estrategias enfocadas a realizar una mayor inversión, destinada a la innovación y mejora de las infraestructuras ferroviarias.

Este proyecto pretende poner en valor y aprovechar la evolución de las técnicas de análisis de datos para tratar de aportar una mejora en el plan de mantenimiento de un sistema tan importante como el sistema ferroviario. Para ello, se desarrollarán técnicas de Machine Learning basadas en datos como el Análisis de Componentes Principales (PCA), capaz de detectar patrones estadísticos o tendencias entre los conjuntos de datos.

De esta manera, se pretende implementar un algoritmo que integre funciones de diagnóstico, detección y predicción de fallos con el fin de evolucionar el plan de mantenimiento de los aparatos de vía hacia una estrategia de mantenimiento predictiva que tenga un impacto notable en la seguridad y eficiencia del transporte ferroviario, así como en la reducción de los costes de operación.

ABSTRACT

In recent decades, there has been an exponential growth in data generation, collection and storage driven by the advancement of technology and digitalization in various fields. The invention of smart mobile devices, the advancement of electronic instrumentation, the development of IoT technology, all this combined with the evolution of storage and computational processing capacity, have been some prominent factors in the dizzying growth of data.

The importance of the evolution of data is based on in its ability to provide information and insights that were previously unimaginable. Data has become an invaluable asset for organizations, enabling them to a better comprehension of their customers, identifying trends, detecting patterns, automating processes and optimizing their operations.

The need to manage and obtain value from this massive generation of data has driven the development of advanced data analysis techniques, such as Machine Learning (ML), whose main function is to extract valuable information from data in order to generate new knowledge that will enable innovation, decision-making assistance, improve efficiency and generate new services and business models.

In turn, as a result of the increasing global concern about the emission of polluting gases and climate change, the railway system has become more important in recent years due to its energy efficiency, low emissions, lower environmental impact and efficient use of space. This rise in the importance of railroads has led to the adoption of strategies focused on greater investment, aimed at innovation and improvement of railway infrastructures.

This project aims to value and take advantage of the evolution of data analysis techniques to try to bring an improvement in the maintenance plan of a system as important as the railway system. For this purpose, Machine Learning techniques based on data such as Principal Component Analysis (PCA) will be developed, capable of detecting statistical patterns or trends among data sets.

In this way, it is intended to implement an algorithm that integrates diagnostic functions, detection and prediction of failures in order to evolve the maintenance plan of track devices towards a predictive maintenance strategy that has a significant impact on the safety and efficiency of rail transport, as well as on the reduction of operating costs.

AGRADECIMIENTOS

Me gustaría dedicar unas palabras de agradecimiento a todas aquellas personas que, directa o indirectamente, han participado en este proyecto.

Quiero dar las gracias particularmente a Pablo Pascual Muñoz y David García Sánchez, codirectores de este proyecto, por confiar en mí para realizar un trabajo de fin de máster innovador como el presente, por su enseñanza, por su amabilidad y predisposición a ayudarme en todo momento y cómo no, por su buen humor que me ha hecho más ameno el trabajo desarrollado.

Por último, y más importante, quiero agradecer a mi familia y amigos su paciencia y todo su apoyo durante la elaboración de este proyecto. Con mención especial a María, por ser mi mayor apoyo haciéndolo todo más fácil.

GLOSARIO

Aguja: una aguja ferroviaria es un dispositivo utilizado en las vías férreas para cambiar la dirección de los trenes o para permitir el acceso a diferentes vías. Consiste en una estructura metálica que consta de un corazón o punto de unión y dos ramas o lenguas. Estas ramas pueden moverse mediante un mecanismo para conectarse con diferentes vías, lo que permite a los trenes cambiar de una vía a otra.

Array: término utilizado en el campo del análisis de datos para denominar a estructuras de datos de un mismo tipo en formato tabla.

Contraaguja: la contraaguja ferroviaria, también conocida como aguja de protección o contra-cambio, es un tipo especial de aguja que se utiliza para evitar que los trenes cambien de vía de manera involuntaria. Se coloca en una posición bloqueada, impidiendo el paso de los trenes a una vía específica.

Dataset: conjunto de datos ordenados según distintos atributos y generalmente estructurados en formato tabla, matriz o archivo. Se trata de una herramienta muy utilizada en el campo de la inteligencia artificial y la ciencia de datos, con el objetivo de entrenar modelos, elaborar análisis o realizar simulaciones.

Data Frame: estructura de datos bidimensional ordenada por filas y columnas, similar a una matriz, pero con la diferencia de que cada columna puede ser de un tipo de dato distinto, característica que la hace muy versátil e idónea para aplicaciones de análisis de datos.

Entorno de Desarrollo Integrado (IDE): es un conjunto de herramientas que simplifican enormemente la programación y el desarrollo del software. Incluye un editor de texto, un editor de proyectos que contiene módulos con distintas funcionalidades y un compilador para la depuración de errores.

Garganta de guía: parte más estrecha de la vía en la zona del cruzamiento.

Gemelo digital: es un modelo virtual que refleja con exactitud un objeto físico, proceso o sistema. Se utilizan para realizar pruebas, simulaciones y estudiar el comportamiento de un producto digital, con el objetivo de utilizar ese conocimiento obtenido en optimizar el producto real.

IoT: se refiere a la red de objetos físicos que están equipados con sensores, software y otras tecnologías que les permiten conectarse e intercambiar datos a través de Internet. Estos objetos pueden incluir dispositivos cotidianos como teléfonos inteligentes, dispositivos portátiles, electrodomésticos, vehículos y maquinaria industrial. El concepto detrás del IoT es permitir que estos objetos recopilen y compartan datos, se comuniquen entre sí y realicen diversas tareas de forma autónoma o con mínima intervención humana.

Junta de Contraaguja (JCA): junta situada en la contraaguja a la altura del extremo del talón de aguja.

Outlier: punto o un conjunto de puntos de datos que se alejan significativamente del resto de los datos en un conjunto. Estos valores son inusuales o atípicos en comparación con el patrón general del conjunto de datos.

Personally Identifiable Information (PII): se define como datos que podrían utilizarse por sí solos para identificar de manera directa, contactar o localizar con precisión a una persona. Por ejemplo: nombres de usuarios, números de teléfono, correos electrónicos.

Random Forests: es un algoritmo de aprendizaje automático que se basa en la combinación de múltiples árboles de decisión para realizar tareas de clasificación y regresión. La idea principal es crear un conjunto de árboles de decisión independientes y combinar sus predicciones para obtener un resultado final. Cada árbol de decisión se construye a partir de una muestra aleatoria del conjunto de datos original y utiliza un subconjunto aleatorio de características o variables. Es una técnica poderosa y versátil que se utiliza en diversos campos, como la ciencia de datos, la minería de datos y la inteligencia artificial.

Red neuronal: es un modelo de aprendizaje automático inspirado en el funcionamiento del cerebro humano. Está compuesto por un conjunto interconectado de nodos llamados neuronas artificiales, que trabajan en conjunto para procesar y analizar datos. Cada neurona en una red neuronal tiene una función de activación que determina su salida en función de las entradas y los pesos asociados a esas conexiones. Los pesos representan la importancia relativa de las conexiones entre las neuronas y se ajustan durante el proceso de entrenamiento de la red.

Redes convolucionales: también conocidas como Convolutional Neural Networks (CNN), son un tipo especializado de arquitectura de redes neuronales artificiales diseñadas para procesar datos con una estructura de cuadrícula, como imágenes o datos de series temporales. La característica principal de las redes convolucionales es la incorporación de capas convolucionales, que aplican filtros convolucionales a los datos de entrada. Estos filtros son matrices pequeñas que se deslizan por la imagen para extraer características locales en diferentes ubicaciones. La convolución permite capturar patrones espaciales en los datos, como bordes, texturas o formas específicas.

Sistema de álgebra computacional (CAS): software que permite manipular de manera sencilla expresiones matemáticas simbólicas con un grado de dificultad elevado y calcular el resultado de dichas expresiones.

Sistema de encerrojamiento: sistema que permite el movimiento síncrono de las dos agujas y, además, una vez posicionadas y desaparecida la fuerza que las ha impulsado, debe mantenerlas en esa posición, sin verse afectadas por vibraciones y golpes producidos al paso de las circulaciones.

Vida útil remanente (RUL): es la cantidad de tiempo restante antes de que un sistema deje de trabajar dentro de sus límites de funcionamiento.

ÍNDICE DE TABLAS

Tabla 1. Modelos clásicos basados en datos 35
Tabla 2. Tabla resumen de los 41 parámetros medidos en una inspección de un aparato de vía tipo A 84
Tabla 3. Tabla resumen del vector de datos de 29 variables 85
Tabla 4. Tabla resumen del vector de datos de 12 variables 86
Tabla 5. Tabla resumen del vector de datos de 6 variables 87

ÍNDICE DE FIGURAS

Figura 1. Partes de un riel 19
Figura 2. Partes de un desvío ferroviario [5] 21
Figura 3. Zona de cambio de vía dentro de un desvío ferroviario [5] 21
Figura 4. Zona de cruzamiento dentro de un desvío ferroviario 22
Figura 5. Medición del ancho de vía 23
Figura 6.Esquema del paso libre de rueda en el cambio y la entrecalle mínima 24
Figura 7. Esquema de la cota de protección y la entrecalle carril-contracarril 24
Figura 8. Esquema de la Ciencia de Datos [7] 28
Figura 9. Distribución de modelos basados en datos para el mantenimiento predictivo de vías férreas [12] 34
Figura 10. Diagrama de la metodología CRIPS-DM [37] 37
Figura 11. Diagrama de las subtareas de la metodología CRISP-DM [37] 41
Figura 12. Esquema de campos de aplicación de datos sintéticos 45
Figura 13. Estimación del uso de datos sintéticos en aplicaciones AI hasta el año 2030 47
Figura 14. Esquema gráfico del Análisis de Componentes Principales 49
Figura 15. Ejemplo de nuevo sistema de coordenadas de componentes principales 49
Figura 16. Ejemplo de porcentaje de varianza explicada por cada componente principal 50
Figura 17. Ejemplo de implementación de PCA en Python 56
Figura 18. Resultado de la implementación de PCA en Python 57
Figura 19. Generación de datos sintéticos en los parámetros “entrecalle mínima” y “paso libre de rueda en el cambio”, dentro del SCRIPT 1 61
Figura 20. Generación de datos sintéticos en los parámetros “cota de protección” y “entrecalle carril-contracarril”, dentro del SCRIPT 1 62
Figura 21. Data Frame generado por el SCRIPT 1 63
Figura 22. Proporciones de varianza arrojadas por el PCA implementado en el SCRIPT 2 65
Figura 23. Diagramas de barras de las proporciones de varianza arrojadas por el PCA aplicado a 12 variables 66
Figura 24.Diagramas de barras de las proporciones de varianza arrojadas por el PCA aplicado a 6 variables 67
Figura 25. Resultados de la ejecución del SCRIPT 4 69
Figura 26. Influencia del tamaño de muestra en la precisión del PCA 71
Figura 27. Prueba de validación del "ancho de vía en el punto de medición de la entrecalle mínima"72
Figura 28. Hoja de control de inspección de aparato de vía tipo A 83

Figura 29. Prueba de validación de la variable "entrecalle mínima" 87
Figura 30. Prueba de validación de la variable "paso libre de rueda en el cambio" 88
Figura 31. Prueba de validación de la variable "entrecalle carril-contracarril" 88
Figura 32. Prueba de validación de la variable "cota de protección" 89
Figura 33. Prueba de validación de dos variables de zonas distinta de la estructura 89

ÍNDICE

RESUMEN 3

ABSTRACT 4

AGRADECIMIENTOS 5

GLOSARIO 6

ÍNDICE DE TABLAS 9

ÍNDICE DE FIGURAS 10

1 INTRODUCCIÓN 14

1.1 CONTEXTUALIZACIÓN 14

1.2 ALCANCE Y OBJETIVOS 16

1.3 MARCO TÉCNICO 19

2 ESTADO DEL ARTE 30

3 METODOLOGÍA 37

3.1 MODELO ANALÍTICO 37

3.2 COMPRESIÓN DE LOS DATOS 42

3.3 MODELADO DE DATOS: PCA 48

3.4 IMPLEMENTACIÓN PCA EN PYTHON 53

4 APLICACIÓN AL CASO DE ESTUDIO 58

4.1 ANÁLISIS DE LOS DATOS INICIALES 58

4.2 GENERACIÓN DATOS SINTÉTICOS 60

4.3 ESTUDIO DE CORRELACIONES CON PCA 64

4.4 SIMULACIÓN DE ESCENARIOS PARA VALIDACIÓN 70

5 CONCLUSIONES 74

BIBLIOGRAFÍA 76

ANEXO I 83

ANEXO II 90

1 INTRODUCCIÓN

1.1 CONTEXTUALIZACIÓN

Desde que en 1825 circuló el primer ferrocarril recorriendo la línea Stockton-Darlington, el transporte ferroviario ha contribuido de manera notable al desarrollo económico, social y cultural en todo el mundo. El ferrocarril supuso una pieza fundamental en la revolución industrial, aceleró el crecimiento económico del mundo pues impulsó el comercio y favoreció la reducción de costes en la producción. Y, gracias al avance de la tecnología, el ferrocarril ha ido evolucionando hasta convertirse en uno de los medios de transporte más empleado en todo el mundo, formando una red que comunica ciudades de todo el mundo y que permite transportar elevados volúmenes de mercancía a bajo coste.

Según el INE, durante el año 2022, en España se transportaron 22,3 millones de toneladas de mercancía a través de sistemas ferroviarios y aproximadamente 540 millones de pasajeros se desplazaron empleando este medio de transporte [1]. Estos datos son un reflejo de la importancia de la red ferroviaria para el desarrollo social y económico del país en la actualidad.

La creciente preocupación por el cambio climático que está sufriendo el planeta, está provocando que países de todo el mundo se unan para adoptar, de manera conjunta, políticas y estrategias como la Agenda 2030 o el Pacto Verde Europeo, con el fin de reducir drásticamente la emisión de gases contaminantes. En este contexto, el ferrocarril adquiere un papel esencial en la ayuda para cumplir con estos objetivos, ya que es la forma de transporte más sostenible del mundo, generando solamente el 0,5 % de las emisiones de gases de efecto invernadero asociadas a los sistemas de transporte [2]. Por ello, muchas de las estrategias comentadas con anterioridad están enfocadas a realizar una mayor inversión, destinada a la innovación y mejora de las infraestructuras ferroviarias.

Además de ser el transporte más sostenible y menos contaminante del mundo, otro factor en el que destaca el ferrocarril es en la seguridad. Es el modo de transporte terrestre con la menor incidencia de accidentes mortales, convirtiéndolo en el más seguro dentro de esta categoría.

Dentro de la infraestructura ferroviaria, los aparatos de vía, comúnmente conocidos como cambios de vías o desvíos, juegan un papel esencial en la conexión de la red ferroviaria, ya que permiten las intersecciones de vías y realizar cambios de direcciones, creando de esta manera innumerables rutas que componen todo el sistema ferrovial. Un dato que manifiesta la importancia de los aparatos de vía dentro del sistema ferroviario es que, a pesar de componer menos del 5% de la infraestructura completa, abarcan alrededor del 17 % del presupuesto de mantenimiento.[3]

Por otro lado, en las últimas décadas se ha producido una generación masiva de datos como consecuencia del avance de la tecnología y la digitalización de diversos ámbitos.

La expansión de la conectividad global y el acceso generalizado a Internet, han permitido a personas, empresas y distintos dispositivos generar y compartir datos a una escala sin precedentes. Las redes sociales han reunido grandes volúmenes de información personal y social, incluyendo distintos formatos como imágenes y videos. Las plataformas de comercio electrónico, como Amazon, han acumulado datos sobre las preferencias de compra de millones de usuarios. Los motores de búsqueda, como Google, registran las consultas de búsqueda de miles de millones de personas cada día.

Además, el desarrollo de los dispositivos móviles inteligentes y de los sensores que llevan incorporados, ha permitido que sean capaces de recopilar información sobre nuestra ubicación geográfica, actividades diarias, interacciones sociales y hábitos de consumo.

Otra tecnología que ha ayudado al crecimiento de la generación de datos es la tecnología Internet of Things (IoT), gracias a la cual se produce la interconexión de dispositivos y sensores en diversos entornos, como hogares inteligentes, ciudades inteligentes, industria manufacturera y atención médica. Los sensores y dispositivos IoT recopilan datos en tiempo real sobre el medio ambiente, la infraestructura, los procesos industriales y otros aspectos, lo que proporciona oportunidades para el análisis y la optimización de gran alcance.

La evolución exponencial de la capacidad de almacenamiento y del procesamiento computacional también han sido dos factores fundamentales en el crecimiento vertiginoso de los datos. Las empresas u organizaciones ahora pueden retener y analizar grandes volúmenes de información utilizando tecnologías como el almacenamiento en la nube y el procesamiento distribuido.

Esta generación masiva de datos, junto con los avances tecnológicos comentados, ha impulsado el desarrollo de técnicas de análisis de datos avanzadas, como el aprendizaje automático, cuya función principal es extraer información valiosa de los datos con el objetivo de generar conocimientos accionables o ayudar en la toma de decisiones.

El Machine Learning es una rama multidisciplinar de la inteligencia artificial que se centra en el desarrollo de algoritmos y modelos que permiten a las computadoras aprender y mejorar su rendimiento en tareas específicas, a través de la experiencia y la observación de datos. En la actualidad se aplica en una amplia variedad de campos y tiene numerosas aplicaciones prácticas como la ayuda en la toma de decisiones, la mejora de la eficiencia operativa o la identificación de patrones y tendencias. En esta última aplicación, destaca el modelo de análisis de componentes principales, también conocido como Principal Components Analysis (PCA) en inglés.

Además, el Machine Learning es un campo en constante evolución, donde se investigan y desarrollan continuamente nuevos enfoques y técnicas para mejorar el rendimiento y la eficiencia de los modelos. Sin embargo, también presenta desafíos, como el almacenamiento y procesamiento eficiente de

grandes volúmenes de datos, la privacidad y la seguridad de la información, la calidad y veracidad de los datos, y la necesidad de habilidades especializadas en análisis de datos.

El desarrollo de algoritmos estadísticos avanzados basados en datos, como es el caso del PCA, unido a la capacidad de almacenamiento de grandes volúmenes de datos en tiempo real, han provocado un consecuente surgimiento de nuevas herramientas y estrategias en otros campos, debido al nuevo conocimiento generado por estas técnicas innovadoras. Es el caso del mantenimiento predictivo, estrategia de mantenimiento desarrollada gracias a la recopilación masiva de datos en tiempo real y a la evolución de las técnicas de análisis de datos, que facilitan la detección temprana de anomalías y la identificación de patrones de fallo en los equipos.

Dentro de este marco, en el que se destaca la importancia de los sistemas ferroviarios y los avances experimentados en el campo de las tecnologías de la información, ha surgido la posibilidad de realizar un trabajo de investigación en paralelo a un proyecto actual de ingeniería, contando con la colaboración y supervisión de David García Sánchez, coordinador del aula Tecnalia, empresa responsable del proyecto. El objeto de investigación es el desarrollo de un modelo predictivo, con el fin de anticipar la detección de anomalías estructurales en segmentos especiales de vía antes de que estas se produzcan, permitiendo de esta manera realizar un mantenimiento predictivo y reducir considerablemente los costes de operación. Para llevar a cabo este proyecto, aprovechando la destacada evolución tecnológica en los campos de la inteligencia artificial y la ciencia de datos, se emplearán algunas técnicas de análisis y tratamiento de datos como Machine Learning.

Este trabajo se enmarca dentro del programa "AULA TECNALIA" mediante el cual se establece una colaboración entre la Universidad de Cantabria y Tecnalia con el objetivo de elaborar trabajos fin de Master bajo la tutorización compartida Universidad y el Centro Tecnológico. Tecnalia es el primer centro de investigación aplicada y desarrollo tecnológico en España y posee un importante departamento de investigación en Edificios e Infraestructuras Inteligentes y Resilientes dentro de la división de Transición Energética, Climática y Urbana.

1.2 ALCANCE Y OBJETIVOS

En la actualidad, el plan de mantenimiento de muchas vías ferroviarias y elementos especiales de vía en España sigue una estrategia de mantenimiento preventivo, basado principalmente en inspecciones periódicas programadas. Estas inspecciones se realizan en intervalos regulares definidos según el tipo de elemento examinado, en ellas se analiza el estado de la estructura ferroviaria y, en función de ese estado, se define una acción de mantenimiento.

Este proyecto se ha desarrollado para el estudio concreto de los aparatos de vía de tipo A. Este tipo de aparatos de vía se inspeccionan mediante reconocimientos visuales de las vías y empleando equipos de medición especializados para detectar irregularidades en los rieles como desgaste o deformaciones. Tanto el reconocimiento visual como las mediciones realizadas se registran en un formulario de control (Véase la Figura 28 del ANEXO I) que posteriormente es evaluado con el fin de determinar la acción de control necesaria.

El mantenimiento predictivo es una técnica utilizada en la industria para predecir fallos o problemas en los equipos y maquinaria antes de que ocurran, lo que permite tomar acciones preventivas y evitar paradas no planificadas. Se trata de una estrategia que mejora al mantenimiento preventivo, permitiendo reducir los costes de operación y mejorar la seguridad. Por ello es una estrategia cada vez más implementada en diversos sectores industriales. El Machine Learning, es una herramienta fundamental para poder desarrollar un plan de mantenimiento predictivo, ya que es capaz de analizar grandes cantidades de datos e identificar patrones y tendencias ocultas.

Así pues, el presente proyecto pretende desarrollar un algoritmo basado en el análisis de datos que permita evolucionar hacia un modelo de mantenimiento predictivo de los segmentos especiales de vía como aparatos de vía o dilatadores.

Comprende los siguientes objetivos técnicos principales:

- Análisis del potencial de los métodos de detección de anomalías basados en datos, proponiendo un nuevo planteamiento que incorpore las técnicas de aprendizaje automático mediante el reconocimiento de patrones estadísticos.
- Diagnosis o evaluación de la condición de los aparatos de vía que permite detectar, identificar o localizar el fallo.
- Implementación de una herramienta de valor que permita evolucionar la estrategia de mantenimiento hacia una gestión predictiva del mismo.
- Recomendación en términos de mantenimiento.

Para llevar a cabo los objetivos principales propuestos se desarrollarán las siguientes tareas:

- Seleccionar los parámetros e indicadores que mejor definen la condición del activo (segmento especial de vía), basándose en el estado del arte actual para el control y seguimiento de los segmentos especiales de vía.
- Diseñar un algoritmo para la detección de daños estructurales en los segmentos especiales de vía.

- Verificar los algoritmos diseñados mediante datos sintéticos.
- Proponer estrategias de intervención y mantenimiento

1.3 MARCO TÉCNICO

Debido a los tecnicismos de diversos ámbitos utilizados en el desarrollo de este trabajo, se ha incluido este apartado donde se detallan conceptos y parámetros con los que se ha trabajado a lo largo de todo el proyecto.

Partes de una vía ferroviaria

1. **Riel**: Los rieles ferroviarios son las barras de acero alargadas y estrechas que forman la pista sobre la cual se desplazan los trenes. Los rieles se colocan en pares y están sujetos a los durmientes o traviesas. Pueden ser de diferentes perfiles y tamaños dependiendo del tipo de vía y la carga esperada. El riel está formado por tres partes:
    - ❖ Hongo: es la superficie sobre la cual las ruedas de los trenes se desplazan. Tiene una forma plana o ligeramente convexa para garantizar un contacto adecuado con las ruedas y minimizar el desgaste.
    - ❖ Alma: es la parte central del riel que conecta la cabeza y la base. Proporciona resistencia y rigidez al riel para soportar las cargas y fuerzas que actúan sobre él.
    - ❖ Patín: es la parte inferior del riel que descansa sobre los durmientes o traviesas. Proporciona estabilidad y soporte estructural al riel.

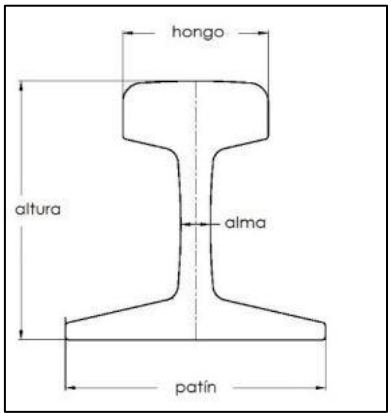


Figura 1. Partes de un riel

2. **Durmientes**: también conocidos como traviesas, son las estructuras horizontales que sostienen y mantienen los rieles en su posición correcta. Pueden ser de madera o acero. Los durmientes se instalan a intervalos regulares a lo largo de la vía para soportar el peso del tren y distribuirlo de manera uniforme.
3. **Balasto**: se trata de una capa de material granular (generalmente grava o piedra triturada) que se coloca debajo de los durmientes. El balasto actúa como una base firme que ayuda a

distribuir las cargas del tren y a drenar el agua de la vía. Además, ayuda a amortiguar las vibraciones y proporciona estabilidad a los rieles.

4. **Eclisa ferroviaria**: también conocida como junta de riel o junta de carril, es un dispositivo utilizado para unir dos rieles ferroviarios en una vía. La eclisa ferroviaria consta de dos piezas de acero que se colocan en cada lado de los rieles adyacentes. Estas piezas generalmente tienen una forma en U y están diseñadas para abrazar los rieles en los extremos. Se sujetan a los rieles mediante pernos y tuercas. Su objetivo principal es proporcionar una conexión segura y resistente entre los dos rieles, asegurando que estén alineados correctamente y manteniendo la continuidad de la pista.

Aparatos de vía

Un aparato de vía o desvío es un dispositivo esencial en sistemas ferroviarios que permite el cambio de dirección, la distribución del tráfico, el cruce de vías y otras operaciones necesarias para guiar y dirigir el tráfico ferroviario. Según su ubicación, cada aparato de vía cumple una función específica para la que ha sido diseñado y contribuye a la seguridad y eficiencia del transporte ferroviario.

En España, existen seis tipos de desvíos normalizados en función de su longitud y de la velocidad máxima de paso permitida [4].

I. **Tipo A**: velocidad máxima de paso por vía directa de 140 km/h y de 30 por desviada.
II. **Tipo B**: velocidad máxima de paso por vía directa de 160 km/h y de 60 por desviada.
III. **Tipo C**: velocidad máxima de paso por vía directa de 200 km/h y de 60 por desviada.
IV. **Tipo V**: velocidad máxima de paso por vía directa de 200 km/h y de 100 por desviada.
V. **Tipo AV**: velocidad máxima de paso por vía directa de 300 km/h y de 160 por desviada.
VI. **Tipo AV+**: velocidad máxima de paso por vía directa de 350 km/h y de 220 por desviada.

A su vez, un desvío se divide en tres zonas bien diferenciadas, como se puede observar en la Figura 2. [5]

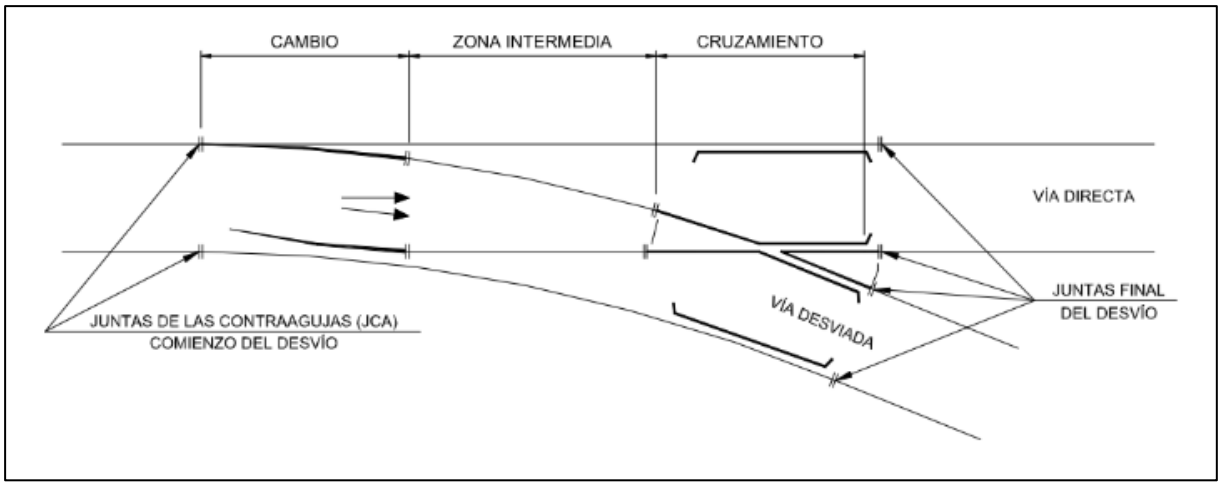


Figura 2. Partes de un desvío ferroviario [5]

a) Cambio: Se denomina cambio a la parte del desvío donde se lleva a cabo la separación de los carriles, permitiendo dirigir el sentido del tráfico. Para tal fin, el cambio posee unos conjuntos de agujas-contraagujas dispuestos sobre traviesas que permiten desviar al tren en la dirección deseada (vía directa o desviada) [5]. Las agujas son elementos móviles excepto en su extremo más próximo al cruzamiento, llamado talón, y se mueven solidariamente mediante un tirante. Sin embargo, las contraagujas son fijas y exteriores a éstas. Para accionar las agujas se disponen motores, quedando acopladas en su posición final a las contraagujas mediante un sistema de encerrojamiento.

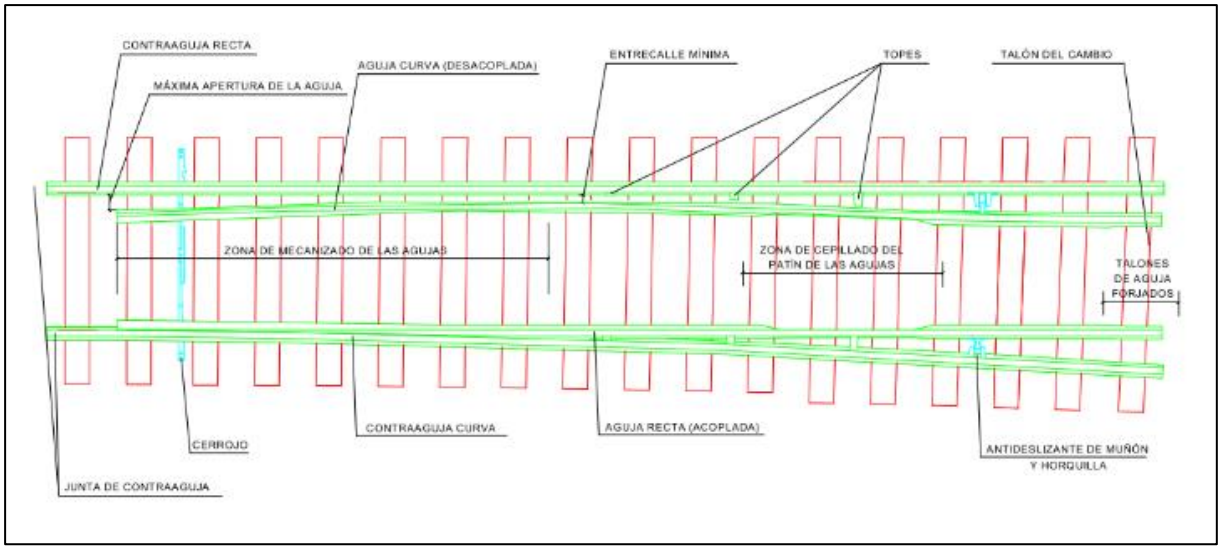


Figura 3. Zona de cambio de vía dentro de un desvío ferroviario [5]

b) **Carriles de unión**: se denomina a la parte intermedia de un desvío, la cual une el cambio con el cruzamiento. Esta parte está formada, generalmente, por cuatro carriles, cuyo perfil es el mismo que el que corresponde a un carril convencional. De esta forma, dos de los carriles

corresponden a la vía directa y los otros dos restantes a la vía desviada. En algunos casos se sitúan juntas aislantes, permitiendo separar la señal eléctrica en el desvío.

c) **Cruzamiento**: está formado por tres elementos principales: corazón, contracarriles y carriles exteriores. El corazón se trata de una pieza muy robusta, puede ser recto, curvo e incluso móvil, y su objetivo es el de guiar a las ruedas en la intersección. Se trata de un elemento crítico ya que está sometido a frecuentes impactos al paso de las ruedas. Además, es en él donde se materializa el corte o discontinuidad de uno de los carriles de la vía directa, lo que se conoce como laguna. Por otro lado, los extremos de los carriles interiores del cruzamiento se denominan patas de liebre, las cuales soportan el peso de la rueda cuando circula por la laguna. Por último, los contracarriles se componen de un trozo de carril o de perfil especial situado en el entorno del corazón, muy próximos al carril. Su objetivo es el de evitar que la ruedas puedan descarrilar al paso por la laguna e impedir que tomen una dirección errónea.

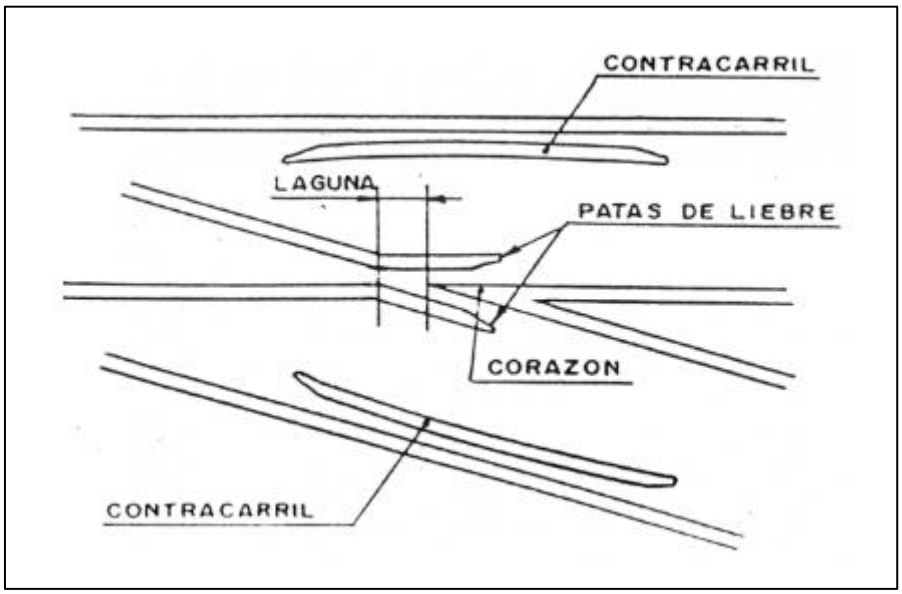


Figura 4. Zona de cruzamiento dentro de un desvío ferroviario

Como se puede observar en la Figura 2, en un desvío ferroviario se distinguen dos vías principales: la vía directa y la vía desviada.

- **Vía Directa**: también conocida como vía principal o vía recta, es la vía que continúa en línea recta sin desviarse después del desvío. Cuando un tren se dirige por la vía directa, sigue su curso normal sin realizar ningún cambio de dirección. Esta vía generalmente es la ruta principal y se utiliza para los trenes que no necesitan desviarse hacia vías secundarias o ramales.

❖ **Vía Desviada**: también conocida como vía secundaria o vía derivada, es la vía a la que se dirige un tren después de pasar por el desvío. En lugar de continuar recto, el tren se desvía hacia esta vía, cambiando su dirección original. La vía desviada puede llevar al tren a una vía secundaria, un ramal o incluso a otra línea principal, dependiendo de la configuración y diseño del sistema ferroviario.

Dentro de los distintos tipos de aparatos de vía, el presente proyecto se centra en implementar un modelo de detección y predicción de defectos en los aparatos de vía ferroviaria de tipo A. Existe un conjunto de parámetros que permiten medir y evaluar el estado de este tipo de aparatos de vía. Dichos parámetros, con los que posteriormente se va a trabajar para implementar un algoritmo de reconocimiento de patrones estadísticos, vienen definidos en la norma de ADIF NAV 7-3-8.2 [6] y se van a detallar brevemente a continuación:

- **Ancho de vía**: es la distancia menor entre las líneas perpendiculares al plano de rodadura y que cortan a cada perfil de la cabeza del carril en un margen entre 0 y 14 mm por debajo de dicho plano (Véase Figura 5). El ancho de vía se mide en varios puntos del aparato de vía.

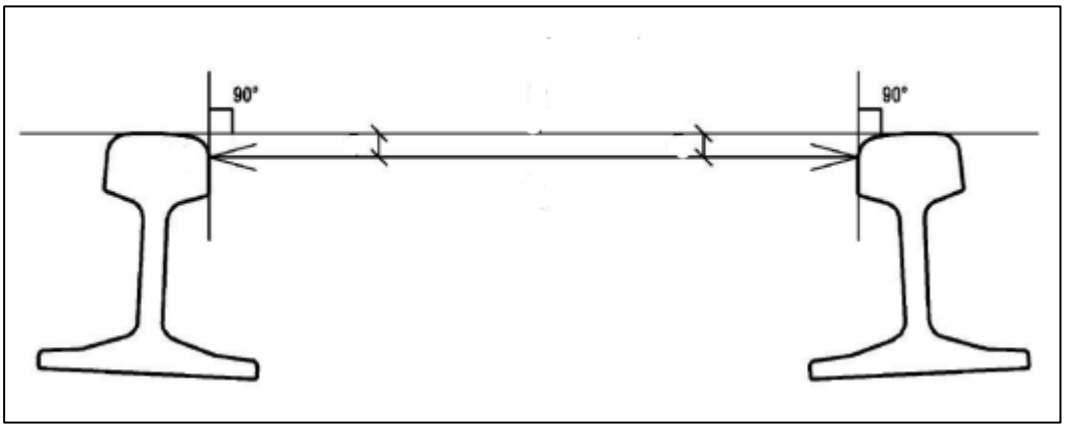


Figura 5. Medición del ancho de vía

- **Ancho de vía directa**: es el ancho de la vía directa del aparato.
- **Ancho de vía desviada**: es el ancho de la vía desviada del aparato
- **Anchura mínima de la garganta de guía**: distancia mínima horizontal en las entrecalles corazón-pata de liebre. La anchura de la garganta de guía será tal que al paso de las ruedas no se produzcan choques o desgastes en la garganta. Se realiza una medida para cada vía.
- **Entrecalle mínima**: es la distancia mínima entre la cara activa de la contraaguja y la cara inactiva de la aguja cuando esta se encuentra desacoplada. Se medirá en el punto más próximo entre aguja y contraaguja. Se realiza una medida para cada una de las agujas. (Véase Figura 6)

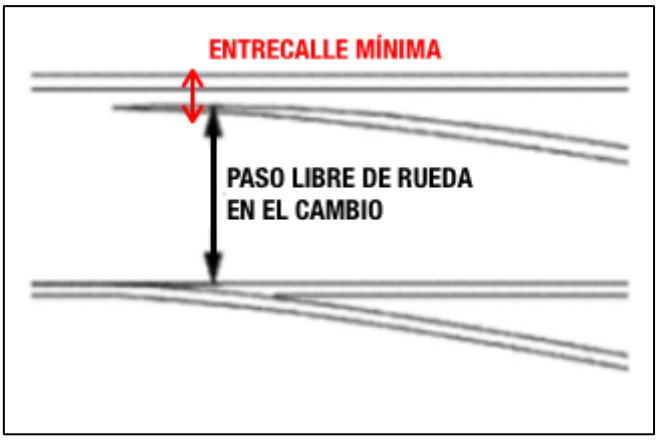


Figura 6.Esquema del paso libre de rueda en el cambio y la entrecalle mínima

- **Paso libre de rueda en el cambio**: es el valor resultante de la resta de la entrecalle mínima al ancho de vía. Se realiza una medida para cada una de las agujas. (Véase Figura 6)
- **Cota de protección**: es la distancia que debe existir entre la cara activa del contracarril y la punta del corazón que impide que el tren golpee la punta del corazón. Se realiza una medida para cada vía. (Véase Figura 7)

Figura 7. Esquema de la cota de protección y la entrecalle carril-contracarril

- **Entrecalle carril-contracarril**: distancia entre la contraaguja y el contracarril, igual a la resultante de la resta entre la cota de protección y el ancho de vía. Debe ser lo bastante amplia como para que no se produzca el enconamiento de la pestaña de rueda de mayor espesor posible. Se realiza una medida para cada vía. (Véase Figura 7)

- **Paso libre de rueda en las puntas de cruzamiento**: distancia entre el contracarril y la cara vertical de la pata de liebre a la altura de la punta del corazón. El contracarril y la pata de liebre deben estar posicionados de manera que permitan el paso de la rueda en la punta del cruzamiento. Se realiza una medida para cada vía.
- **Paso libre de rueda en la entrada de pata de liebre**: es la distancia entre la cara activa de la pata de liebre y la cara activa del carril opuesto. Se mide en el punto de entrada de la pata de liebre con objeto de evitar el choque o roce entre la cara interna de la rueda y la pata de liebre. Se realiza una medida para cada vía.
- **Paso libre de rueda en la entrada de contracarril**: distancia entre el contracarril y el carril del otro lado medido en la entrada del contracarril. Esta distancia garantiza que el contracarril no trabaja antes del punto de entrada. Se realiza una medida para cada vía.
- **Profundidad mínima de la garganta de guía**: profundidad mínima desde el plano de rodadura hasta el fondo de la entrecalle corazón-pata de liebre. Se realiza una medida para cada vía.
- **Sobreelevación o altura de contracarriles**: el contracarril está diseñado para trabajar de forma activa a una determinada distancia de la contraaguja, diferente para cada tipo de aparato, mejorando el guiado de las ruedas a su paso por el aparato de vía. En el momento en el que el contracarril está más alto de lo diseñado se produce la sobreelevación del contracarril. Se realiza una medida para cada vía.
- **Descuadre de las juntas de contraaguja**: los dos semicambios deben permanecer en paralelo para garantizar el correcto funcionamiento del aparato. El descuadre de las juntas de contraaguja o el descuadre de las puntas de las agujas indicarán que el aparato no está bien alineado. Se mide mediante una escuadra de vía para la formación de un triángulo determinando el descuadre de los granetazos que fijan la posición teórica de la punta de las agujas. Se realiza una medida para cada vía.

Tipos de mantenimientos

Existen diversas clasificaciones de los modelos de mantenimiento según la fuente consultada y según el ámbito o sector donde se refiera. En este documento, se van a definir los tres tipos de mantenimiento que son de carácter general para todos los sectores industriales:

Mantenimiento correctivo: es una estrategia de mantenimiento que se lleva a cabo cuando ocurre una avería o un fallo en un equipo, sistema o infraestructura. Es decir, el mantenimiento correctivo se realiza después de que se haya producido un problema y tiene como objetivo principal restaurar el funcionamiento normal lo más rápido posible.

El mantenimiento correctivo puede ser reactivo, si se realiza en respuesta directa a un fallo o problema que ha sido detectado, o puede ser programado cuando se tiene conocimiento de la necesidad de una reparación específica.

Cuando ocurre una avería, se sigue un procedimiento general para el mantenimiento correctivo. En primer lugar, se realiza una evaluación inicial para determinar la causa y la gravedad del fallo. Una vez que se ha identificado la causa del fallo, se procede a tomar medidas para corregirla.

Es importante tener en cuenta que el mantenimiento correctivo se considera una medida de contingencia y no es un enfoque ideal para el mantenimiento a largo plazo. Puede resultar costoso y disruptivo, ya que implica tiempos de inactividad no planificados y puede afectar la productividad y la eficiencia operativa.

Sin embargo, el mantenimiento correctivo sigue siendo una parte esencial del panorama general del mantenimiento, ya que es necesario abordar las averías y problemas que inevitablemente ocurren. Es particularmente relevante cuando no se ha implementado un mantenimiento preventivo o predictivo adecuado, lo que aumenta el riesgo de fallos y la necesidad de intervenciones correctivas.

Mantenimiento preventivo: es una estrategia de mantenimiento que se lleva a cabo de manera planificada y regular con el objetivo de prevenir fallos, averías y problemas en los equipos, sistemas o infraestructuras. A diferencia del mantenimiento correctivo, que se realiza después de que ocurra un problema, este tipo de mantenimiento busca anticiparse a los fallos y tomar medidas preventivas para garantizar un funcionamiento seguro y prolongar la vida útil de los activos.

La estrategia preventiva se basa en la idea de que es más eficiente y económico prevenir problemas antes de que ocurran, en lugar de esperar a que se produzcan y luego llevar a cabo reparaciones costosas y tiempos de inactividad prolongados.

El desarrollo de un plan de mantenimiento preventivo involucra varias etapas. En primer lugar, se realiza un análisis de los equipos y sistemas para identificar los componentes críticos y las áreas propensas a fallos. Se establecen intervalos de tiempo o hitos basados en el tiempo de funcionamiento, la edad del equipo o los requisitos del fabricante para realizar inspecciones, ajustes y actividades de mantenimiento.

Algunas de las actividades que implican este tipo de mantenimiento son: inspecciones regulares, lubricación y limpieza, ajuste y calibraciones, reemplazo de piezas y componentes, pruebas y verificaciones o actualizaciones y mejoras, entre otras.

La implementación de un plan de mantenimiento preventivo ofrece múltiples beneficios. Entre ellos se encuentran la reducción de los tiempos de inactividad no planificados, la mejora de la seguridad

operativa, el aumento de la vida útil de los activos, la optimización del rendimiento y la eficiencia, y el control de los costes de mantenimiento a largo plazo.

Este tipo de modelo de mantenimiento se aplica en una gran variedad de sectores, desde la industria manufacturera hasta el mantenimiento de vehículos, equipos electrónicos, sistemas de energía, y más. Los sistemas ferroviarios y en concreto los aparatos de vía también utilizan la estrategia preventiva.

Mantenimiento predictivo: es una estrategia de mantenimiento que se basa en la monitorización y análisis continuo de los equipos y sistemas para predecir posibles fallos y planificar acciones preventivas antes de que ocurran problemas graves. Esto permite realizar acciones de mantenimiento preventivo de manera más eficiente y reducir los tiempos de inactividad no planificados.

A diferencia del mantenimiento preventivo, que se basa en intervalos de tiempo predefinidos, el mantenimiento predictivo utiliza tecnologías avanzadas y datos en tiempo real para tomar decisiones informadas sobre el mantenimiento.

El mantenimiento predictivo se centra en la recolección y análisis de datos de los equipos. Estos datos se recopilan mediante el uso de sensores, instrumentos de monitorización o sistemas de supervisión específicos. Luego, se analizan utilizando técnicas y herramientas de análisis, como el análisis de tendencias, el análisis de espectro de frecuencia, la inteligencia artificial o el aprendizaje automático.

A partir del análisis de los datos, se pueden obtener indicadores y patrones que permiten predecir el comportamiento y el estado de los equipos. Esto incluye la identificación de tendencias, cambios significativos o anomalías en los datos que pueden indicar problemas en desarrollo. Con esta información, se pueden tomar decisiones sobre el mantenimiento necesario, como reparaciones, ajustes, sustitución de componentes o incluso paradas programadas.

Es importante destacar que la implementación del mantenimiento predictivo requiere una infraestructura adecuada, incluyendo sensores, sistemas de monitorización, herramientas de análisis de datos y personal capacitado en su uso. Además, se necesita una gestión eficiente de los datos recopilados y un análisis adecuado para obtener resultados significativos y de calidad.

El mantenimiento predictivo frente al preventivo permite mejorar la capacidad de diagnóstico, lo cual a su vez hace posible planificar mejor las intervenciones reduciendo los periodos de inactividad, aprovechar mejor los intervalos de trabajo, determinar los procedimientos correctivos óptimos y, como consecuencia de todo ello, reducir y ajustar los costes de mantenimiento.

En sectores como el automovilístico, el ferroviario o el espacial, así como en industrias donde las paradas en la cadena de producción tienen un impacto directo en la facturación, este tipo de estrategia tiene un alto impacto positivo en seguridad y a nivel económico.

Ciencia de datos (Data Science)

A lo largo de todo el documento se va a hacer referencia a diferentes técnicas de Machine Learning no supervisado y terminología relacionada con el campo de la ciencia de datos. Por ello, se van a definir varios conceptos que pongan en contexto y ayuden a entender mejor el trabajo descrito en este documento.

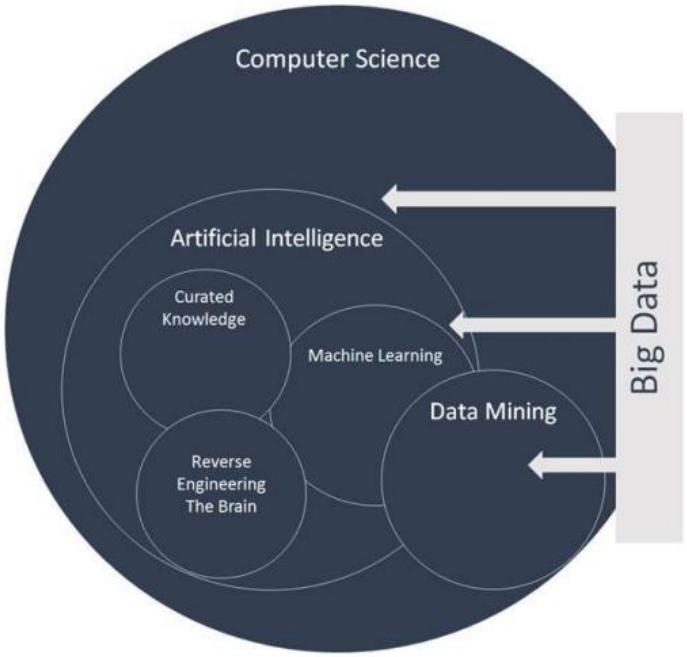


Figura 8. Esquema de la Ciencia de Datos [7]

La **ciencia de datos** es un campo multidisciplinario que se centra en la extracción de conocimientos y perspectivas a partir de conjuntos de datos estructurados y no estructurados. Utiliza técnicas estadísticas, matemáticas, computacionales y de visualización para analizar grandes volúmenes de datos y descubrir patrones, tendencias y correlaciones que pueden ser utilizados para tomar decisiones informadas y resolver problemas complejos.

La ciencia de datos involucra diferentes etapas en el proceso de análisis de datos. En primer lugar, implica la recopilación y preparación de datos, donde se seleccionan y limpian los conjuntos de datos relevantes para el análisis. Luego, se realiza el análisis exploratorio de datos para comprender mejor las características y relaciones presentes. A continuación, se aplican técnicas y algoritmos de modelado de datos para generar modelos predictivos o descriptivos. Finalmente, se interpretan los resultados y se comunican a las partes interesadas de una manera comprensible.

Por otro lado, como se puede observar en la Figura 8, la **minería de datos** o (**Data Mining**), es un campo de la ciencia de datos y de la estadística enfocado al descubrimiento de patrones, tendencias y relaciones en grandes conjuntos de datos. El objetivo principal de la minería de datos es descubrir conocimientos ocultos o implícitos en los datos, que puedan ser utilizados para tomar decisiones o realizar predicciones. Para ello, emplea técnicas de Machine Learning basadas en modelos estadísticos y sistemas de bases de datos como la técnica PCA, la cual se implementará en el desarrollo del presente proyecto.

Por su parte, la **inteligencia artificial** (**IA**) es un campo más amplio que se centra en la creación de sistemas y programas capaces de realizar tareas que requieren inteligencia humana. La IA busca desarrollar sistemas que puedan razonar, comprender, aprender, planificar y adaptarse, entre otras capacidades cognitivas. El aprendizaje automático es una de las técnicas utilizadas dentro del campo de la inteligencia artificial, pero la IA también puede incluir otras técnicas, como el procesamiento del lenguaje natural, la visión por computadora o la robótica, entre otras.

Otra de las subdisciplinas de la ciencia de datos es el **Machine Learning**. Se centra en el desarrollo de algoritmos y modelos matemáticos y estadísticos que permiten a las máquinas aprender y mejorar automáticamente a partir de datos sin ser programadas explícitamente. Permite a las computadoras analizar datos, identificar patrones y tomar decisiones o realizar predicciones sin una intervención humana constante. Como se puede observar en la Figura 8, los diferentes modelos que integran esta disciplina son utilizados dentro del campo de la IA y de la minería de datos.

El aprendizaje automático se puede clasificar en cuatro tipos principales: aprendizaje supervisado, aprendizaje no supervisado, aprendizaje semisupervisado y aprendizaje por refuerzo. En el aprendizaje supervisado, se entrenan modelos utilizando datos etiquetados previamente, es decir, se proporciona información de entrada y salida para que el algoritmo aprenda a hacer predicciones. Son difíciles de implementar por la gran cantidad de información que necesitan. En el aprendizaje no supervisado, no hay etiquetas en los datos y el algoritmo busca patrones y estructuras ocultas en los datos. En el aprendizaje semisupervisado, se dispone de un conjunto de datos parcialmente etiquetado, donde solo una fracción de los ejemplos tiene etiquetas asociadas. El objetivo es utilizar tanto los datos etiquetados como los no etiquetados para mejorar el rendimiento de los modelos de aprendizaje automático. Por último, en el aprendizaje por refuerzo, un agente aprende a través de la interacción con un entorno, recibiendo recompensas o castigos en función de sus acciones.

2 ESTADO DEL ARTE

La vía férrea es una de las partes más críticas del sistema ferroviario. Los accidentes causados por el estado de conservación de las vías constituyeron un 30-40% del total de accidentes durante la última década en Estados Unidos [8]. Por lo tanto, para evitar interrupciones en la red ferroviaria, las vías férreas deben ser monitorizadas y mantenidas regularmente.

A nivel mundial, la industria ferroviaria gasta una gran cantidad de dinero en mantenimiento y proyectos de renovación. Por ejemplo, el gasto anual de mantenimiento de la infraestructura ferroviaria británica fue más de mil millones de libras en 2015; casi dos tercios de los empleados de la organización Network Rail se dedican a trabajo de mantenimiento [9]. En los Estados Unidos, más de la mitad de los costes de mantenimiento ferroviario son relacionados con las vías [10].

Para reducir los costes de mantenimiento, es crucial encontrar estrategias de mantenimiento adecuadas. En la literatura, las estrategias de mantenimiento para sistemas ferroviarios incluyen mantenimiento correctivo, preventivo, basado en condiciones y predictivo, las cuales se describieron en el apartado 1.3.

La estrategia de mantenimiento predictivo es la más deseable porque reduce las tasas de fallo en las vías y minimiza los costes de mantenimiento al extender la vida útil de los componentes de la vía y permitir que los operadores planifiquen operaciones de mantenimiento antes de tiempo [11].

Existe una gran abundancia de datos disponible en la industria ferroviaria, así como referencias bibliográficas. En 2020 se publicó un exhaustivo trabajo de revisión bibliográfica sobre el mantenimiento ferroviario, que incluye una sugerente estadística sobre las publicaciones. Se utiliza este trabajo como referencia para analizar el estado del arte del mantenimiento en general de las vías férreas, y más particularmente el mantenimiento predictivo de éstas [12].

Por ejemplo, en 2015 se publicó un artículo [13] que presentaba una metodología para predecir la vida útil remanente (RUL) de ruedas y vagones, mediante la fusión de datos de tres tipos de detectores, incluidos el detector de carga de impacto de rueda, sistemas de visión artificial y detectores de geometría óptica. En este artículo se crea una variedad de nuevas características a partir de la normalización de funciones, las características de la señal y los conjuntos de datos estadísticos históricos. Los datos no disponibles son generados por “missForest”, un algoritmo de imputación de valores faltantes no paramétrico basado en Random Forests. Finalmente, se implementan y comparan varias técnicas de minería de datos para predecir el RUL de ruedas y vagones en una red ferroviaria de clase I de EE. UU. Las pruebas numéricas muestran que la metodología propuesta puede predecir con

precisión el RUL de los componentes de un vagón, particularmente en un rango de tiempo de considerable precisión.

Otro ejemplo de mantenimiento de las vías férreas es un artículo publicado en 2016, que se basa en la detección de defectos ferroviarios por cámaras de video, aprovechando la evolución dentro de la industria de los sistemas de percepción y visión por computación en la última década. Los datos en formato imagen de carril tienen dos etiquetas: defectuoso y no defectuoso. Está muy desequilibrado hacia la clase no defectuosa y tiene una gran cantidad de muestras de datos sin etiquetar disponibles para técnicas de aprendizaje semisupervisado. En este artículo, se investiga si los candidatos defectuosos positivos seleccionados de los datos no etiquetados pueden ayudar a mejorar el equilibrio entre las dos clases y aumentar el rendimiento en la detección de un tipo específico de defectos llamados “squads”. También se comparan las técnicas de muestreo de datos y se concluye que las técnicas semisupervisadas son una alternativa razonable para mejorar el rendimiento en aplicaciones como la detección de “squads” en vías ferroviarias a partir de imágenes.[14]

Según el artículo “A review on machinery diagnostics and prognostics implementing condition-based maintenance” [15], los modelos utilizados para evaluar las condiciones de la vía férrea con fines de diagnóstico y pronóstico se pueden agrupar en modelos mecánicos y modelos basados en datos.

Los **modelos mecánicos** se basan en el conocimiento mecánico del comportamiento de los componentes y se basan en suposiciones simplificadas de los componentes de la vía. Los enfoques basados en datos, que no tienen tales dependencias, se han aplicado cada vez más en el mantenimiento predictivo de vías férreas. El análisis de los conjuntos de datos de medición ferroviaria con enfoques basados en datos ha sido recientemente un área de interés, tanto en el ámbito académico como en el industrial.

Los **métodos basados en datos** descubren conjuntos de características viables y criterios de decisión a partir de los datos observados. Estos métodos incluyen modelos estadísticos y modelos de Machine Learning [16]. La principal diferencia entre estos dos tipos radica en el objetivo principal del análisis. Los modelos estadísticos hacen inferencias sobre las relaciones entre las variables, mientras que los modelos de Machine Learning se centran en hacer precisa predicciones. Ambos tipos pueden manejar datos multivariados y de alta dimensión, y extraer relaciones ocultas entre el estado de la vía y los datos de medición. En general, los métodos basados en datos pueden ayudar a los ingenieros ferroviarios a comprender mejor el estado de la vía férrea y hacer las correspondientes decisiones de mantenimiento. Sin embargo, el rendimiento de los métodos basados en datos depende de la elección adecuada de los modelos de preprocesamiento y análisis de datos.

Hay numerosas revisiones en la literatura sobre la aplicación y los desafíos en el mantenimiento predictivos de las vías férreas. Sin embargo, la mayoría de estos estudios se centran en un aspecto específico de la vía férrea. Por ejemplo, en el artículo publicado por I. Soleimanmeigouni et al. [17], se identifican, clasifican y discuten los modelos existentes sobre la degradación y el mantenimiento de la geometría de la vía. El objetivo es identificar los problemas y desafíos de las diferentes metodologías y modelos disponibles, así como proporcionar observaciones críticas sobre cada contribución. Finalmente, el documento proporciona instrucciones para futuras investigaciones.

Por otra parte, en "A review of rail track degradation prediction models" se desarrolla un estudio de los modelos de predicción de la degradación de la vía basados en modelos mecánicos, modelos estadísticos y modelos de inteligencia artificial [16]. Asimismo, Sol-Sánchez y D'Angelo [18] realizan una revisión de la literatura centrada en la efectividad de las principales técnicas convencionales y materiales para el diseño y mantenimiento de vías, así como soluciones innovadoras que se están desarrollando para reducir la degradación de la vía. Otros artículos referidos a estudios sobre la aplicación del análisis de datos en un aspecto específico de vía férrea se pueden encontrar en la literatura [19][20][21]. Según el conocimiento de los autores, la literatura en este campo sufre de la falta de un estudio global que cubra todas las soluciones basadas en datos en ambos detección, predicción y toma de decisiones de mantenimiento de defectos en las vías férreas.

En el artículo "Systematic Literature Review on Data-Driven Models for Predictive Maintenance of Railway Track: Implications in Geotechnical Engineering" [12], desarrollan un análisis detallado de modelos de mantenimiento predictivo de vías férreas basados en datos proporciona una taxonomía para clasificar la literatura existente basada en tipos de modelos y tipos de aplicaciones. Los autores ponen el énfasis está en la selección de métodos apropiados de extracción de características y modelos basados en datos para diferentes conjuntos de datos, rastrear defectos y estrategias de mantenimiento. Este trabajo proporciona un resumen completo de los enfoques que se están desarrollando a día de hoy en este campo y el desempeño de las técnicas actuales de última generación.

Los modelos basados en datos incluyen modelos estadísticos y de aprendizaje automático. Para los métodos estadísticos, el propósito es estimar una pequeña cantidad de parámetros de una gran colección de muestras. La suposición básica es que los datos se ajustan a una hipótesis específica, como la distribución de Weibull [22]. Esto contrasta con el aprendizaje automático, donde un conjunto generalmente grande de parámetros del modelo se estima a partir de grandes cantidades de muestras.

El modelo de aprendizaje automático puede extraer más información de los datos sin un conocimiento a priori, como es el caso del artículo publicado por M. Rosyidi et al. [23], en el cual se lleva a cabo una

investigación para calcular el RUL de un sistema de cruce de ferrocarril automático con el fin de estimar el periodo de mantenimiento requerido. En este artículo se emplea la técnica de Machine Learning, PCA, con el fin de analizar y validar la vida útil remanente útil calculada.

Los modelos estadísticos se adaptan mejor a la inferencia sobre las relaciones entre parámetros, mientras que el objetivo del aprendizaje automático es hacer las predicciones más precisas, ya sea de regresión o de clasificación. De esta manera, los métodos de aprendizaje automático pueden contribuir a la toma de decisiones de mantenimiento ferroviario de una manera más directa y robusta.

En la Figura 9, obtenida del artículo "Systematic Literature Review on Data-Driven Models for Predictive Maintenance of Railway Track: Implications in Geotechnical Engineering" [12] se proporciona un resumen de la distribución de publicaciones en las que se aplicaron modelos basados en datos en el campo de mantenimiento de predicción de vías férreas. Hay que tener en cuenta que las estadísticas aquí solo incluyen publicaciones que brindan descripciones detalladas de datos de entrada, construcción de modelos y resultados de salida. De esta forma, los autores identificaron un total de 109 publicaciones.

La Figura 9 revela una preferencia por la aplicación de métodos basados en datos donde cada color representa un tipo de método basado en datos. Vale la pena mencionar que la mayoría de los artículos utilizan aprendizaje automático (74 %) en lugar de modelos estadísticos (26 %). Los modelos clásicos de aprendizaje automático dominan en el aprendizaje automático. El algoritmo de aprendizaje automático clásico más empleado fue la máquina de vectores de soporte (SVM) (33 %), seguida de las redes neuronales artificiales (ANN) (26 %) y los modelos basados en árboles (21 %). Los enfoques avanzados de aprendizaje automático de modelos de aprendizaje profundo, modelos de aprendizaje no supervisado y modelos de conjuntos representan un 22% combinado de las aplicaciones.

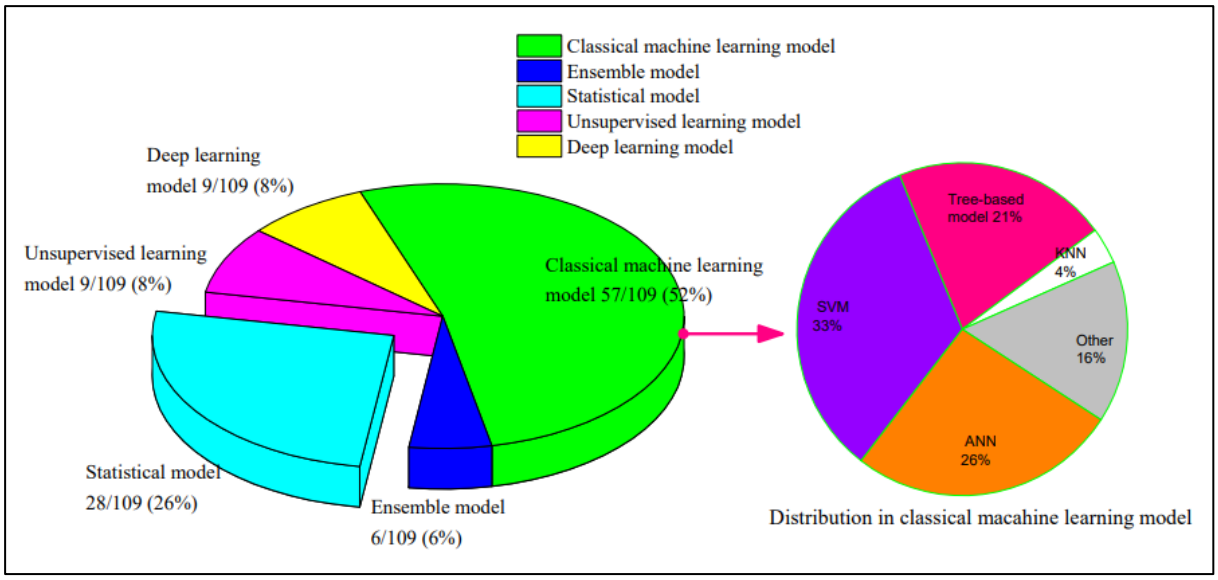


Figura 9. Distribución de modelos basados en datos para el mantenimiento predictivo de vías férreas [12]

Como señalan los autores, hasta la fecha, los investigadores han mostrado un gran interés en los modelos clásicos de aprendizaje automático, especialmente durante los últimos años. El número de artículos sobre modelos estadísticos también ha aumentado lentamente, aunque su proporción de todas las publicaciones ha disminuido ligeramente. Los modelos avanzados de aprendizaje automático (modelos no supervisados, conjuntos y de aprendizaje profundo) han comenzado a utilizarse en los últimos cinco años. Esto puede haber ocurrido debido a la capacidad de estos modelos avanzados para explotar los conjuntos de datos modernos de medición ferroviaria, que pueden ser de distintos tipos según sus características: de gran volumen, multifuente, altamente desequilibrados o de alto ruido [12].

Modelos clásicos basados en datos en mantenimiento predictivo ferroviario

Existen dos grandes grupos dentro de los modelos clásicos basados en datos: modelos estadísticos y modelos clásicos de aprendizaje automático. La Tabla 1 analiza las ventajas y desventajas de los modelos clásicos basados en datos y cómo se emplean en el mantenimiento predictivo de vías férreas.

Modelos clásicos	Tipos	Ventajas	Desventajas	Ejemplos de aplicación
Modelos clásicos estadísticos	Modelos de regresión	Simples; Interpretabilidad	Requiere conocimiento profundo de los datos	[24]
	Modelos de distribución de probabilidad	Simples; Buena interpretabilidad	Basado en hipótesis específicas	[25]
	Modelo de series temporales de datos	Buena interpretabilidad	Requiere de series temporales de datos	[26]
	Métodos Bayesianos	Estable; Mejor funcionamiento con *datasets* pequeños	Suposición de distribución previa requerida	[27]
	Procesos estocásticos	Mejor funcionamiento para modelos predictivos	Basado en hipótesis específicas	[28]
Modelos clásicos de Machine Learning	ANN	Robusto; No requiere de conocimiento experto	Pobre interpretabilidad	[29]
	SVN	Eficiente en conjuntos de datos pequeños	Pobre interpretabilidad	[30]
	Modelo basado en árboles de decisión	Buena interpretabilidad	Reajuste requerido en datos con ruido	[31]
	KNN	Simple; Buena interpretabilidad	Sensible a la distribución de los datos	[32]

Tabla 1. Modelos clásicos basados en datos

Modelos avanzados de aprendizaje automático en mantenimiento predictivo de vías férreas

Una nueva tendencia en el aprendizaje automático son las redes neuronales con un número cada vez mayor de capas y se conocen como algoritmos de aprendizaje profundo. Estos métodos rara vez requieren un procesamiento previo de los datos, ya que pueden aprender la representación directamente. Estos métodos se han aplicado en muchas aplicaciones complejas, como imagen, audio, video, lenguaje natural, análisis de sentimientos y predicción de deslizamientos de tierra [33]. También se ha demostrado que estos métodos de aprendizaje profundo son ventajosos para respaldar la toma de decisiones para la ingeniería de vías férreas. Los modelos típicos de aprendizaje profundo aplicados en este campo incluyen redes neuronales convolucionales (CNN), redes neuronales recurrentes (RNN) y modelos de memoria a corto plazo (LSTM).

El aprendizaje no supervisado tiene como objetivo encontrar patrones automáticamente a partir de datos no etiquetados. Los métodos de agrupamiento y las técnicas de reducción de dimensionalidad son los métodos no supervisados más utilizados en la ingeniería de vías férreas. La agrupación brinda información sobre las distribuciones de datos y normalmente se usa en el procesamiento de datos.

Para mejorar el rendimiento de los modelos individuales de aprendizaje automático, la combinación de dos o más modelos para construir modelos de conjunto es un enfoque utilizado por muchos investigadores [34], [35], [36]. El aprendizaje en conjunto crea un modelo avanzado al combinar las fortalezas de un conjunto de modelos base. Esto puede reducir el sesgo de las predicciones o clasificaciones finales, ya que los resultados dependen menos de un modelo en particular [34]. La agregación y el apilamiento son los dos métodos principales para combinar los modelos base.

A modo de resumen, y según la revisión de la literatura, se observa que los modelos basados en datos pueden evitar el reemplazo innecesario de los componentes de la vía, ahorrar costes y mejorar la seguridad, la disponibilidad y la eficiencia del servicio ferroviario.

Entre los métodos basados en datos, los modelos de aprendizaje automático son cada vez más populares en este campo. Los métodos de aprendizaje profundo, aprendizaje no supervisado y conjuntos están atrayendo una atención cada vez mayor. Es probable que los modelos estadísticos para el mantenimiento predictivo de vías no desaparezcan a corto plazo, principalmente debido a su capacidad para proporcionar inferencias informativas sobre las relaciones entre los parámetros y los procesos de degradación de vías. Entre las aplicaciones de los modelos basados en datos, la irregularidad de la geometría de la vía, el defecto de la cabeza de la vía y la detección de componentes faltantes de la vía son los tres problemas principales encontrados en la literatura. La predicción de rupturas de vías también ha recibido una atención creciente en los últimos cuatro años. El tipo de datos recopilados son los factores más importantes que influyen en la selección del modelo.

3 METODOLOGÍA

En este apartado se van a describir cada uno de los procedimientos, métodos, modelos y técnicas que se han empleado para poder desarrollar este proyecto de investigación y, de esta manera, lograr los objetivos establecidos previamente.

3.1 MODELO ANALÍTICO

Para llevar a cabo este proyecto de investigación se ha seguido un marco metodológico orientado a la analítica y la obtención de modelos basados en datos, es decir, orientado a la minería de datos (Data Mining), que es la ciencia en la que se enmarca este proyecto.

El modelo metodológico empleado es conocido como CRISP-DM (Cross Industry Standard Process for Data Mining) y fue concebido en 1996 por tres empresas expertas en el mercado de la minería de datos como eran NCR, Daimler Chrysler y SPSSS. Este modelo es descrito y desarrollado por Jaume Miralles Solé en su libro [37], en el cual asegura que se trata de una de las metodologías más utilizadas en el campo de la minería de datos.

La metodología CRISP-DM integra todas las etapas que componen el ciclo de vida de un proyecto de minería de datos, desde la fase de comprensión del problema hasta la puesta en funcionamiento de sistemas automatizados analíticos, predictivos y/o prospectivos.

La Figura 10 muestra el diagrama de las etapas que componen dicha metodología.

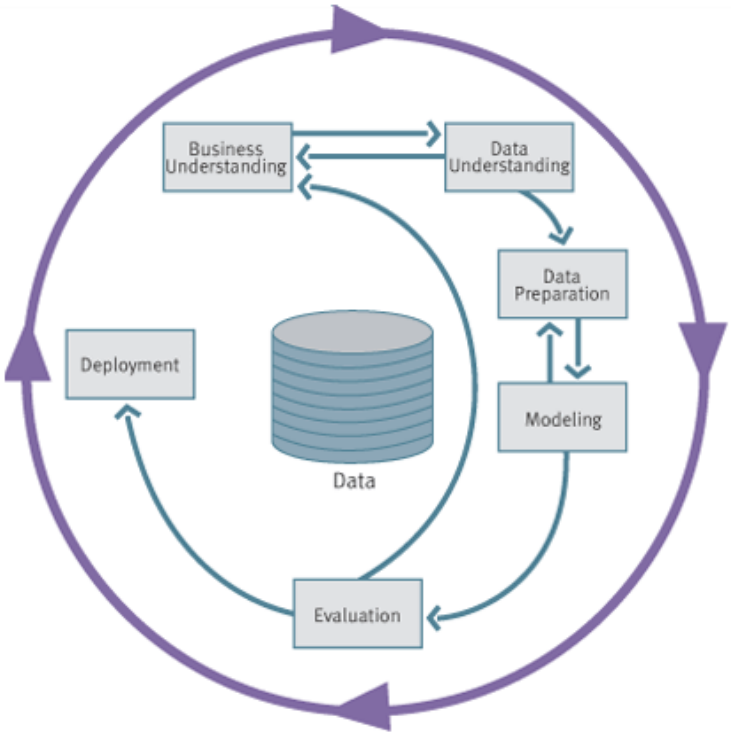


Figura 10. Diagrama de la metodología CRIPS-DM [37]

Como se puede observar en la figura, el modelo metodológico se trata de un proceso cíclico formado por seis fases. Un aspecto fundamental es que se trata de un proceso iterativo, en el cual las fases no tienen un único y restringido sentido, sino que la mayoría son bidireccionales. Esto permite que se pueda volver a una fase anterior si el resultado obtenido no cumple con los objetivos de esa fase y de esta forma se pueda alcanzar el producto deseado de una manera óptima.

A continuación, se describe cada una de las seis fases que constituyen el marco metodológico CRISP-DM.[38]

- **Fase I. Comprensión del negocio**

  Esta fase inicial comprende un estudio exhaustivo del problema que se quiere resolver, así como establecer los requisitos y objetivos del mismo. Después, ese conocimiento y esa comprensión del problema se traduce en un problema de minería de datos y en una planificación preliminar diseñada para alcanzar los objetivos.

- **Fase II. Estudio y comprensión de los datos**

  Esta fase comprende las tareas de recopilación y estructuración de los datos en tablas, donde se describen y se clasifican según su formato, atributo, calidad o cantidad entre otros aspectos. De esta forma, los datos se ordenan de manera rigurosa para facilitar su análisis en la siguiente fase del proceso.

  Esta fase y la siguiente son críticas en el desarrollo del proyecto ya que, si las tareas descritas no se llevan a cabo correctamente, pueden provocar fallos en cadena en fases posteriores del proyecto. Por ello, esta fase junto a las dos siguientes fases son las que demandan el mayor esfuerzo y tiempo en un proyecto de minería de datos.

- **Fase III. Análisis y tratamiento de los datos**

  Se trata de la etapa previa a la fase de modelado, la cual se encarga de seleccionar, limpiar, generar y ordenar conjuntos de datos correctos, preparándolos de esta manera para la fase siguiente.

  Es considerada la fase crítica de cualquier proyecto de minería de datos, ya que cualquier error en los datos que no se haya detectado y resuelto en esta fase, se trasladaría hasta la fase de modelado, provocando una reducción en la exactitud de los modelos o incluso, generando resultados erróneos.

Como ya se ha comentado previamente, esta fase se encuentra estrechamente relacionada con la fase de modelado, puesto que, cada técnica de modelado requiere de un procesado determinado de los datos, llegando hasta el punto de tener que modificar la técnica de modelado en el caso de no poder realizar el tratamiento de datos requerido. Por esta razón, las fases de preparación y de modelado interactúan de forma permanente.

Debido a la trascendencia a la que se ha hecho referencia, se estima que en esta fase se invierte el 75% del total del tiempo del proyecto.

- **Fase IV. Modelado**

  En esta fase, partiendo de los datos procesados en la fase anterior, se seleccionan y aplican las técnicas de modelado que mejor se ajustan al problema que se precisa resolver.

  El primer paso dentro de esta fase consiste en escoger los algoritmos de modelado más adecuados al problema planteado.

  Posteriormente, se configuran los valores de los parámetros que se usarán para los algoritmos de Machine Learning. Dichos parámetros dependerán de las características de los datos y de las especificaciones del resultado que se quiera lograr con el modelo.

  Una vez configurados los parámetros, se determinan las métricas de evaluación y se implementan los algoritmos seleccionados previamente, construyendo de esta manera los modelos que consigan cumplir con los objetivos del proyecto.

  Suelen existir varias técnicas o algoritmos para un mismo tipo de problema de minería de datos. Algunas de esas técnicas requieren de características específicas respecto a la forma de los datos. Por lo tanto, casi siempre en cualquier proyecto se acaba volviendo a la fase de preparación de datos varias veces hasta llegar a una cohesión de las dos fases que permita encontrar la solución óptima.

- **Fase V. Evaluación**

  En esta fase se evalúan los resultados obtenidos en la fase de modelado mediante el análisis de las métricas de evaluación, cuya finalidad es conocer la bondad de los modelos generados y analizar el cumplimiento de los requisitos establecidos al comienzo del proyecto. Se debe tener en cuenta que, la fiabilidad calculada para el

modelo se aplica solamente para el conjunto de datos sobre los que se realizó el análisis.

Las fases que integran esta metodología no son independientes, sino que todas ellas están vinculadas entre sí con el objetivo común de alcanzar los objetivos marcados de la manera más precisa y óptima posible. Por ello, en esta etapa no sólo se evalúa la fase de modelado, sino que se realiza una profunda depuración del proceso completo, analizando las tareas desarrolladas y los resultados obtenidos en cada fase. En caso de detectar algún error o punto de mejora, se evalúa a partir de qué fase se vuelve a iterar el proceso.

Otra función importante de esta fase es determinar si hay alguna cuestión trascendental de negocio que no haya sido considerada adecuadamente o que deba tener mayor influencia dentro del proyecto.

- **Fase VI. Despliegue**

  Una vez que el modelo ha sido construido, validado y ha generado unos resultados concluyentes, se llevan a cabo acciones y se toman decisiones en base a los resultados obtenidos.

  Normalmente, un proyecto de minería de datos no concluye en la implementación del modelo, sino que se debe asegurar el mantenimiento de la aplicación, así como documentar y presentar los resultados de manera comprensible para el usuario con el objetivo de lograr un incremento del conocimiento.

  Dependiendo de los requisitos, la fase de despliegue puede ser tan simple como la generación de un informe o tan compleja como la implantación automatizada de un proceso de análisis de datos.

La Figura 11 representa de manera esquemática las seis fases, incluyendo las subtareas correspondientes de cada fase, que forman la metodología CRISP-DM.

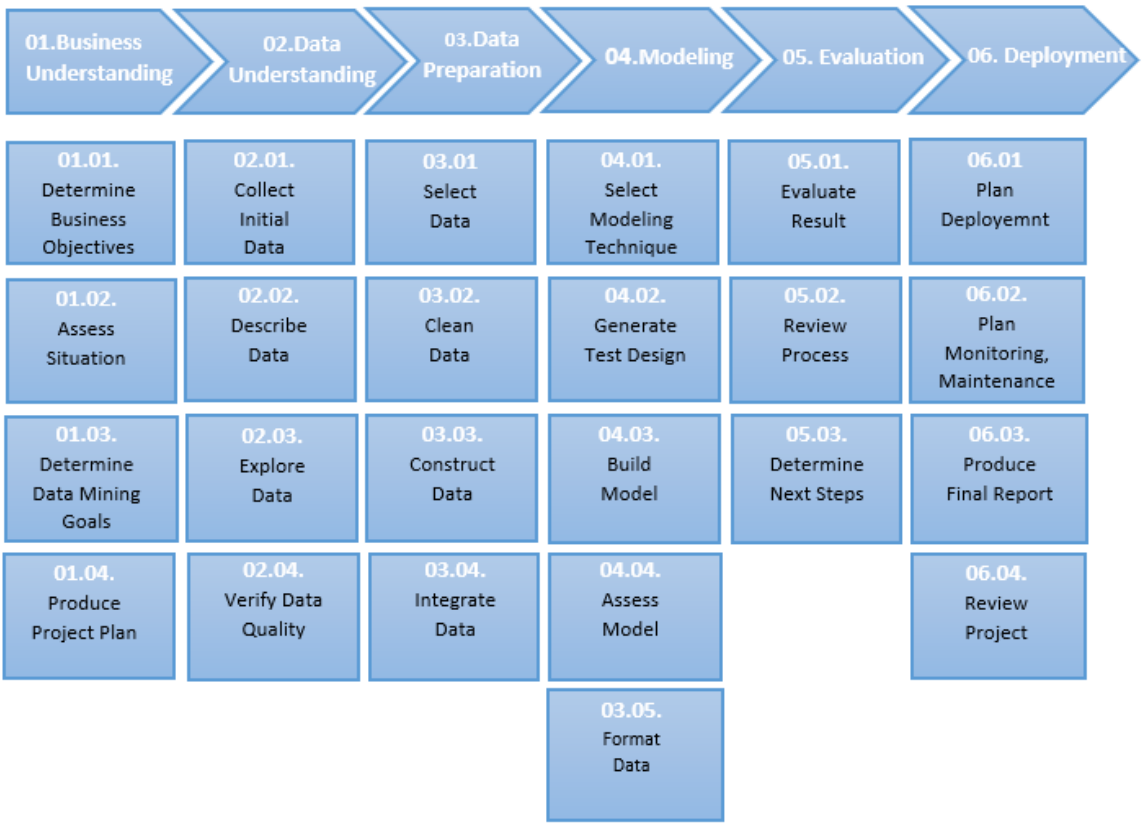


Figura 11. Diagrama de las subtareas de la metodología CRISP-DM [37]

CRISP-DM se trata de una metodología muy completa ya que cuenta con un enfoque empresarial que engloba todo el proceso, considerando la aplicación de los resultados al entorno de los negocios y estableciendo así un contexto mucho más rico que influye en la elaboración de los modelos. Este contexto está relacionado con el hecho de que el marco metodológico es cíclico y, por tanto, el proyecto no acaba una vez se halla el modelo idóneo, sino que pueden aparecer puntos de mejora tras la puesta en marcha o que esté vinculado a otros proyectos. Por ello, es preciso documentarlo de manera exhaustiva para que otros equipos de desarrollo utilicen el conocimiento adquirido y trabajen a partir de él.

Además, la metodología CRISP-DM dispone de cierta flexibilidad que permite adaptarla dependiendo de la rama dentro de la ciencia de datos en la que se implemente (analítica descriptiva, Machine Learning, deep learning, etc.).[39][40]

Esta remarcada flexibilidad ha permitido que para el desarrollo de este proyecto se implemente una adaptación reducida de esta metodología. En ella no se hace uso de las fases referidas al punto de vista empresarial, ya que no es objeto de este proyecto.

3.2 COMPRESIÓN DE LOS DATOS

En la metodología CRISP-DM, la fase denominada 'comprensión de los datos' se considera una etapa esencial dentro de cualquier proyecto que requiera de la ciencia de datos para su desarrollo, como es el caso del proyecto que se expone en este documento.

Como detalla la metodología y se puede observar en la Figura 11, el primer paso dentro de esta fase será la recopilación de los datos, los cuales son la base del proyecto puesto que, sobre ellos se construirán los modelos de Machine Learning que integrarán funciones de diagnóstico, detección y predicción de fallos, ayudando de esta manera en la toma de decisiones.

La ausencia de aplicaciones tecnológicas avanzadas, como la instrumentación electrónica o softwares de almacenamiento y procesamiento de datos, dentro del sector ferroviario, han suscitado que apenas se disponga de datos históricos sobre la infraestructura ferroviaria. Todo ello unido a la privacidad y a la complejidad técnica del tema concreto estudiado en este proyecto, han hecho imposible la recopilación de datos válidos para el desarrollo del mismo. A pesar de los esfuerzos realizados consultando diversas fuentes como empresas del sector o bases de datos online.

Debido a esta falta de disponibilidad de datos sobre el tema de estudio de este proyecto, se ha recurrido a la generación de datos sintéticos, una herramienta innovadora muy utilizada dentro del campo de la ciencia de datos y la inteligencia artificial.

La generación de datos sintéticos se trata de una técnica novedosa y puntera en el mundo de la ciencia de datos y la inteligencia artificial que consiste en originar conjuntos de datos artificiales cuando los conjuntos de datos reales carecen de calidad, volumen o variedad. Tanto los datos sintéticos generados como las herramientas utilizadas para ello pueden ser de distintos tipos y clases, su uso dependerá del caso de aplicación y su estrategia definida.

Según un artículo elaborado por "SYNTHO" [41], existen tres grandes grupos de datos sintéticos que se van a detallar a continuación:

- **Datos ficticios**: son datos generados de manera aleatoria. En consecuencia, las características, correlaciones y patrones estadísticos que se encuentran en los datos originales no se conservan ni reproducen en los datos ficticios generados. Por lo tanto, la representatividad de este tipo de datos es mínima en comparación con los datos originales. Un ejemplo de aplicación es el reemplazo de identificadores directos (PII).
- **Datos basados en reglas**: son datos generados en base a un conjunto predefinido de reglas. Estas reglas son básicamente atributos, especificaciones, correlaciones y patrones estadísticos de los datos reales, que se pretenden reproducir en los datos

sintéticos y que se deben definir previamente. Por lo tanto, la calidad de los datos estará determinada por la calidad del conjunto de reglas. Se trata de un proceso que requiere bastante tiempo y, en consecuencia, se considera poco eficiente.

- **Datos generados por inteligencia artificial**: se trata de datos sintéticos generados por un algoritmo de inteligencia artificial. El modelo de IA se entrena con los datos originales con el objetivo de que aprenda todas sus características, relaciones y patrones estadísticos. Una vez entrenado el modelo, este algoritmo de IA puede generar datos nuevos que contengan las características, las relaciones y los patrones estadísticos del conjunto de datos original. Esto es lo que se conoce como un gemelo de datos sintéticos.

Debido a la falta de datos reales, y teniendo en cuenta que se trata de un tipo de datos sintéticos muy empleado en proyectos de análisis de datos y Machine Learning [42], se ha considerado los datos basados en reglas como el tipo de datos sintéticos que mejor se ajusta a la metodología de desarrollo de este proyecto. Por ello, en la aplicación del caso de estudio se generarán datos sintéticos basados en reglas.

Una vez determinado el tipo de datos sintéticos que se van a utilizar, será conveniente seleccionar las herramientas necesarias para generar este tipo de datos. Existen softwares específicos para la generación de datos sintéticos, como son GPT-J, Synthea o SDV, entre otros. Sin embargo, están diseñados para temas muy concretos. Sobre este tema se ha pronunciado Steven Karan, vicepresidente y jefe de *insights* y *datos* de Capgemini Canadá: "Todavía no ha llegado al mercado una solución comercial lista para usar". “En su lugar, la mayoría de los científicos de datos aprovechan los paquetes preconstruidos para generar conjuntos de datos sintéticos”, afirma.[43]

Existen numerosas herramientas disponibles para crear datos sintéticos. La mayoría se tratan de librerías de código abierto en entornos de programación matemáticos, estadísticos o enfocados al análisis de datos. Algunos ejemplos destacables de estos entornos son: R, Python, SAS y MATLAB, así como herramientas de minería de datos como H2O, Apache Spark y Apache Flink.

A continuación, se presentan algunas de las herramientas más populares para crear datos sintéticos:

- **Synthetic Data Vault (SDV)**: conjunto de librerías implementables en Python que permiten generar tablas, bases de datos relacionales y modelos de series temporales.
- **GPT-J**: algoritmo de inteligencia artificial de código abierto cuya función principal es el procesamiento del lenguaje natural y la generación de texto.

- **Synthea**: herramienta de código abierto popular en el campo de la medicina que genera perfiles de pacientes sintéticos.
- **Synth**: es un software de código abierto que genera series en tiempo real y datos relacionales con el objetivo de entrenar modelos de aprendizaje automático, ocultar información de identificación personal y desarrollar datos de prueba para determinadas aplicaciones.
- **Synthpop**: paquete de librerías de R diseñado para generar datos demográficos sintéticos.
- **Numpy**: es una librería de Python especializada en el cálculo numérico y el análisis de datos para un gran volumen de datos. La ventaja de esta librería es que el procesamiento de los *arrays* se realiza mucho más rápido (hasta 50 veces más) que las listas predeterminadas de Python, lo cual la hace ideal para el procesamiento de vectores y matrices de grandes dimensiones.
- **Scikit-learn**: se trata de otra librería de Python que permite generar conjuntos de datos sintéticos para su uso en regresión, agrupación y clasificación, con el objetivo de producir conjuntos de datos que puedan permitir predicciones.
- **Pandas**: es una librería de código abierto de Python especializada en el procesamiento y el análisis de datos. Permite generar estructuras de datos y realizar operaciones para manipular tablas numéricas y series temporales.
- **SymPy**: biblioteca de Python que simula ser un sistema de álgebra por computadora (CAS) de código abierto. Es empleada por los científicos de datos que requieren de conjuntos de datos sintéticos más personalizados para cumplir objetivos más específicos, ya que permite la creación y el desarrollo de expresiones matemáticas simbólicas personalizadas
- **Pydbgen**: se utiliza para generar conjuntos de datos personales, como números de teléfono o direcciones de correo electrónico.
- **Faker**: paquete de librerías de Python capaz de generar datos sintéticos como nombres, direcciones, correos electrónicos, números de la Seguridad Social y otros datos personales.

Debido a las ventajas que ofrece el entorno de Python para manipular y analizar datos, éste va a ser el software que se utilice para desarrollar el proyecto que se expone en este documento. Por lo tanto, se va a recurrir a algunas librerías de código abierto de Python descritas en este apartado, como son Pandas, Scikit-Learn o Numpy, entre otras, con la finalidad de generar y operar con datos sintéticos.

Los datos sintéticos son una herramienta innovadora y versátil capaz de proporcionar soluciones eficientes y seguras en diversas áreas y escenarios. Se generan en múltiples campos de aplicación para una gran variedad de propósitos, como pruebas de algoritmos, generación de conjuntos de datos de entrenamiento, protección de la privacidad e investigación científica, entre otros.

Figura 12. Esquema de campos de aplicación de datos sintéticos

A continuación, se describen algunos campos de aplicación que ejemplifican la versatilidad y el potencial de los datos sintéticos para aportar soluciones eficientes e innovadoras en diversos ámbitos:

- **Entrenamiento de modelos de inteligencia artificial**: los datos sintéticos se emplean para entrenar algoritmos de Machine Learning cuando los datos reales son escasos o difíciles de obtener. Esto permite a sistemas de IA o análisis de datos desarrollar, afinar e incluso acelerar el aprendizaje de modelos sin necesidad de depender de los datos reales.[44][45][46]
- **Pruebas de software**: los datos sintéticos se utilizan para probar aplicaciones y softwares sin necesidad de utilizar datos reales sensibles. Esto garantiza la privacidad y la seguridad de los datos mientras se realizan pruebas exhaustivas.[47][48]
- **Investigación médica**: los datos sintéticos son especialmente útiles en campos como la investigación médica, donde los datos sintéticos permiten a los investigadores crear escenarios específicos y controlados para probar hipótesis o evaluar el rendimiento de nuevas tecnologías o enfoques sin los riesgos o limitaciones asociados a la divulgación de información personal.[49][50][51]

- **Investigación biológica**: dentro del campo de la biología, los datos sintéticos son útiles para generar secuencias de ADN o proteínas sintéticas utilizando algoritmos basados en las probabilidades de aparición de diferentes bases o aminoácidos. También son empleados en la simulación de fenómenos biológicos, como la dinámica de poblaciones o reacciones bioquímicas.[52][53][54]
- **Investigación química**: los datos sintéticos también son muy utilizados, mediante simulaciones computacionales, para entrenar modelos de aprendizaje automático que puedan predecir propiedades y comportamientos de compuestos químicos, acelerando de esta manera el proceso de investigación y diseño de nuevos materiales y compuestos químicos, así como el descubrimiento de nuevos fármacos.[55]
- **Robótica**: la generación de datos sintéticos puede ayudar a mejorar la precisión de la detección de objetos en aplicaciones robóticas, proporcionando conjuntos de datos sintéticos con variaciones controladas para entrenar y mejorar algoritmos de detección.[56][57][58]
- **Simulación y procesos automatizados**: los datos sintéticos también se aplican en simulaciones y modelos para representar situaciones y escenarios reales o remotos. De esta manera ayudan a evaluar los efectos de una determinada acción sin tener que realizar la acción real, y permiten ajustar los modelos para que estén preparados ante todo tipo de situaciones hipotéticas o improbables. [59][60][61]
- **Educación y formación**: los datos sintéticos se utilizan en entornos educativos para enseñar conceptos y técnicas relacionadas con la manipulación y el análisis de datos sin necesidad de acceder a conjuntos de datos reales. También se generan datos sintéticos para simular diferentes escenarios y condiciones de aprendizaje, de esta manera los investigadores pueden analizar y evaluar el impacto de diferentes variables en el proceso educativo.[62][63]
- **Economía**: los datos sintéticos pueden utilizarse para simular modelos económicos, permitiendo a los investigadores evaluar diferentes escenarios, indicadores y políticas económicas sin necesidad de utilizar datos reales, que pueden ser costosos o difíciles de obtener.[64][65][66]
- **Marketing**: en el mundo del marketing, los datos sintéticos se utilizan para crear perfiles de audiencia que representan características y comportamientos de grupos de consumidores. Al generar diferentes segmentos de audiencia, se pueden realizar pruebas y experimentos en entornos controlados para evaluar el impacto de diferentes estrategias de marketing, mensajes publicitarios o cambios en los productos y servicios.[67][68][69]
- **Metaverso**: al tratarse de un entorno virtual, el metaverso requiere de todo tipo de creaciones y simulaciones virtuales. En este sentido, los datos sintéticos ayudan a generar avatares personalizados, paisajes, edificios y multitud de objetos que integran el entorno virtual.

También permiten la simulación de comportamientos, interacciones sociales y otros escenarios de interacción que contribuyan a la experiencia inmersiva del metaverso.[70][71][72]

Estas son solo algunas de las aplicaciones de los datos sintéticos, y como se puede observar, su uso puede variar dependiendo del contexto y la industria específica. Debido a las dificultades anteriormente comentadas para disponer de datos reales, el presente proyecto se enmarca en el campo de aplicación donde los datos sintéticos se emplean para entrenar algoritmos de Machine Learning cuando los datos reales son escasos o difíciles de obtener.

La generación y empleo de datos sintéticos se encuentra en constante evolución a medida que se desarrollan nuevas tecnologías y se descubren nuevas herramientas que permiten aprovechar su potencial. En consecuencia, su rango de aplicación se está extendiendo a cada vez más ámbitos.

La empresa Gartner, compañía dedicada a la consultoría y la investigación de las tecnologías de la información, ha realizado un estudio predictivo de la evolución de los datos sintéticos. Estiman que en el año 2030 la aplicación de datos sintéticos superará la de los datos reales.[73]

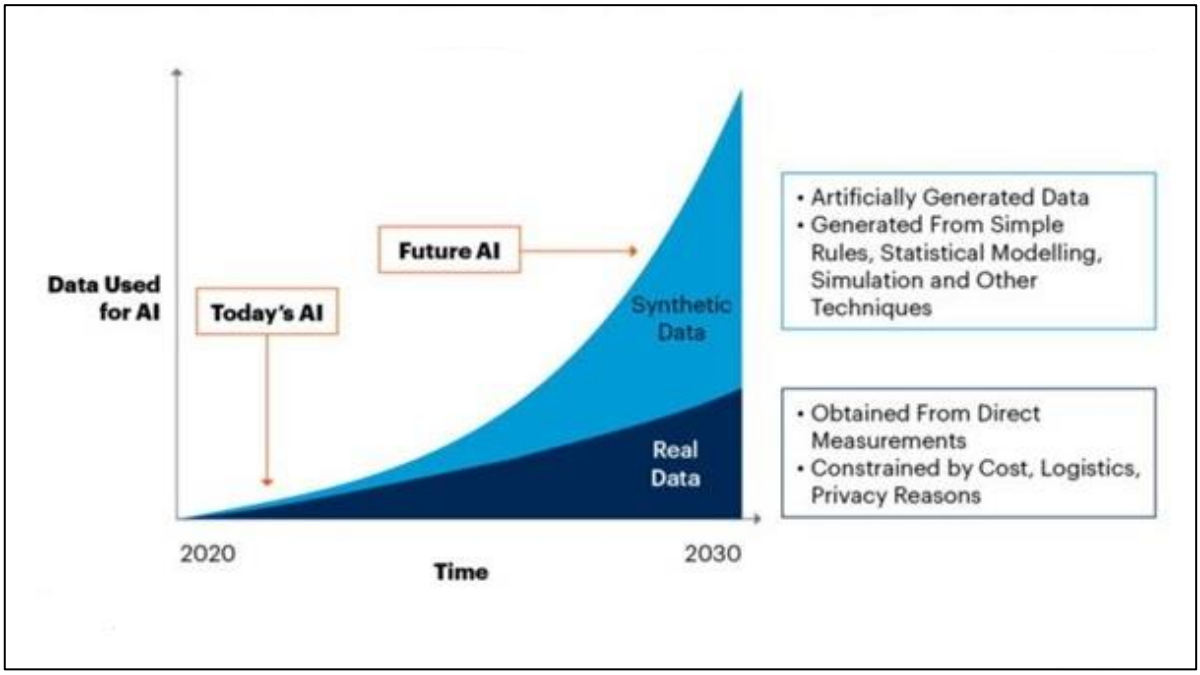


Figura 13. Estimación del uso de datos sintéticos en aplicaciones AI hasta el año 2030

Es importante destacar que, si bien los datos sintéticos pueden ser útiles en determinados contextos, no pueden reemplazar por completo la necesidad de trabajar con datos reales en cualquier tipo de

proyecto relacionado con análisis de datos. Los datos reales son fundamentales para validar los hallazgos y asegurar la aplicabilidad y relevancia de los resultados obtenidos.

3.3 MODELADO DE DATOS: PCA

Esta fase de la metodología consiste en identificar patrones o correlaciones presentes en las variables que miden las condiciones de los aparatos de vía, permitiendo de este modo diferenciar de manera precisa y clara el estado no dañado de cualquier situación de daño que pueda experimentar la estructura ferroviaria. Su objetivo es proporcionar una distinción clara y nítida entre estos dos estados, brindando así una detección efectiva de posibles daños en los aparatos de vía.

Vincular estas relaciones ocultas entre los datos con los dos posibles estados de salud estructural mencionados, es difícil de lograr mediante procedimientos de análisis basados en la física y el comportamiento estructural, debido a las numerosas incertidumbres derivadas del diseño, las condiciones ambientales, la construcción y la explotación. Por lo tanto, es común recurrir a la implementación de modelos estadísticos que permitan detectar o predecir la presencia de daños utilizando técnicas de Machine Learning como el análisis de componentes principales o PCA.

Un aspecto a destacar es que, según la metodología implantada en este proyecto y detallada en el apartado 3.1, para poder crear un modelo estadístico robusto, es necesario realizar previamente una comprensión y un análisis de los datos, y establecer las variables que definen el problema, así como las dimensiones de los *datasets* que se van a utilizar para implementar el modelo estadístico. También es preciso recordar que, en el caso de que el resultado obtenido por el modelo implementado no fuera el deseado, este proceso metodológico permite retroceder a la fase anterior para modificar la caracterización del conjunto de datos, calibrando de manera eficiente el modelo estadístico desarrollado.

El Análisis de Componentes Principales, es una técnica de Machine Learning no supervisado ampliamente utilizada en diversas áreas, como análisis de datos, reconocimiento de patrones, visión por computadora y bioinformática, entre otras. Su objetivo principal es analizar y reducir la dimensionalidad de conjuntos de datos multivariados. Para ello, trata de encontrar un nuevo conjunto reducido de variables, llamadas componentes principales, que capturen la mayor parte de la variabilidad presente en los datos originales.

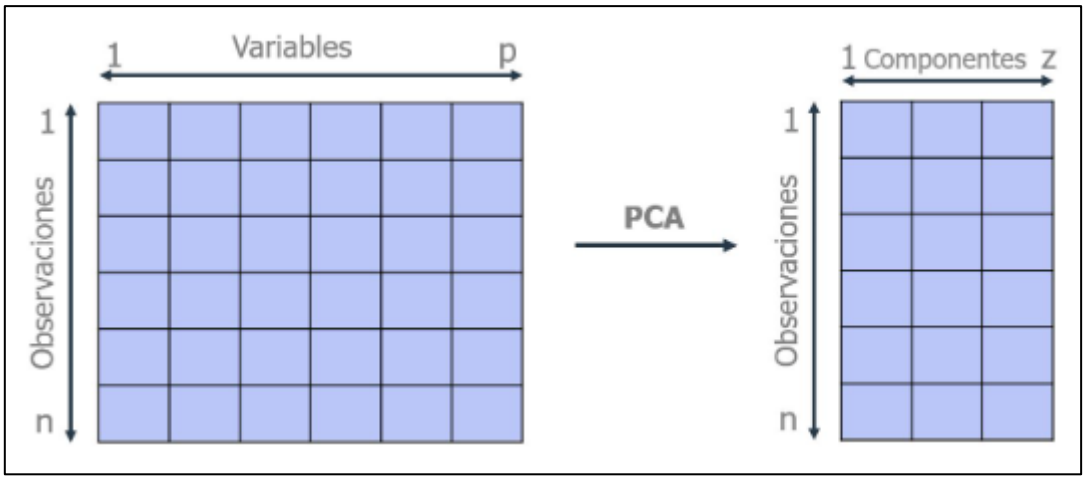


Figura 14. Esquema gráfico del Análisis de Componentes Principales

El PCA se basa en la idea de que los datos multidimensionales a menudo contienen redundancia o correlación entre las variables. Busca transformar los datos originales en un nuevo sistema de coordenadas en el que las variables deben cumplir la condición de no correlación entre sí.

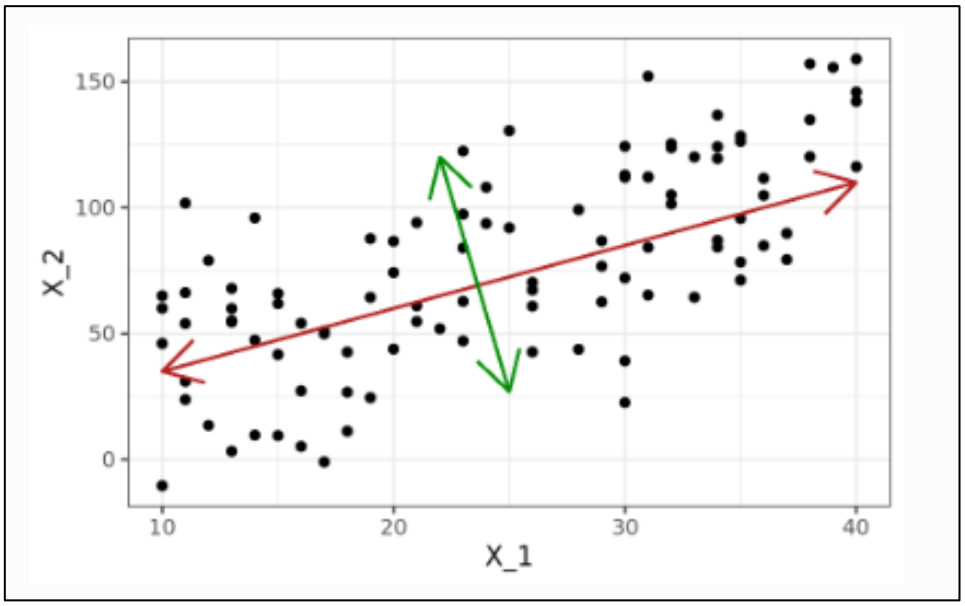


Figura 15. Ejemplo de nuevo sistema de coordenadas de componentes principales

Los componentes principales se obtienen como combinaciones lineales de las variables originales, donde la primera componente principal captura la mayor varianza posible en los datos, la segunda componente principal captura la siguiente mayor varianza, y así sucesivamente.

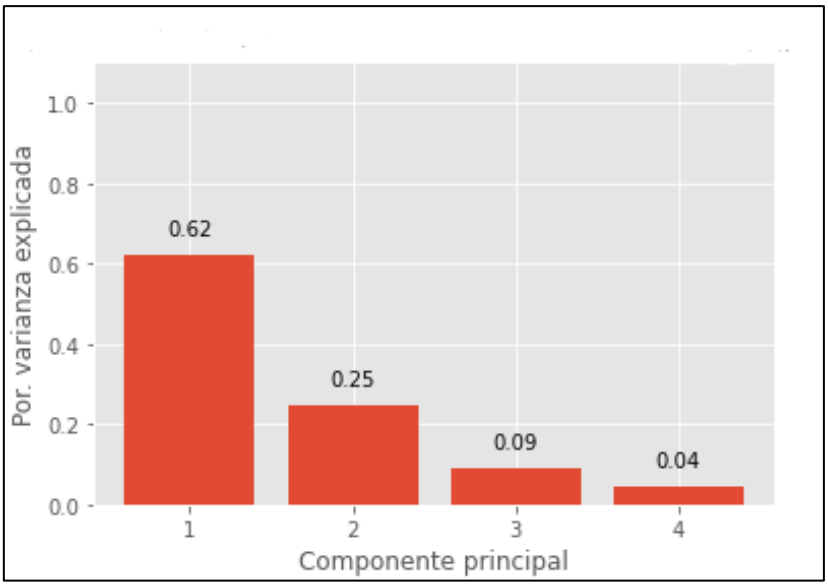


Figura 16. Ejemplo de porcentaje de varianza explicada por cada componente principal

El proceso del análisis de componentes principales consta de los siguientes pasos:

1. **Estandarización de datos**: se aplica una transformación a los datos para que todas las variables tengan media cero y desviación estándar uno. La transformación matemática que se realiza es la siguiente:

$$\mu_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij} \qquad (1)$$

$$\sigma_j{}^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(x_{ij} - \mu_j\right)^2 \qquad (2)$$

$$\bar{x}_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j} \qquad (3)$$

   Siendo μ_j y $\sigma_j{}^2$ la media y la varianza respectivamente de la variable j, mientras que $\bar{x}_{ij}$ son cada uno de los elementos de la nueva matriz de datos normalizada. Este paso es importante para evitar que las variables con escalas grandes dominen el análisis.

2. **Cálculo de la matriz de covarianza**: se calcula la matriz de covarianza o correlación de los datos estandarizados, la cual muestra las relaciones de covarianza o correlación entre todas las variables. Para obtener la matriz de covarianza se realiza la siguiente transformación lineal:

$$T = X * P \quad (4)$$

Donde *X* es la matriz de datos *n*x*m* (*n* muestras y *m* variables) estandarizada y *P* una matriz de transformación que contiene los autovectores de la matriz de covarianza. Con el objetivo de minimizar la redundancia entre variables, se busca una matriz de transformación tal que la covarianza de la nueva matriz transformada *T* sea diagonal. Esto implica que:

$$C_T = \frac{1}{n-1} T^T * T \quad (5)$$

Sustituyendo (4) en (5) se obtiene:

$$C_T = \frac{1}{n-1} P^T * X^T * X * P = P^T * C_x * P \quad (6)$$

La matriz C_x es la matriz de covarianza (*m*x*m*) y tiene la característica de que es simétrica. Al ser simétrica tendrá *m* autovalores y *m* autovectores que forman la base del nuevo espacio de *m* dimensiones.

3. **Cálculo de los componentes principales**: se calculan los vectores propios (también llamados autovectores) y los valores propios (también llamados autovalores) de la matriz de covarianza. De (4) y (6) se sabe que la matriz *P* (*m*x*m*) contiene los autovectores propios de C_x y de la ecuación (7) se pueden calcular la matriz de autovalores Λ que es una matriz diagonal.

$$C_x * P = P * \Lambda \quad (7)$$

Los vectores propios representan las direcciones de máxima variabilidad en los datos, mientras que los valores propios indican la cantidad de varianza capturada por cada componente principal. Los autovectores son ortogonales entre sí, lo que significa que no están correlacionados y son mutuamente perpendiculares.

4. **Selección de componentes principales**: se selecciona un número determinado de componentes principales para retener, generalmente basado en la varianza explicada acumulada. Se eligen aquellos componentes principales que capturan la mayor parte de la varianza total de los datos. De esta manera la nueva matriz de autovectores P' (*m*x*q*) estaría compuesta por *q* vectores propios, siendo *q* las componentes principales seleccionadas. Y lo mismo ocurriría para la nueva matriz de valores propios Λ' (*m*x*q*).

Si se desea reducir la dimensionalidad, se selecciona un número menor de componentes principales. Por el contrario, si se busca preservar la mayor cantidad de información posible, se retienen todos los componentes principales. Se trata de encontrar un equilibrio a la hora de reducir la dimensión del problema, tratando de perder la menor cantidad posible de información.

5. **Transformación de datos**: se proyectan los datos originales en el nuevo espacio de componentes principales. Para hacerlo, se multiplica la matriz de datos estandarizados por la matriz de vectores propios correspondientes a los componentes principales seleccionados. El resultado de esta proyección son los datos transformados en el nuevo sistema de coordenadas.

$$T' = X * P' \quad (8)$$

El campo de aplicación del análisis de componentes principales es muy amplio y diverso, incluyendo la compresión de datos, la exploración visual de datos, la detección de outliers y la clasificación de muestras, entre otras aplicaciones.

En concreto, se trata de una herramienta muy útil para la exploración visual de datos, ya que los componentes principales pueden representarse gráficamente en un espacio de menor dimensión, facilitando de esta manera la identificación de patrones, grupos o relaciones entre variables en los datos.

Esa funcionalidad en el análisis de datos y el reconocimiento de patrones y relaciones en grandes conjuntos de datos es el motivo de que se haya recurrido a esta técnica estadística para implementar un algoritmo que integre funciones de diagnóstico, detección y predicción de fallos en aparatos de vía basado en Machine Learning.

3.4 IMPLEMENTACIÓN PCA EN PYTHON

Desde hace años, Python se encuentra entre los lenguajes de programación más empleados en el desarrollo de software. Se trata de un lenguaje de alto nivel orientado a objetos con un código conciso, una sintaxis simple y una depuración fácil y sencilla, que hacen que la programación sea rápida y eficiente. Este es el motivo por el cual se ha elegido como lenguaje de programación para desarrollar este proyecto.

Sin embargo, como ocurre con la mayoría de los lenguajes de programación, para simplificar, depurar y maximizar la eficiencia de la escritura de código, es recomendable utilizar un Entorno de Desarrollo Integrado (IDE).

Pycharm es un IDE desarrollado por la compañía JetBrains que destaca por ser uno de los entornos para Python más utilizados a nivel global. Grandes empresas como Twitter, Facebook, Amazon o Pinterest lo emplean habitualmente para el desarrollo de aplicaciones y software en Python.

Su compatibilidad con una amplia gama de bibliotecas especializadas en el procesamiento y análisis de datos, así como su enfoque en la eficiencia y la productividad, hacen que Pycharm sea uno de los entornos de programación más utilizados para desarrollar proyectos de data science y Machine Learning.[74]

A continuación, se enumeran las funciones más importantes que hacen que la implementación de modelos de Machine Learning o data science en este entorno sea intuitivo y eficiente.

1. **Editor de código robusto**: PyCharm ofrece un editor completo con verificación de sintaxis, verificación de correspondencia de llaves y comillas, autocompletado, refactoring (realizar cambios rápidos y eficaces en las variables locales o globales) y otras funcionalidades que proporcionan mayor eficiencia en la escritura y edición de código Python.
2. **Depuración avanzada**: PyCharm dispone de una potente herramienta de depuración que permite establecer puntos de interrupción, inspeccionar variables, ejecutar el código paso a paso y analizar el flujo de ejecución. Esto es especialmente útil cuando se trabaja en proyectos de Machine Learning para detectar errores en el modelo o en los datos.
3. **Gestión de entornos virtuales**: PyCharm facilita la creación y gestión de entornos virtuales de Python. Esto es útil para mantener tus dependencias y librerías organizadas, lo cual es especialmente importante en proyectos de data science donde se utilizan muchas bibliotecas diferentes.

4. **Integración con bibliotecas y frameworks populares**: PyCharm está integrado por numerosas bibliotecas y frameworks empleados en data science y Machine Learning, como NumPy, Pandas, Matplotlib, SciPy, TensorFlow, PyTorch o scikit-learn, entre otros. Esto facilita la importación de estas bibliotecas a la vez que brinda acceso a herramientas y funcionalidades específicas para estas áreas.
5. **Integración con Jupyter Notebook**: Jupyter Notebook es una herramienta popular en el ámbito de data science. Pycharm permite crear, abrir y ejecutar Notebooks dentro del propio entorno, lo que proporciona una forma interactiva y visualmente atractiva de trabajar con datos y modelos.
6. **Integración con sistemas de control de versiones**: PyCharm se integra con sistemas de control de versiones populares, como Git, lo que facilita el seguimiento de los cambios en el código, la colaboración con otros desarrolladores y el trabajo en equipo.

La enorme funcionalidad unida a la compatibilidad con multitud de módulos de análisis de datos han sido los motivos por los que se ha elegido Pycharm como IDE para desarrollar este proyecto.

A continuación, se describen las bibliotecas integradas en Pycharm que son especialmente útiles para realizar todo tipo de tareas dentro del campo del análisis de datos, como el preprocesamiento y la manipulación de grandes volúmenes de datos, la implementación de modelos estadísticos como el PCA o la visualización de resultados.[75][76]

- **Numpy**: es una biblioteca fundamental en Python para el cálculo numérico y la manipulación de matrices y *arrays* multidimensionales. Suministra una amplia gama de funciones matemáticas (incluyendo funciones de álgebra lineal) y herramientas que permiten el procesamiento eficiente de grandes conjuntos de datos. Además, incluye un módulo llamado "random" que proporciona funciones para generar números aleatorios. Esto es especialmente útil en simulaciones y validación de software.
- **Pandas:** es una biblioteca construida sobre “numpy” diseñada para agregar funcionalidades adicionales que facilitan la manipulación y el análisis de datos tabulares. Permite estructurar grandes conjuntos de datos de diferentes tipos en una tabla de base de datos conocida como *Data Frame*, esta estructura maximiza la eficiencia del almacenamiento, la manipulación y el análisis de grandes volúmenes de datos. También ofrece una amplia gama de funciones y métodos diseñados para leer y escribir datos en una amplia variedad de formatos, manipular, transformar y limpiar datos, o para realizar cálculos estadísticos básicos, entre otras funciones. Además, “pandas” se integra con otras bibliotecas populares de visualización de datos, como “matplotlib” y “seaborn”, lo que facilita la creación de gráficos y visualizaciones atractivas directamente desde los *Data Frames*.

- **Sklearn**: es una biblioteca empleada popularmente para realizar tareas relacionadas con la minería de datos y el análisis predictivo. Proporciona una amplia gama de algoritmos y herramientas para la manipulación y estandarización de datos, la detección de patrones, la selección de características, la construcción de modelos y la evaluación de rendimiento. Además, incluye una amplia variedad de algoritmos de aprendizaje automático, como clasificación, regresión, agrupamiento, reducción de dimensionalidad o selección de características, entre otros. Para implementar el modelo PCA desarrollado en el presente trabajo, se recurrirá a la función "PCA" que se encuentra integrada dentro de esta biblioteca. Estos algoritmos están implementados de manera eficiente y optimizados para poder utilizarlos con grandes conjuntos de datos.
- **Matplotlib**: es una biblioteca de visualización de datos muy popular en el lenguaje de programación Python. Proporciona una amplia gama de funciones para crear gráficos estáticos, gráficos interactivos, gráficos en 3D, diagramas de dispersión, diagramas de contorno, histogramas y muchos otros tipos de visualizaciones.
- **Seaborn:** es otra biblioteca de visualización de datos para Python basada en "matplotlib". Proporciona una interfaz de alto nivel para crear gráficos estadísticos atractivos e informativos. Además, está diseñada para trabajar en conjunto con la biblioteca de análisis de datos "pandas", generando elementos gráficos a partir de los *Data Frames* creados con "pandas".

Todas estas bibliotecas descritas son compatibles y se integran perfectamente entre ellas, lo que facilita su utilización y, por consiguiente, la implementación de modelos de análisis de datos, como el análisis de componentes principales.[77]

A continuación, se muestra en la Figura 17 un ejemplo de aplicación de PCA en Python empleando las bibliotecas descritas.

```python
# LIBRERÍAS
import numpy as np
import pandas as pd
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
import seaborn as sns

# DESCARGAR DATA SET
from sklearn.datasets import load_breast_cancer
data=load_breast_cancer()
datos=data.data

# PCA
# Entrenamiento modelo PCA con escalado de los datos
estandar = pd.DataFrame(StandardScaler().fit_transform(datos))
modelo_pca=PCA()
prueba=modelo_pca.fit_transform(estandar)

#Cálculo de proporción de varianzas
varianzas=modelo_pca.explained_variance_ratio_
auxiliar = 0
VarianzasAcum=[]
for i in range(30):
    VarianzasAcum.append([])
    VarianzasAcum[i] = round((auxiliar + varianzas[i]),4)
    auxiliar = VarianzasAcum[i]
    varianzas[i] = round(varianzas[i], 4)

# VISUALIZACIÓN
ejex = []
ejex = np.arange(1, 31)
fig, (ax1, ax2) = plt.subplots(1, 2)
ax1.bar(ejex, varianzas,color='skyblue', edgecolor='black')
ax1.set_title('Diagrama de barras de Varianzas')
ax1.set(xlabel='Componentes Principales',ylabel='Varianzas')
ax2.bar(ejex, VarianzasAcum,color='skyblue', edgecolor='black')
ax2.set_title('Diagrama de barras de Varianzas Acumuladas')
ax2.set(xlabel='Componentes Principales',ylabel='Varianzas Acumuladas')
plt.show()
```

Figura 17. Ejemplo de implementación de PCA en Python

El resultado generado por el código de ejemplo de la Figura 17 es el que se muestra en la Figura 18.

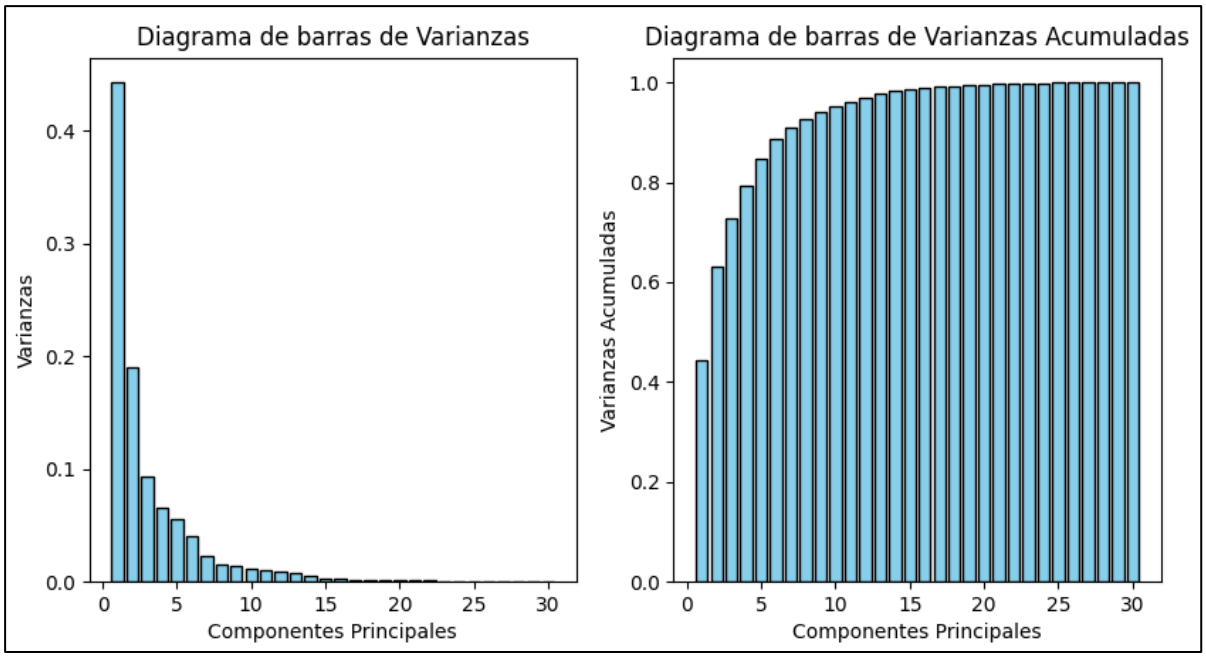


Figura 18. Resultado de la implementación de PCA en Python

4 APLICACIÓN AL CASO DE ESTUDIO

4.1 ANÁLISIS DE LOS DATOS INICIALES

Este proyecto de investigación busca desarrollar una herramienta sencilla que integre funciones de diagnóstico, detección y predicción de fallos en aparatos de vía basada en Machine Learning. Para ello, como se describió en el apartado 3.3, se recurrirá a modelos estadísticos basados en datos que sean capaces de reconocer patrones ocultos entre las variables que definen el sistema.

Según la metodología CRIPS-DM descrita en el apartado 3.1, el primer paso a la hora de desarrollar este tipo de proyectos es realizar un análisis exhaustivo de los datos disponibles para determinar su calidad y validez. Esta fase comprende las tareas de recopilación y estructuración de los datos en tablas, donde se describen y se clasifican según su formato, atributo, calidad o cantidad entre otros aspectos. De esta forma, los datos se ordenan de manera rigurosa para facilitar su posterior análisis.

Este proyecto se ha desarrollado para el estudio concreto de los aparatos de vía de tipo A. Estos dispositivos son examinados a través de inspecciones visuales de las vías y mediante el uso de equipos de medición especializados para detectar posibles irregularidades en los rieles, como desgaste o deformaciones. Toda la información recopilada durante el reconocimiento visual y las mediciones de las variables que determinan el estado estructural del aparato de vía se registra en una hoja de control idéntica a la de la Figura 28 contenida en el ANEXO I

De dicho formulario de control se puede extraer una primera lista con todas los parámetros o variables que describen el estado de salud del aparato de vía. Esta primera lista se ha estructurado en formato tabla y contiene las cuarenta y una variables extraídas de la hoja de control junto con una breve descripción de lo que mide cada una de ellas, un umbral de funcionamiento y valores teóricos. Esta tabla se puede encontrar como Tabla 2 en el ANEXO I.

Es importante destacar que el rango de funcionamiento contenido en la descripción de cada variable será fundamental en la generación de datos sintéticos, con el fin de implementar valores que representen un estado saludable del aparato de vía y con los que poder entrenar el modelo.

El objetivo de esta fase es definir un vector de datos que recoja los valores de cada variable, de tal forma que cada vector de datos represente una inspección realizada, generando de esta manera conjuntos de datos o *datasets* con los que trabajar en la fase de modelado. Pero para ello será necesario estudiar la dimensión óptima de dicho vector, es decir, será necesario realizar un análisis de las variables y filtrar aquellas que sean imprescindibles y que aporten la máxima información. Buscando de esta manera definir el sistema a modelar de la manera más simplificada posible para que el modelo implementado sea óptimo y proporcione resultados con una elevada precisión.

A continuación, se elaboró un estudio de cada variable, analizando las características, la funcionalidad y la información aportada por cada una de ellas. Para llevar a cabo este estudio, se recurrió a Instrucciones Técnicas de ADIF en las que se describe las características de cada uno de los parámetros que miden el estado del aparato de vía, así como la metodología seguida para la inspección de cada uno de ellos.[6][78]

Este estudio reveló la existencia de dos relaciones entre seis parámetros que se detallan a continuación:

Ecuación 1

$$Paso\ libre\ de\ rueda\ en\ el\ cambio = Ancho\ de\ vía\ (paso\ libre\ de\ rueda) - Entrecalle\ mínima$$

Ecuación 2

$$Ancho\ de\ vía\ (cota\ de\ protección) = Cota\ de\ protección + Entrecalle\ carril - contracarril$$

Cabe destacar que estas dos relaciones se cumplen tanto para los parámetros de vía directa como los de vía desviada, por lo tanto, serían en total cuatro relaciones entre doce variables distintas. También es importante recordar que los anchos de vía mencionados son distintos ya que se miden en distintos puntos del aparato de vía, como se detalló en el apartado 1.3.

Después de realizar un profundo estudio y analizar las características y funciones de cada una de las variables que integran la Tabla 2 del ANEXO I, se realizó una primera criba de algunas variables que, según el criterio adoptado en este proyecto, no aportaban información útil para la implementación del modelo que se busca desarrollar. En concreto, se descartaron catorce variables, todas ellas eran mediciones de ancho de vía en múltiples puntos que se han considerado irrelevantes para el caso de estudio de este proyecto.

Más tarde, se añadieron a la tabla de variables dos nuevos parámetros, “paso libre de rueda en el cambio en la vía directa” y “paso libre de rueda en la vía desviada”. Estos dos parámetros aparecían en la hoja de control inicial pero que no se habían considerado debido a que no existían datos sobre ellos. Sin embargo, tras el estudio y la documentación realizada sobre los parámetros medidos en la inspección de aparatos de vía, se llegó a la conclusión de que estos dos parámetros aportaban una información ampliamente relevante sobre el estado de los desvíos de tipo A.

En consecuencia, la tabla de variables se reajustó quedando integrada finalmente por veinte y nueve variables que definían un vector de datos de veinte y nueve componentes. La tabla actualizada con veinte y nueve variables se puede encontrar como Tabla 3 en el ANEXO I.

A pesar de tratarse de un vector de unas dimensiones demasiado grandes que dificultan el procesamiento de datos y la fase de modelado, se pasó a la siguiente fase para poder empezar a programar el algoritmo y realizar las primeras pruebas con el vector de datos resultante.

4.2 GENERACIÓN DATOS SINTÉTICOS

Como se comentó en apartados anteriores, durante el desarrollo de este proyecto se han encontrado grandes dificultades a la hora de recopilar datos válidos sobre las variables que describen el estado de los aparatos de vía tipo A, las cuales están recogidas en la Tabla 2 del ANEXO I.

Por ello, se ha recurrido a la generación de datos sintéticos, una técnica muy empleada en el campo de la ciencia de datos y la inteligencia artificial y en casos como el de este proyecto, en los que no se dispone de datos suficientes. o éstos son de baja calidad.

Esta técnica permite crear conjuntos de datos sintéticos de distintos tipos, en este caso los datos sintéticos se han generado en base a un conjunto de reglas que se han definido previamente. Estas reglas son una serie de correlaciones y condiciones de contorno que hacen que los datos permanezcan dentro de unos valores que reflejen un estado saludable del aparato y de esta manera, aunque sean sintéticos, tengan un significado físico y representen la realidad.

Se ha desarrollado un programa ejemplo en Python donde se generan cien muestras de datos sintéticos para las veinte y nueve variables que componen el vector de datos de la Tabla 3. El código de este programa ejemplo se encuentra en SCRIPT 1 contenido en el ANEXO II.

La variable “nt” contenida en el SCRIPT 1, indica el número de muestras que contiene el *dataset*. El número de muestras representa el número de vectores de datos que, acumulados uno detrás de otro, forman la matriz de datos o *dataset*. A su vez, los vectores de datos que componen el *dataset* son los definidos previamente en la fase de análisis de los datos, por ello se trata de una fase crítica y donde más tiempo se invierte, ya que no seleccionar correctamente las variables que definen el sistema puede provocar graves errores de predicción en la fase de modelado.

La biblioteca “numpy” dispone de una función denominada “random.randint(x,y,n)” que permite generar valores aleatorio dentro de un rango definido por los parámetros “x” e “y”. El número de valores que genera lo determina el parámetro “n” de la función, mientras que si se omite este parámetro se generará un único valor. Esta función se ha usado durante toda la fase de modelado para generar datos sintéticos de entrenamiento.

En el ejemplo realizado en el SCRIPT 1, haciendo uso de la función “random.randint”, se han generado cien muestras sintéticas en base a los rangos de funcionamiento definidos para cada variable en la Tabla 3. De esta manera, los datos sintéticos se generarán de manera aleatoria pero siempre dentro de un umbral que representa el estado saludable de la estructura. Sin embargo, como se comentó en el apartado 4.1, algunos de estos parámetros están relacionados entre ellos. Esto implica que la generación de datos sintéticos de estas variables sea más compleja ya que es necesario incluir una serie de condiciones de contorno. Para implementar la generación de datos de estas doce variables correlacionadas, se dividen en dos grupos:

En el grupo de la “entrecalle mínima” y el “paso libre de rueda en el cambio”, en primer lugar, se genera sintéticamente los valores de los “anchos de vía” para las vías directa y desviada en el mismo punto de la estructura. Posteriormente, se implementa un bucle en el que se obliga a las variables “entrecalle mínima” y “paso libre de rueda en el cambio”, generadas sintéticamente, a cumplir la relación geométrica que las vincula con el “ancho de vía” (**Ecuación 1**) y mantenerse dentro de sus respectivos rangos de funcionamiento. (Véase Figura 19)

```
#Entrecalle mínima y Paso libre de rueda en el cambio
for j in range(nt):
    aux=np.random.randint(58,69,1)
    aux2=np.random.randint(58,69,1)
    while (matriz[2][j]-aux) > 1617 :
        aux = np.random.randint(58, 69, 1)
    matriz[8][j]=aux                #Entrecalle mínima DIREC
    matriz[9][j]=matriz[2][j]-aux #Paso libre de rueda en el cambio DIREC
    while (matriz[6][j]-aux2) > 1617 :
        aux2 = np.random.randint(58, 69, 1)
    matriz[12][j]=aux2               #Entrecalle mínima DESV
    matriz[13][j]=matriz[6][j]-aux2 #Paso libre de rueda en el cambio DESV
```

Figura 19. Generación de datos sintéticos en los parámetros “entrecalle mínima” y “paso libre de rueda en el cambio”, dentro del SCRIPT 1

En el grupo de la “cota de protección” y la “entrecalle carril-contracarril”, en primer lugar, se genera sintéticamente los valores de las “cotas de protección” para las vías directa y desviada en el mismo punto de la estructura. Posteriormente, se implementa un bucle en el que se obliga a las variables “entrecalle carril-contracarril” y “ancho de vía”, generadas sintéticamente, a cumplir la relación geométrica que las vincula con la “cota de protección” (**Ecuación 2**) y mantenerse dentro de sus respectivos rangos de funcionamiento. (Véase Figura 20)

```
#Cota de protección, Entrecalle carril-contracarril y Anchos de Vía
matriz[10]=np.random.randint(1626,1632,nt) #Cota de protección DIREC
matriz[14]=np.random.randint(1626,1632,nt) #Cota de protección DESV
for j in range(nt):
    aux=np.random.randint(38,46,1)
    aux2=np.random.randint(38,46,1)
    while ((matriz[10][j]+aux) < 1666):
        aux=np.random.randint(38,46,1)
    matriz[11][j]=aux              #Entrecalle carril-contracarril DIREC
    matriz[3][j]=matriz[10][j]+aux #Ancho de vía DIREC
    while ((matriz[14][j]+aux2)<1666):
        aux2=np.random.randint(38,46,1)
    matriz[15][j]=aux2              #Entrecalle carril-contracarril DESV
    matriz[7][j]=matriz[14][j]+aux2 #Ancho de vía DESV
```

Figura 20. Generación de datos sintéticos en los parámetros "cota de protección" y "entrecalle carril-contracarril", dentro del SCRIPT 1

Debido a una cuestión de visibilidad y eficiencia a la hora de programar, la matriz de datos sintéticos se ha programado de tal manera que las filas fueran las variables de los vectores de datos y las columnas las distintas muestras sintéticas. Sin embargo, todas las funciones, que se han empleado durante el desarrollo del proyecto con el objetivo de implementar técnicas de análisis de datos, trabajan con *data frames*. Este tipo de estructura de datos tienen el orden traspuesto al que se ha seguido para programar la matriz de datos, es decir, las filas de los *data frames* contienen las muestras y las columnas las variables. Por ello, antes de pasar la matriz de datos a formato *data frame*, se realiza la trasposición de la matriz con la función "T" de la librería "numpy".

La función "DataFrame" de la biblioteca "pandas" transforma las estructuras de tipo *array* generadas con la biblioteca "numpy" en *data frames*.

Además, dentro del SCRIPT 1 se puede observar cómo se ha programado un fragmento de código de "verificación" con el objetivo de agilizar la depuración del código y comprobar que ninguno de los datos generados sintéticamente se saliera del umbral establecido para cada parámetro.

Finalmente, este script genera un *data frame* de cien muestras sintéticas que se expone en la Figura 21. Como se puede observar, cada fila corresponde a un vector de datos que contiene un valor para cada una de las veinte y nueve variables que forman cada vector de datos.

	Var1	Var2	Var3	Var4	Var5	...	Var25	Var26	Var27	Var28	Var29
0	1666	1671	1675	1667	1675	...	1616	1611	49	45	23
1	1678	1672	1666	1671	1676	...	1601	1601	60	45	17
2	1677	1671	1673	1667	1678	...	1592	1604	50	46	19
3	1674	1677	1668	1670	1671	...	1618	1616	52	46	22
4	1677	1676	1669	1670	1668	...	1592	1595	52	42	16
..	...	...	...	...	...	...	...	...	...	...	...
95	1667	1668	1677	1669	1673	...	1616	1617	51	43	16
96	1673	1671	1669	1670	1671	...	1616	1597	62	45	18
97	1676	1669	1668	1669	1672	...	1601	1617	63	44	24
98	1674	1673	1673	1670	1670	...	1607	1610	47	47	25
99	1673	1670	1678	1671	1669	...	1592	1614	46	41	25

Figura 21. Data Frame generado por el SCRIPT 1

La Figura 21 refleja un evidente sobredimensionamiento de los vectores de datos que provoca ineficiencia operativa a la hora de programar el software. En consecuencia, se debe retroceder a la fase de análisis de los datos y definir un nuevo vector de datos que continúe describiendo el estado del aparato de vía de una manera simplificada.

4.3 ESTUDIO DE CORRELACIONES CON PCA

Después de generar los datos sintéticos en base a las condiciones particulares de cada parámetro, se avanza a la fase de modelado, en la que se va a trabajar con el método estadístico PCA que permite detectar patrones y tendencias entre variables.

El objetivo de la fase de modelado es encontrar tendencias o correlaciones, adicionales a las ya conocidas, dentro del estado no dañado de la estructura para posteriormente simular un estado dañado y comprobar si las correlaciones encontradas se pierden cuando la estructura resulta dañada. Esta estrategia fue desarrollada por Ana Fernández-Navamuel en su trabajo [11].

En primer lugar, se ha aprovechado el algoritmo programado para la generación de datos de las veinte y nueve variables y se ha implementado el PCA sobre estos parámetros para comprobar el correcto funcionamiento de esta técnica en Python. El código completo se puede ver en el SCRIPT 2.

Para implementar la técnica PCA en Python, en primer lugar, se realiza una estandarización del *data frame* que se va a analizar, empleando la función "StandardScaler" de la biblioteca "sklearn". A continuación, se ejecuta el PCA con la función "PCA", también de la misma biblioteca, y se almacenan los resultados en la variable que se le asigne a la llamada de la función. En este proyecto, los resultados obtenidos del PCA se han almacenado en una variable denominada "modelo_pca".

El PCA arroja múltiple información que va desde los autovalores y autovectores de la matriz de covarianza, calculada en el proceso para obtener las componentes principales, hasta la proporción de varianza del *dataset* que representa cada componente principal.

Durante el desarrollo de este proyecto se ha recurrido a la información aportada por los porcentajes de varianza abarcados por cada componente principal, ya que se trata de una información valiosa a la hora de detectar la existencia de variables dominantes o dependientes y de esta manera poder reconocer patrones o correlaciones.

La Figura 22 muestra las proporciones de varianza y la acumulación de éstas proyectadas por el PCA implementado en el SCRIPT 2. Se puede observar que las proporciones son todas del mismo orden y de un valor muy bajo excepto la de las últimas cuatro componentes principales, marcadas con un recuadro rojo, que son nulas. Estos resultados arrojan dos conclusiones:

1. El PCA está bien implementado ya que el hecho de haber introducido cuatro ecuaciones en la generación de datos sintéticos implica que habrá cuatro variables que serán dependientes. Este hecho se ve reflejado en los resultados generados por el PCA, ya que las variables que son dependientes no representan prácticamente ningún porcentaje de la varianza del *dataset* y, por tanto, las proporciones de varianza arrojadas por el PCA deben ser prácticamente nulas.

2. No hay ninguna variable que predomine por encima del resto ya que todas las variables representan un porcentaje de la varianza muy similar.

	PROP. VARIANZA	PROPORCIÓN_ACUM		PROP. VARIANZA	PROPORCIÓN_ACUM
PC1	0.086777	0.086777	PC16	2.786858e-02	0.842700
PC2	0.079419	0.166196	PC17	2.415822e-02	0.866858
PC3	0.075209	0.241405	PC18	2.302046e-02	0.889879
PC4	0.072928	0.314334	PC19	2.192673e-02	0.911805
PC5	0.060642	0.374976	PC20	1.859145e-02	0.930397
PC6	0.060293	0.435269	PC21	1.729877e-02	0.947696
PC7	0.055598	0.490867	PC22	1.655355e-02	0.964249
PC8	0.052607	0.543474	PC23	1.324235e-02	0.977492
PC9	0.047260	0.590734	PC24	1.235667e-02	0.989848
PC10	0.044029	0.634763	PC25	1.015182e-02	1.000000
PC11	0.041893	0.676656	PC26	1.711384e-32	1.000000
PC12	0.040190	0.716845	PC27	1.526229e-33	1.000000
PC13	0.036889	0.753734	PC28	4.770522e-34	1.000000
PC14	0.032278	0.786011	PC29	3.773577e-34	1.000000
PC15	0.028820	0.814831			

Figura 22. Proporciones de varianza arrojadas por el PCA implementado en el SCRIPT 2

Tras observar los resultados arrojados por el PCA y comprobar que no existía ninguna correlación que hiciera predominar a un conjunto de variables por encima del resto, se llegó a la conclusión de que era necesario simplificar el sistema reduciendo el número de variables, pero sin perder la representatividad del estado de la estructura, con el objetivo de optimizar el procesamiento y análisis de datos.

En consecuencia, se redujo la dimensión del vector de datos a doce parámetros que se consideraron que aportaban más información y representatividad acerca del estado de la estructura. Concretamente, se trata de las doce variables relacionadas entre ellas a través de cuatro ecuaciones geométricas que se detallaron en el apartado 4.1. Al igual que se hizo con los anteriores vectores de datos, este vector de doce variables se encuentra descrito en la Tabla 4.

Una vez definido el nuevo vector de datos, se siguió el mismo proceso que se llevó a cabo con el vector de datos de veinte y nueve variables. Se implementó un algoritmo que en primer lugar generaba un *dataset* de la misma cantidad de muestras sintéticas, pero en este caso para doce variables y a continuación se aplicaba el PCA con el objetivo de encontrar algún patrón oculto entre los parámetros. Este nuevo algoritmo está contenido en el SCRIPT 3.

Aprovechando la reducción notable del dimensionamiento de los datos, se programó el algoritmo para que los resultados arrojados por el PCA se proyectaran en un diagrama de barras, con el objetivo de mejorar la visualización y optimizar el análisis de datos.

La biblioteca “matplotlib” proporciona multitud de herramientas para la visualización de datos. Entre ellas, la función “bar” dispone de varias opciones para realizar diagramas de barras de distintas características. Además, también se ha utilizado la función “subplot” que permite presentar varias gráficas al mismo tiempo.

Los resultados arrojados por el PCA implementado en el SCRIPT 3 pueden verse en la Figura 23.

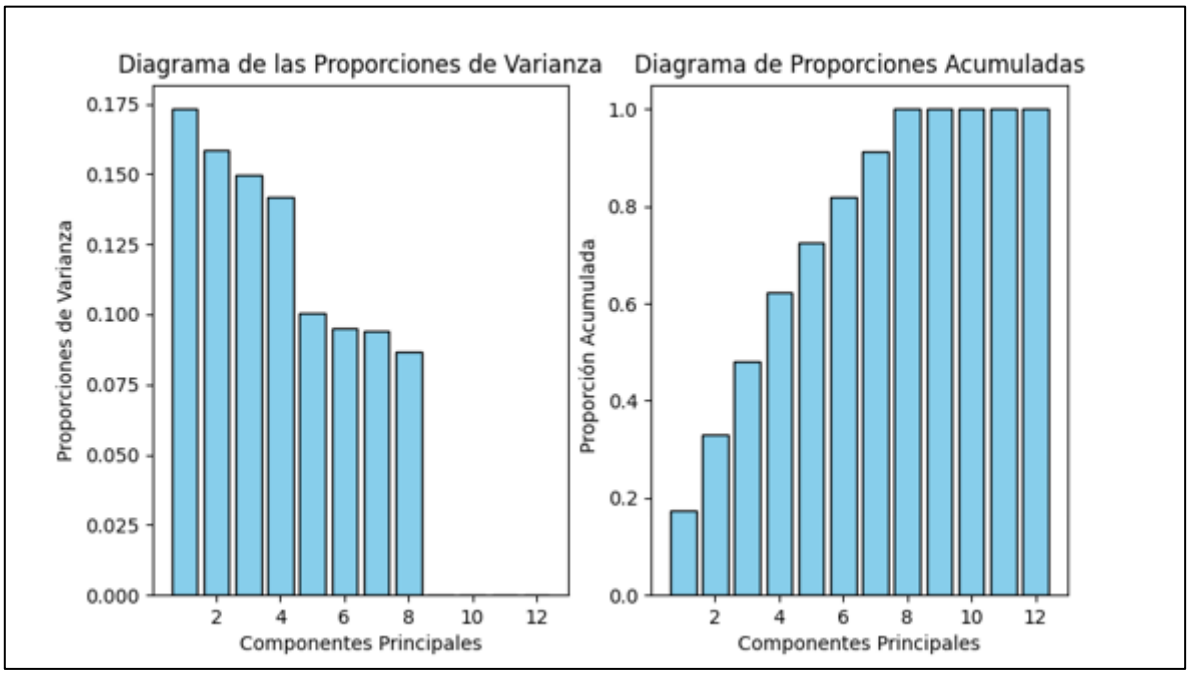


Figura 23. Diagramas de barras de las proporciones de varianza arrojadas por el PCA aplicado a 12 variables

Analizando los resultados arrojados por el PCA para las doce variables, se puede observar que seguía sucediendo lo mismo que para el caso de las veinte y nueve variables. Todas las proporciones de varianza son de valores similares excepto cuatro de ellas debido a que se ha seguido manteniendo las cuatro relaciones geométricas que generan cuatro variables dependientes.

Retrocediendo nuevamente a la fase de ‘estudio y comprensión de los datos’, y volviendo a analizar en profundidad las características de las doce variables que se habían acotado, se observó que las variables correspondientes a la vía directa eran independientes de las de la vía desviada. Por tanto, se decidió reducir a la mitad el vector de datos, generar un *dataset* sintético para las seis variables de la vía directa y volver a aplicarle el PCA. Este vector se encuentra descrito en la Tabla 5.

Sin embargo, los resultados arrojados por el PCA no muestran diferencias con los anteriores casos. En la Figura 24 se puede observar cómo no existe ninguna componente principal que domine sobre el resto y que existen dos componentes cuya proporción de varianza es nula debido a las dos relaciones geométricas (**Ecuación 1** y **Ecuación 2**) explicadas en el apartado 4.1.

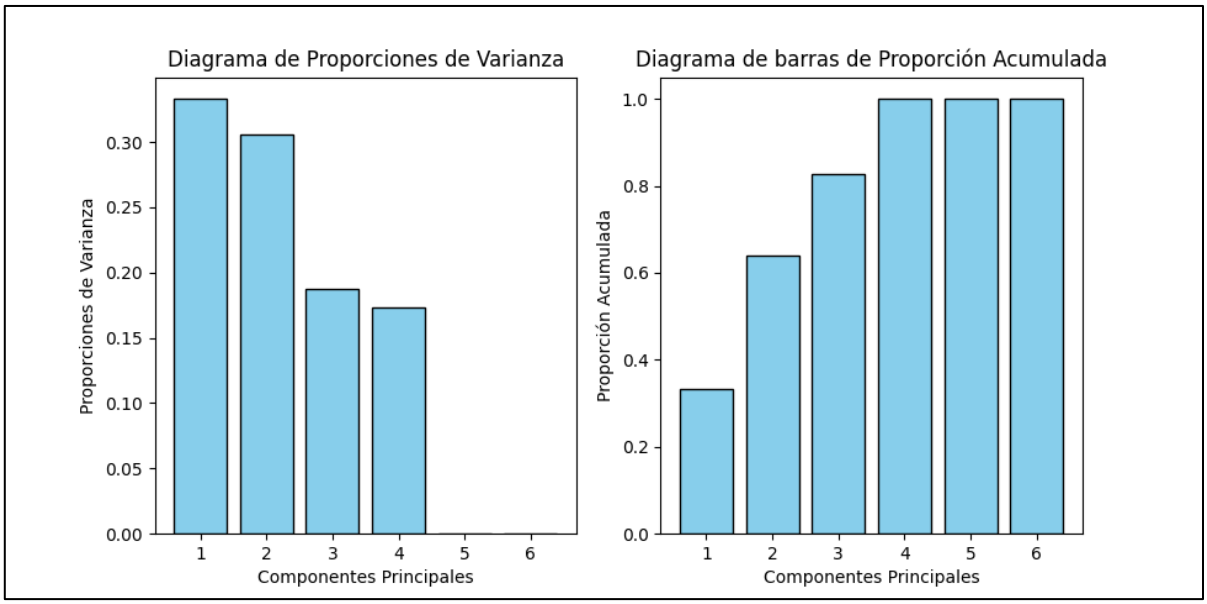


Figura 24.Diagramas de barras de las proporciones de varianza arrojadas por el PCA aplicado a 6 variables

Se llegó a la conclusión de que, al generar todos los datos de manera sintética salvo las variables que dependían de las relaciones geométricas, no era posible el surgimiento de correlaciones entre variables ya que éstas estaban generadas de manera completamente aleatoria.

Después de alcanzar esta conclusión, se programó un algoritmo donde se realizaron una serie de pruebas en las que se implementaban correlaciones sintéticas de manera adicional al proceso de generación de datos que se venía realizando hasta ahora, con el fin de comprobar si el modelo las detectaba. El algoritmo completo se puede ver en el SCRIPT 4.

En concreto se realizaron tres pruebas que se detallan a continuación:

- **Prueba 1**: se implementa una correlación sintética en la que la "entrecalle mínima" se obtiene como resultado de la muestra anterior de dicha variable más una combinación lineal de la muestra anterior y dos muestras anterior del "ancho de vía". En esta primera prueba, todas las variables correlacionadas pertenecen al mismo punto de medición.

  Ecuación 3

  $$entcalle\ mín(i) = entcalle\ mín(i-1) + \ ancho\ de\ vía(i-1) - ancho\ de\ vía\ (i-2)$$

- **Prueba 2**: se implementa una correlación sintética en la que el "paso libre de rueda" se obtiene como resultado de la muestra anterior de dicha variable más una combinación lineal de la muestra anterior y dos muestras anterior del "ancho de vía del punto donde se mide la cota de protección". Es decir, se trata de una correlación de variables de distintos puntos de medición.

  Ecuación 4

$$paso\ libre(i) = paso\ libre(i-1) + 2.1 * (ancho\ de\ vía\ de\ la\ cota\ de\ protección(i-1) - ancho\ de\ vía\ de\ la\ cota\ de\ protección\ (i-2))/2$$

- **Prueba 3**: se implementa una correlación sintética en la que la "entrecalle mínima" se obtiene como resultado de la muestra anterior de dicha variable más una combinación lineal de la muestra anterior y dos muestras anterior de la "entrecalle carril-contracarril". Es decir, se trata de una correlación entre variables del mismo orden de magnitud, pero medidas en puntos distintos de la estructura.

  Ecuación 5

$$ecalle\ min(i) = entrecalle\ min(i-1) + entrecalle\ contracarril\ (i-1) - entrecalle\ contracarril(i-2)$$

Estas tres pruebas se implementaron en el SCRIPT 4 y se les aplicó el PCA para observar si el modelo detectaba las correlaciones sintéticas añadidas. También se aplicó el PCA a un *dataset* generado sintéticamente siguiendo el mismo proceso que se venía realizando hasta ahora y con la misma magnitud de muestras que el resto de las pruebas.

Haciendo uso de la función "plot" de la biblioteca "matplotlib", se ha implementado un análisis gráfico en el que se proyecta la evolución de las proporciones de varianza para cada componente principal de cada una de las tres pruebas realizadas y del *dataset* generado de manera aleatoria. La Figura 25 muestra la gráfica comparativa descrita.

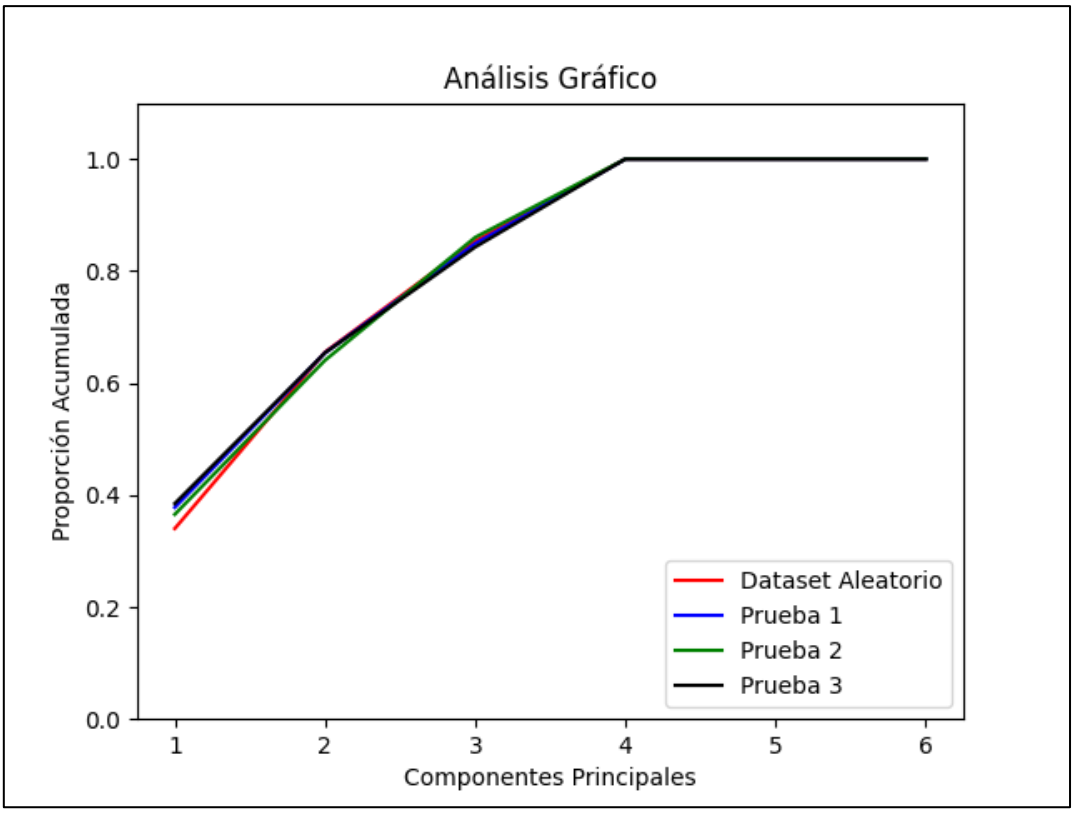


Figura 25. Resultados de la ejecución del SCRIPT 4

Se puede observar cómo todas las gráficas prácticamente se solapan en una. Esto quiere decir que el modelo no ha sido capaz de detectar las correlaciones sintéticas ya que las proporciones de varianza de todas las pruebas son muy similares entre ellas y muy similares a las del *dataset* aleatorio a su vez. Además, los datos generados en base a estas correlaciones sintéticas se salían del rango de funcionamiento, lo que hace que pierdan todo el valor de análisis.

Una hipótesis de este suceso es que, aunque se hayan implementado nuevas correlaciones, éstas son sintéticas y continúan estando basadas en datos aleatorios, por lo que el algoritmo no es capaz de detectar ninguna correlación adicional que aporte información nueva al problema.

4.4 SIMULACIÓN DE ESCENARIOS PARA VALIDACIÓN

En un principio, la línea de investigación del proyecto era tratar de identificar patrones y tendencias entre los valores de los distintos parámetros que definían el estado saludable de un aparato de vía tipo A, con el objetivo de implementar un modelo que fuese capaz de detectar la distorsión de dichas correlaciones cuando ocurriera un fallo en la estructura.

Después de no lograr detectar correlaciones entre los valores de estado saludable y, por consiguiente, no conseguir avanzar en la lía de investigación propuesta, se tomó la decisión de realizar un cambio de enfoque en la investigación. La nueva línea de investigación, con una estrategia inversa a la anterior, iba a tratar de implementar un modelo que detectara correlaciones cuando los datos manifestaran situaciones anómalas que hicieran presagiar que se iba a producir un fallo o que existía algún defecto en la estructura.

Teniendo en cuenta que este nuevo enfoque iba a conllevar la realización de numerosas comparaciones entre *datasets* que permitieran detectar patrones en situaciones anómalas, se elaboró un estudio para comprobar la influencia del número de muestras de un *dataset* en la precisión de la técnica PCA.

El estudio elaborado consistía en implementar un algoritmo (SCRIPT 5) que generara *datasets* compuestos por seis variables y distintos tamaños de muestras (10,100,1000 y 10000), aplicarles la técnica PCA e iterar este proceso un número de veces considerable, con el objetivo de proyectar la variación de proporción de varianza acumulada arrojada por el PCA para cada tamaño de muestra. En concreto, se comparó la proporción acumulada de la tercera componente principal ya que aportaba resultados bastante esclarecedores. Los resultados del estudio se pueden observar de manera gráfica en la Figura 26.

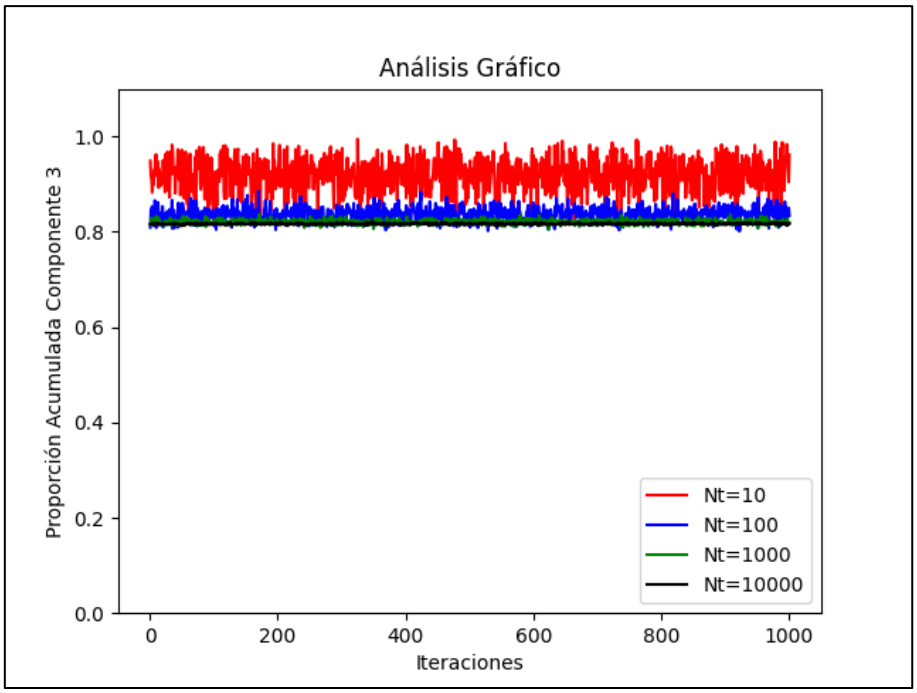


Figura 26. Influencia del tamaño de muestra en la precisión del PCA

Se puede observar que la variación en los *datasets* con pequeños tamaños de muestra es muy grande y conforme el tamaño de muestra aumenta, esta variación va disminuyendo. Por tanto, se puede concluir que, tal y como explica la teoría vista en el apartado 3.3, aumentar el tamaño de muestra del *dataset* mejora la precisión de la técnica PCA.

Otra conclusión destacable extraída del estudio es que para poder realizar comparaciones y análisis entre *datasets*, éstos deben tener el mismo tamaño de muestra, ya que, como se puede observar en la Figura 26, si se comparan dos *datasets* de distintos tamaños de muestra, las proporciones de varianza arrojadas por el PCA pueden inducir a detección de falsos patrones.

Teniendo todas estas conclusiones en cuenta, se ha desarrollado un algoritmo (SCRIPT 6) que es capaz de detectar determinadas anomalías en el estado de la estructura que permiten desarrollar funciones de detección y predicción de fallos en los aparatos de vía tipo A.

El algoritmo está entrenado con *datasets* compuestos por las seis variables de la Tabla 5 y un tamaño de muestra de diez muestras que, en la actualidad, según las inspecciones programadas en el plan preventivo, corresponderían a cinco años de adquisición de muestras (dos inspecciones al año). Al introducirle un *dataset* con las características descritas, el algoritmo es capaz de detectar, mediante el reconocimiento de patrones estadísticos de la técnica PCA si se produce algún aumento o disminución brusco en cualquiera de las variables que definen el estado de la estructura.

En el SCRIPT 6 se puede contemplar cómo se desarrollan seis pruebas de validación del algoritmo en las que se diseñan funciones de crecimiento sintéticas que simulan el incremento de cualquiera de las seis variables debido al desgaste, defecto o fallo del aparato de vía. Las funciones de crecimiento sintéticas suman al valor de la muestra anterior un valor aleatorio entre '0' y '1', simulando un crecimiento progresivo y a la vez anómalo para cualquier parámetro de la estructura. Por ende, las funciones sintéticas tienen la siguiente forma:

$$variable(i) = variable\ (i-1)\ +\ np.random.randint(0,2,1)$$

La Figura 27 muestra una comparación entre un *dataset* que contiene valores saludables y un *dataset* con la variable "ancho de vía en el punto de medición de la entrecalle mínima" modificada sintéticamente con la función de crecimiento descrita previamente. Se realiza un proceso iterativo de 1000 iteraciones con el objetivo de obtener resultados concluyentes.

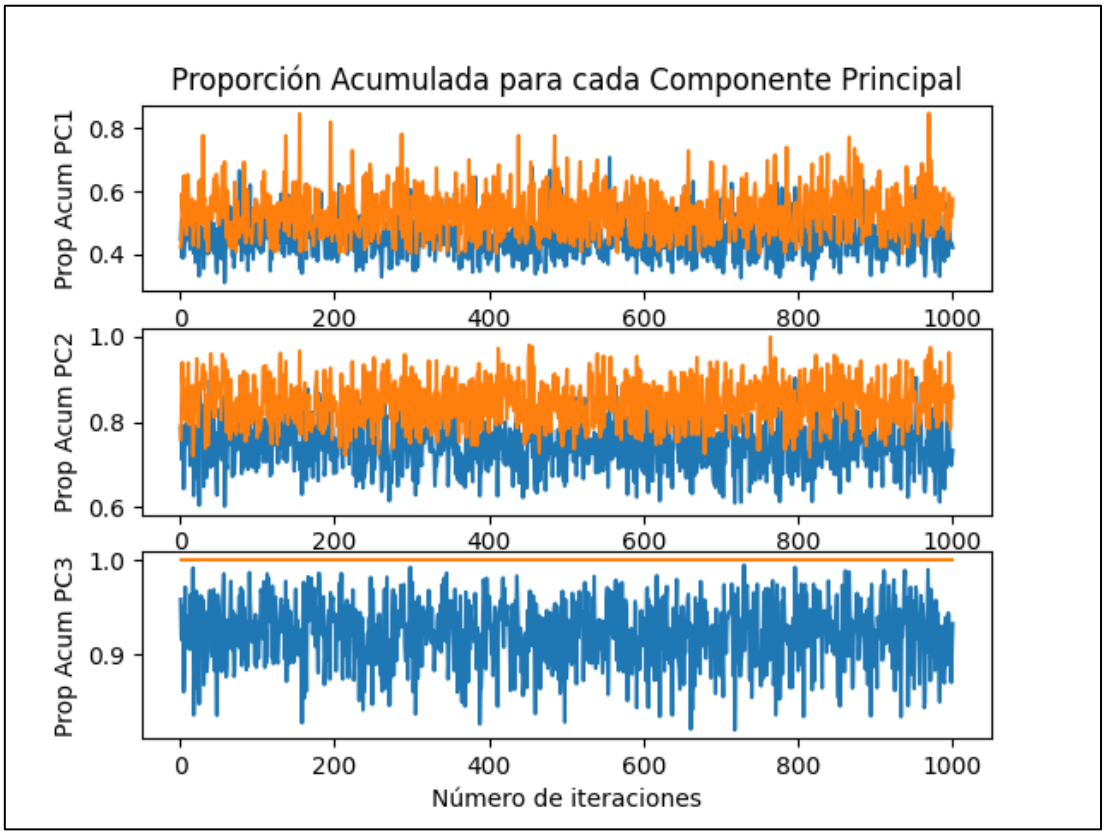


Figura 27. Prueba de validación del "ancho de vía en el punto de medición de la entrecalle mínima"

Se puede observar que, así como en las dos primeras componentes principales no se puede extraer ninguna conclusión, la proporción acumulada de la tercera componente refleja una distinción significativa entre el *dataset* generado con datos saludables y el *dataset* modificado con la función de

crecimiento sintética. Esta distinción de resultados entre el *dataset* en buen estado de salud y el *dataset* que simula una anomalía sintética valida el algoritmo de detección y predicción de fallos implementado.

En el ANEXO I, de la Figura 29 a la Figura 33, se recogen los resultados gráficos del resto de pruebas de validación realizadas en el SCRIPT 6. En ellas se puede constatar la validez del algoritmo para detectar situaciones anómalas concretas en las seis variables que definen la condición del aparato de vía tipo A.

El algoritmo además es capaz de detectar si se ha producido un fallo concreto en la aguja, elemento fundamental del aparato de vía sometido a un elevado estrés mecánico debido a cargas dinámicas, fuerzas laterales, impactos y vibraciones. Este tipo de fallo es difícil de detectar porque el desgaste producido en la aguja es invisible en la medición del "ancho de vía", ya que aumentaría el "paso libre de rueda en el cambio" pero a su vez disminuiría en la misma proporción la "entrecalle mínima". Esto es debido a las condiciones geométricas de la estructura. (Véase Figura 30).

Por último, es reseñable destacar que el algoritmo también es capaz de detectar un fallo simultáneo en dos zonas distintas del aparato de vía (Figura 33). Esta función dota al algoritmo de una mayor capacidad de detección haciéndolo más robusto y eficiente.

5 CONCLUSIONES

El presente proyecto ha conseguido desarrollar una herramienta de detección de anomalías que pueden reflejar un estado de fallo en la estructura o derivar en un fallo de manera repentina. Esta herramienta se trata de un algoritmo implementado con una técnica de reconocimiento de patrones estadísticos como es el PCA, la cual es muy empleada en la actualidad en el campo del Machine Learning con el objetivo de mejorar la eficiencia operativa y ayudar en la toma de decisiones de cualquier ámbito o sector.

El algoritmo desarrollado dispone de un gran potencial de aplicación, puesto que permite mejorar el plan de mantenimiento de los sistemas ferroviarios, cuya estrategia de mantenimiento actual consiste en un procedimiento de aplicación de tareas preventivas. La mejora supondría una evolución hacia un plan de mantenimiento predictivo que permitiera reducir el número de paradas no planificadas, mejorar la seguridad y reducir los costes de operación.

Este algoritmo constituye una herramienta piloto verificada y validada (datos sintéticos) para unas especificaciones concretas. En caso de conocer datos reales de las variables tratadas en este proyecto, se podría realizar un estudio exhaustivo de dichas variables y ajustar la herramienta piloto para que fuese capaz de detectar posibles tendencias o correlaciones ocultas entre los distintos parámetros, con la suficiente antelación para poder llevar a cabo un mantenimiento predictivo.

A pesar de que el algoritmo se ha diseñado para un tipo concreto de segmento especial de vía, con unas características y parámetros determinados, la metodología seguida en el desarrollo de este proyecto es aplicable a cualquier infraestructura ferroviaria. Esto dota al trabajo realizado de una gran versatilidad y abre un abanico de nuevas líneas de investigación.

Nuevas líneas de investigación

Los resultados obtenidos en el presente proyecto corresponden a un tipo concreto de aparato de vía. Sin embargo, la metodología y el modelo implementado es compatible con cualquier tipo de segmento especial de vía ferroviaria. Para ello, se deberá analizar los parámetros del nuevo tipo de aparato de vía y comparar con los parámetros considerados en ente proyecto. En caso de ser distintos será necesario comenzar desde el principio de la metodología desarrollada en este proyecto. Comenzando con la fase de comprensión y análisis de los datos, donde se deberá realizar un estudio para determinar cuáles son los parámetros o variables que mejor definen el estado estructural del aparato de vía y finalizando con la fase de evaluación del modelo.

La herramienta desarrollada en el presente trabajo está validada, mediante datos sintéticos, para funciones crecientes y decrecientes especificas diseñadas para este proyecto. Existe una mejora potencial del modelo desarrollado que se basaría en realizar un estudio matemático que genere funciones reales del desgaste del aparato de vía reflejado en los distintos parámetros y, de esta manera, utilizar dichas funciones para entrenar el modelo implementado con el objetivo de detectar estas nuevas tendencias proyectadas por dichas funciones.

Por último, otra nueva línea de investigación sería el desarrollo de una mejora en los sistemas de adquisición de datos, incluyendo sensores, instrumentos de monitorización y sistemas de supervisión específicos. Todo ello debería ir acompañado de una evolución de los equipos de almacenamiento de datos incluyendo tecnologías como el almacenamiento en la nube y el procesamiento distribuido. Esto permitiría disponer de una gran fuente de datos y ayudaría a implementar un mayor número de técnicas avanzadas de análisis de datos que aportaran nueva información que permitiera reajustar la planificación de mantenimiento de la empresa con el objetivo de hacerla más eficiente.

BIBLIOGRAFÍA

[1] "Instituto Nacional de Estadística," *https://www.ine.es/index.htm*, 2023.

[2] "Ministerio de Transportes, Movilidad y Agenda Urbana," *https://www.mitma.gob.es/el-ministerio/campanas-de-publicidad/2021-anio-europeo-del-ferrocarril/conociendo-el-ferrocarril*, 2023.

[3] Network Rail, "ENGINEERING VIDEOS HELP BRING RAIL LIFE TO THE CLASSROOM," *https://www.networkrailmediacentre.co.uk/news/engineering-videos-help-bring-rail-life-to-the-classroom*, Apr. 2012.

[4] Dirección Técnica (RENFE) and Dirección de Mantenimiento de Infraestructura (RENFE), "NAV 3-6-0.1: DESVÍOS. Características de los tipos y modelos.," Jul. 1992.

[5] "Aparatos de vía: los desvíos ferroviarios," *https://masqueingenieria.com/blog/aparatos-de-via-los-desvios/*, 2016.

[6] Grupo de trabajo GT-208, "NAV 7-3-8.2: INSPECCIÓN DE APARATOS DE VÍAS," Feb. 2022.

[7] J. P. Mora, "¿Qué es la Ciencia de Datos, el aprendizaje automático (ML), el Big Data y cuáles son sus usos?"

[8] O. of S. A. Federal Railroad Administration, "Ten year accident/incident overview," *https://safetydata.fra.dot.gov/OfficeofSafety/publicsite/Query/TenYearAccidentIncidentOverview.aspx*, 2022.

[9] U. National Audit Office: London, R. Sheeran, and A. Jenner, "A Short Guide to Network Rail," Jul. 2015.

[10] A. López-Pita, P. F. Teixeira, C. Casas, A. Bachiller, and P. A. Ferreira, "Maintenance costs of high-speed lines in Europe state of the art," *Transp Res Rec*, no. 2043, 2008, doi: 10.3141/2043-02.

[11] A. Jezzini, M. Ayache, L. Elkhansa, B. Makki, and M. Zein, "Effects of predictive maintenance(PdM), Proactive maintenace(PoM) & Preventive maintenance(PM) on minimizing the faults in medical instruments," in *2013 2nd International Conference on Advances in Biomedical Engineering, ICABME 2013*, 2013. doi: 10.1109/ICABME.2013.6648845.

[12] J. Xie, J. Huang, C. Zeng, S. H. Jiang, and N. Podlich, "Systematic literature review on data-driven models for predictive maintenance of railway track: Implications in geotechnical engineering," *Geosciences (Switzerland)*, vol. 10, no. 11. 2020. doi: 10.3390/geosciences10110425.

[13] Z. Li and Q. He, “Prediction of Railcar Remaining Useful Life by Multiple Data Source Fusion,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, 2015, doi: 10.1109/TITS.2015.2400424.

[14] S. Hajizadeh, A. Núñez, and D. M. J. Tax, “Semi-supervised Rail Defect Detection from Imbalanced Image Data,” in *IFAC-PapersOnLine*, 2016. doi: 10.1016/j.ifacol.2016.07.014.

[15] A. K. S. Jardine, D. Lin, and D. Banjevic, “A review on machinery diagnostics and prognostics implementing condition-based maintenance,” *Mechanical Systems and Signal Processing*, vol. 20, no. 7. 2006. doi: 10.1016/j.ymssp.2005.09.012.

[16] A. Falamarzi, S. Moridpour, and M. Nazem, “A review of rail track degradation prediction models,” *Australian Journal of Civil Engineering*, vol. 17, no. 2. 2019. doi: 10.1080/14488353.2019.1667710.

[17] I. Soleimanmeigouni, A. Ahmadi, and U. Kumar, “Track geometry degradation and maintenance modelling: A review,” *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, vol. 232, no. 1. 2018. doi: 10.1177/0954409716657849.

[18] M. Sol-Sánchez and G. D’Angelo, “Review of the design and maintenance technologies used to decelerate the deterioration of ballasted railway tracks,” *Construction and Building Materials*, vol. 157. 2017. doi: 10.1016/j.conbuildmat.2017.09.007.

[19] N. Elkhoury, L. Hitihamillage, S. Moridpour, and D. Robert, “Degradation Prediction of Rail Tracks: A Review of the Existing Literature,” *The Open Transportation Journal*, vol. 12, no. 1, 2018, doi: 10.2174/1874447801812010088.

[20] C. Higgins and X. Liu, “Modeling of track geometry degradation and decisions on safety and maintenance: A literature review and possible future research directions,” *Proc Inst Mech Eng F J Rail Rapid Transit*, vol. 232, no. 5, 2018, doi: 10.1177/0954409717721870.

[21] M. Chenariyan Nakhaee, D. Hiemstra, M. Stoelinga, and M. van Noort, “The Recent Applications of Machine Learning in Rail Track Maintenance: A Survey,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019. doi: 10.1007/978-3-030-18744-6_6.

[22] R. An, Q. Sun, F. Wang, W. Bai, X. Zhu, and R. Liu, “Improved Railway Track Geometry Degradation Modeling for Tamping Cycle Prediction,” *J Transp Eng A Syst*, vol. 144, no. 7, 2018, doi: 10.1061/jtepbs.0000149.

[23] M. Rosyidi *et al.*, “Predictive Maintenance with PCA Approach for Multi Automated Railroad Crossing System (ARCS) in the Framework of Prognostic and Health Management (PHM) Planning,” in *Journal of Physics: Conference Series*, 2022. doi: 10.1088/1742-6596/2322/1/012090.

[24] J. Sadeghi and H. Askarinejad, “Development of improved railway track degradation models,” *Structure and Infrastructure Engineering*, vol. 6, no. 6, 2010, doi: 10.1080/15732470801902436.

[25] M. Audley and J. D. Andrews, “The effects of tamping on railway track geometry degradation,” *Proc Inst Mech Eng F J Rail Rapid Transit*, vol. 227, no. 4, 2013, doi: 10.1177/0954409713480439.

[26] C. Jia, W. Xu, F. Wang, and H. Wang, “Track irregularity time series analysis and trend forecasting,” *Discrete Dyn Nat Soc*, vol. 2012, 2012, doi: 10.1155/2012/387857.

[27] H. F. Lam, J. H. Yang, Q. Hu, and C. T. Ng, “Railway ballast damage detection by Markov chain Monte Carlo-based Bayesian method,” *Struct Health Monit*, vol. 17, no. 3, 2018, doi: 10.1177/1475921717717106.

[28] L. Bai, R. Liu, Q. Sun, F. Wang, and P. Xu, “Markov-based model for the prediction of railway track irregularities,” *Proc Inst Mech Eng F J Rail Rapid Transit*, vol. 229, no. 2, 2015, doi: 10.1177/0954409713503460.

[29] H. Guler, “Prediction of railway track geometry deterioration using artificial neural networks: A case study for Turkish state railways,” *Structure and Infrastructure Engineering*, vol. 10, no. 5, 2014, doi: 10.1080/15732479.2012.757791.

[30] Y. Jiang, H. Wang, G. Tian, Q. Yi, J. Zhao, and K. Zhen, “Fast classification for rail defect depths using a hybrid intelligent method,” *Optik (Stuttg)*, vol. 180, 2019, doi: 10.1016/j.ijleo.2018.11.053.

[31] A. Falamarzi, S. Moridpour, and M. Nazem, “Development of a tram track degradation prediction model based on the acceleration data,” *Structure and Infrastructure Engineering*, vol. 15, no. 10, 2019, doi: 10.1080/15732479.2019.1615963.

[32] C. W. Tan, G. I. Webb, F. Petitjean, and P. Reichl, “Machine learning approaches for tamping effectiveness prediction,” *2017 International Heavy Haul Association Conference (IHHA*, no. September, 2017.

[33] Y. LeCun, G. Hinton, and Y. Bengio, “Deep learning (2015), Y. LeCun, Y. Bengio and G. Hinton,” *Nature*, vol. 521, 2015.

[34] L. Rokach, “Ensemble-based classifiers,” *Artif Intell Rev*, vol. 33, no. 1–2, 2010, doi: 10.1007/s10462-009-9124-7.

[35] I. Cárdenas-Gallo, C. A. Sarmiento, G. A. Morales, M. A. Bolivar, and R. Akhavan-Tabatabaei, “An ensemble classifier to predict track geometry degradation,” *Reliab Eng Syst Saf*, vol. 161, 2017, doi: 10.1016/j.ress.2016.12.012.

[36] A. Lasisi and N. Attoh-Okine, “Machine Learning Ensembles and Rail Defects Prediction: Multilayer Stacking Methodology,” *ASCE ASME J Risk Uncertain Eng Syst A Civ Eng*, vol. 5, no. 4, 2019, doi: 10.1061/ajrua6.0001024.

[37] J. Miralles Solé, *Proyectos de Inteligencia Artificial*. 2020.

[38] I. B. Machines. IBM, “Manual CRISP-DM de IBM SPSS Modeler,” *IBM Corporation*, p. 56, 2012, [Online]. Available: http://www.ibm.com/spss.%0Aftp://public.dhe.ibm.com/software/analytics/spss/documentation/modeler/15.0/es/CRISP-DM.pdf

[39] S. Moro, R. M. S. Laureano, and P. Cortez, “Using data mining for bank direct marketing: An application of the CRISP-DM methodology,” in *ESM 2011 - 2011 European Simulation and Modelling Conference: Modelling and Simulation 2011*, 2011.

[40] W. Y. Ayele, “Adapting CRISP-DM for idea mining a data mining process for generating ideas using a textual dataset,” *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 6, 2020, doi: 10.14569/IJACSA.2020.0110603.

[41] Syntho, “What is synthetic data?,” *https://www.syntho.ai/es/what-is-synthetic-data/*, 2023.

[42] A. Fernández-Navamuel, “Desarrollo de una herramienta de detección de daños para la toma de decisión en gestión de puentes(Damage detection tool development for decision making in Bridge Management),” 2018.

[43] M. Korolov, “What is synthetic data? Generated data to help your AI strategy,” *https://www.cio.com/article/305902/what-is-synthetic-data-generated-data-to-help-your-ai-strategy.html*, Mar. 2022.

[44] S. Nikolenko, *Synthetic Data for Deep Learning*. 2019.

[45] S. Baressi Šegota, N. Anđelić, M. Šercer, and H. Meštrić, “Dynamics Modeling of Industrial Robotic Manipulators: A Machine Learning Approach Based on Synthetic Data,” *Mathematics*, vol. 10, no. 7, 2022, doi: 10.3390/math10071174.

[46] J. W. Kim and B. Jang, “Deep learning-based privacy-preserving framework for synthetic trajectory generation,” *Journal of Network and Computer Applications*, vol. 206, 2022, doi: 10.1016/j.jnca.2022.103459.

[47] C. Tan, “A model-based approach to generate dynamic synthetic test data,” in *Proceedings - 2019 IEEE 12th International Conference on Software Testing, Verification and Validation, ICST 2019*, 2019. doi: 10.1109/ICST.2019.00063.

[48] K. Meinke, “Active machine learning to test autonomous driving,” in *Proceedings - 2021 IEEE 14th International Conference on Software Testing, Verification and Validation Workshops, ICSTW 2021*, 2021. doi: 10.1109/ICSTW52544.2021.00055.

[49] J. Dahmen and D. Cook, “SynSys: A synthetic data generation system for healthcare applications,” *Sensors (Switzerland)*, vol. 19, no. 5, 2019, doi: 10.3390/s19051181.

[50] A. Goncalves, P. Ray, B. Soper, J. Stevens, L. Coyle, and A. P. Sales, “Generation and evaluation of synthetic patient data,” *BMC Med Res Methodol*, vol. 20, no. 1, 2020, doi: 10.1186/s12874-020-00977-1.

[51] J. Noguer, I. Contreras, O. Mujahid, A. Beneyto, and J. Vehi, “Generation of Individualized Synthetic Data for Augmentation of the Type 1 Diabetes Data Sets Using Deep Learning Models,” *Sensors*, vol. 22, no. 13, 2022, doi: 10.3390/s22134944.

[52] S. Achuthan *et al.*, “Leveraging deep learning algorithms for synthetic data generation to design and analyze biological networks,” *Journal of Biosciences*, vol. 47, no. 3. 2022. doi: 10.1007/s12038-022-00278-3.

[53] B. F. L. Sieow, R. De Sotto, Z. R. D. Seet, I. Y. Hwang, and M. W. Chang, “Synthetic Biology Meets Machine Learning,” in *Methods in Molecular Biology*, 2023. doi: 10.1007/978-1-0716-2617-7_2.

[54] G. Mlslrll *et al.*, “Data Integration and Mining for Synthetic Biology Design,” *ACS Synth Biol*, vol. 5, no. 10, 2016, doi: 10.1021/acssynbio.5b00295.

[55] J. L. A. Gardner, Z. Faure Beaulieu, and V. L. Deringer, “Synthetic data enable experiments in atomistic machine learning,” *Digital Discovery*, vol. 2, no. 3, pp. 651–662, 2023, doi: 10.1039/D2DD00137C.

[56] J. Tobin, “Real-World Robotic Perception and Control Using Synthetic Data,” 2019. [Online]. Available: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2019/EECS-2019-104.html

[57] J. Arents, M. Greitans, and B. Lesser, “Construction of a Smart Vision-Guided Robot System for Manipulation in a Dynamic Environment,” 2021.

[58] A. P.-C. Agarwal, A. Vladyslav, and P.-C. Chen, “Exelon Uses Synthetic Data Generation of Grid Infrastructure to Automate Drone Inspection,” *https://developer.nvidia.com/blog/exelon-uses-synthetic-data-generation-of-grid-infrastructure-to-automate-drone-inspection/*, 2023.

[59] T. Nguyen *et al.*, “Pennsyn2real: Training object recognition models without human labeling,” *IEEE Robot Autom Lett*, vol. 6, no. 3, 2021, doi: 10.1109/LRA.2021.3070249.

[60] G. Paulin and M. Ivasic-Kos, “Review and analysis of synthetic dataset generation methods and techniques for application in computer vision,” *Artif Intell Rev*, 2023, doi: 10.1007/s10462-022-10358-3.

[61] L. Zhang, A. Gonzalez-Garcia, J. Van De Weijer, M. Danelljan, and F. S. Khan, “Synthetic Data Generation for End-To-End Thermal Infrared Tracking,” *IEEE Transactions on Image Processing*, vol. 28, no. 4, 2019, doi: 10.1109/TIP.2018.2879249.

[62] B. Flanagan, R. Majumdar, and H. Ogata, “Fine Grain Synthetic Educational Data: Challenges and Limitations of Collaborative Learning Analytics,” *IEEE Access*, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3156073.

[63] J. J. Vie, T. Rigaux, and S. Minn, “Privacy-Preserving Synthetic Educational Data Generation,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2022. doi: 10.1007/978-3-031-16290-9_29.

[64] C. De Nova and C. Carbajal De Nova, “Synthetic data. A proposed method for applied risk management,” 2017.

[65] S. Assefa, D. Dervovic, M. Mahfouz, T. Balch, P. Reddy, and M. Veloso, “Generating synthetic data in finance: opportunities, challenges and pitfalls,” 2020.

[66] K. Thompson and H. J. Kim, “Incorporating Economic Conditions in Synthetic Microdata for Business Programs,” *Journal of Survey Statistics and Methodology*, vol. 10, no. 3. 2022. doi: 10.1093/jssam/smab054.

[67] F. K. Dankar, M. K. Ibrahim, and L. Ismail, “A Multi-Dimensional Evaluation of Synthetic Data Generators,” *IEEE Access*, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3144765.

[68] M. A. Fitriani and D. C. Febrianto, “Data Mining for Potential Customer Segmentation in the Marketing Bank Dataset,” 2021.

[69] J. Marin, “Using Synthetic Data in Digital Marketing,” *https://towardsdatascience.com/using-synthetic-data-in-digital-marketing-c972b96e5c*, Oct. 2022.

[70] P. Martinez-Gonzalez *et al.*, “UnrealROX+: An Improved Tool for Acquiring Synthetic Data from Virtual 3D Environments,” in *Proceedings of the International Joint Conference on Neural Networks*, Institute of Electrical and Electronics Engineers Inc., Jul. 2021. doi: 10.1109/IJCNN52387.2021.9534447.

[71] E. Bonetto, C. Xu, and A. Ahmad, “GRADE: Generating Realistic Animated Dynamic Environments for Robotics Research,” Mar. 2023, [Online]. Available: http://arxiv.org/abs/2303.04466

[72] A. Siyaev and G. S. Jo, “Neuro-Symbolic Speech Understanding in Aircraft Maintenance Metaverse,” *IEEE Access*, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3128616.

[73] A. Linden, “Is Synthetic Data the Future of AI?,” *https://www.gartner.com/en/newsroom/press-releases/2022-06-22-is-synthetic-data-the-future-of-ai#:~:text=Gartner%20estimates%20that%20by%202030,regionally%20from%20August%20through%20November.*, 2022.

[74] P. Bruce, A. Bruce, and P. Gedeck, *Estadística práctica para ciencia de datos con R y Python*. 2022.

[75] A. L. S. Saabith, T. Vinothraj, and M. M. M. Fareez, “A review on Python libraries and Ides for Data Science,” *International Journal of Research in Engineering and Science*, vol. 09, no. 11, 2021.

[76] I. Stancin and A. Jovic, “An overview and comparison of free Python libraries for data mining and big data analysis,” in *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2019 - Proceedings*, 2019. doi: 10.23919/MIPRO.2019.8757088.

[77] J. Amat Rodrigo, “PCA con Python,” *https://www.cienciadedatos.net/documentos/py19-pca-python.html*, Dec. 2020.

[78] Dirección Técnica (RENFE) and Dirección de Mantenimiento de Infraestructura (RENFE), “NAV 7-5-3.1: CONSERVACIÓN DE LA VÍA. Mantenimiento de desvíos y otros aparatos de vía,” 1995.

ANEXO I

Tipo A (v<30km/h), [1668 mm]

LEYENDA: Actuación programada / Actuación inmediata

VALORES LÍMITES MÍNIMOS: (1) 1.626 mm Valor ETI; (2) 70 mm Valor ETI; Cota de seguridad; Valor ETI

ESPACIO RESERVADO PARA LAS EMPRESAS:

TIPO APARATO: (Seleccionar) Tipo A (v>30km/h)[1668 mm] | UBICACIÓN: Irun | Nº APARATO: T7 | LINEA: | FECHA: 10/10/2021

CAMBIO (*)		1er Cerrojo		2º Cerrojo		3er Cerrojo		4º Cerrojo	
APERTURA EN BIELA (**)		V direc	V desv	V direc	V desv	V direc	V desv	V direc	V desv
	Real								
Nominal:165 mm; Tol.: [-15, 15] mm	Corregido								
ENCERROJAMIENTO (**)		V direc	V desv	V direc	V desv	V direc	V desv	V direc	V desv
	Real								
Nominal:45 mm; Tol.: [-10, 10] mm	Corregido								

ANCHOS Y PERALTES DE VÍA DIRECTA					ANCHOS Y PERALTES DE VÍA DESVIADA				
ANCHO			PERALTE		ANCHO			PERALTE	
[-2, 10] mm Progr		[-3, 15] mm Inmediata	[-5, 5] mm Progr	[-10, 10] mm Inmediata	[-2, 10] mm Progr		[-3, 15] mm Inmediata	[-5, 5] mm Progr	[-10, 10] mm Inmediata
TRAVIESA	TEORICO	REAL	TEORICO	REAL	TRAVIESA	TEORICO	REAL	TEORICO	REAL
-15	1668	1661,0	0						
-10	1668	1665,0	0						
-5	1668	1668,0	0						
JCA	1668	1674,0	0						
5	1668	1668,0	0		5	1668	1682,0	0	
10	1668	1661,0	0		10	1668	1682,0	0	
15	1668	1658,0	0		15	1668	1684,0	0	
20	1668	1668,0	0		20	1668	1682,0	0	
25	1668	1668,0	0		25	1668	1680,0	0	
30	1668	1667,0	0		30	1668	1678,0	0	
35	1668	1667,0	0		35	1668	1675,0	0	
40	1668	1669,0	0		40	1668	1673,0	0	
45	1668	1668,0	0		45	1668	1665,0	0	
	1668		0			1668		0	
	1668		0			1668		0	
	1668		0			1668		0	
	1668		0			1668		0	
	1668		0			1668		0	
	1668		0			1668		0	
	1668		0			1668		0	

VIA DIRECTA		VIA DESVIADA	
PASO LIBRE DE RUEDA EN EL CAMBIO: < 1618 mm			
REAL	CORREGIDO	REAL	CORREGIDO
ENTRECALLE MINIMA: ≥ 58 mm; Tol.: < 58 mm Progr, < 55 mm Inmed			
REAL	CORREGIDO	REAL	CORREGIDO
64		88	
(1) COTA DE PROTECCION: 1628 mm; Tol.: [-2, 3] mm Prog, [-2, 5] mm Inmediata			
REAL	CORREGIDO	REAL	CORREGIDO
1626,0		1625,0	
ENTRECALLE CARRIL - CONTRACARRIL: 40 mm Tol.: [-2, 5] mm			
REAL	CORREGIDO	REAL	CORREGIDO
41		42	
PASO DE RUEDA LIBRE EN LAS PUNTAS DEL CRUZAMIENTO: < 1590 mm			
REAL	CORREGIDO	REAL	CORREGIDO
1586		1573	
PASO DE RUEDA LIBRE EN LA ENTRADA DE PATA LIEBRE: <1620 mm			
REAL	CORREGIDO	REAL	CORREGIDO
1591		1592	
PASO DE RUEDA LIBRE EN LA ENTRADA DE CONTRACARRIL: <1620 mm			
REAL	CORREGIDO	REAL	CORREGIDO
1610		1597	
ANCHURA DE LA GARGANTA DE GUÍA: >38 mm			
REAL	CORREGIDO	REAL	CORREGIDO
61		61	
PROFUNDIDAD DE LA GARGANTA DE GUÍA: >40 mm			
REAL	CORREGIDO	REAL	CORREGIDO
43		43	
(2) SOBREELEVACIÓN DE CONTRACARRILES: 20 mm Tol.: [-5, 5] mm			
REAL	CORREGIDO	REAL	CORREGIDO
1		1	

Figura 28. Hoja de control de inspección de aparato de vía tipo A

Tabla 2. Tabla resumen de los 41 parámetros medidos en una inspección de un aparato de vía tipo A

TABLA DE VARIABLES INICIALES				
VARIABLES	**IDENTIFICADOR**	**DESCRIPCIÓN [mm]**	**VALOR**	**REGLAS TOL**
Ancho/-15/direc	1	Ancho de vía directa a 15 traviesas de la JCA **[1666,1678] [1665,1683]**		1668,0
Ancho/-10/direc	2	Ancho de vía directa a 10 traviesas de la JCA **[1666,1678] [1665,1683]**		1668,0
Ancho/-5/direc	3	Ancho de vía directa a 5 traviesas de la JCA **[1666,1678] [1665,1683]**		1668,0
Ancho/JCA/direc	4	Ancho de vía directa en la JCA **[1666,1678] [1665,1683]**		1668,0
Ancho/5/direc	5	Ancho de vía directa 5 traviesas después de la JCA **[1666,1678] [1665,1683]**		1668,0
Ancho/10/direc	6	Ancho de vía directa 10 traviesas después de la JCA **[1666,1678] [1665,1683]**		1668,0
Ancho/15/direc	7	Ancho de vía directa 15 traviesas después de la JCA **[1666,1678] [1665,1683]**		1668,0
Ancho/20/direc	8	Ancho de vía directa 20 traviesas después de la JCA **[1666,1678] [1665,1683]**		1668,0
Ancho/25/direc	9	Ancho de vía directa 25 traviesas después de la JCA **[1666,1678] [1665,1683]**		1668,0
Ancho/30/direc	10	Ancho de vía directa 30 traviesas después de la JCA **[1666,1678] [1665,1683]**		1668,0
Ancho/35/direc	11	Ancho de vía directa 35 traviesas después de la JCA **[1666,1678] [1665,1683]**		1668,0
Ancho/40/direc	12	Ancho de vía directa 40 traviesas después de la JCA **[1666,1678] [1665,1683]**		1668,0
Ancho/45/direc	13	Ancho de vía directa 45 traviesas después de la JCA **[1666,1678] [1665,1683]**		1668,0
Entrecalle mínima/direc	14	Se medirá en el punto más próximo entre la aguja y la contraaguja **[>58] [>55]**		58,0
Cota de protección/direc	15	Distancia que debe existir entre la cara activa del contracarril y la punta del corazón. **Ancho Ibérico**: **[1626,1631] [1626,1633]**		1628,0
Entrecalle carril-contracarril/direc	16	Se mide en el mismo punto que la cota de protección **[38,45]**		40,0
Paso en la punta del cruzamiento/direc	17	Distancia entre el contracarril y la cara vertical de la pata de liebre a la altura de la punta del corazón **[<1590]**		1590,0
Paso en la entrada de pata de liebre/direc	18	Distancia entre la cara activa de la pata de liebre y la cara activa del carril opuesto **[<1620]**		1620,0
Paso en la entrada de contracarril/direc	19	Distancia entre la cara activa del contracarril y el carril del hilo opuesto de su vía **[<1620]**		1620,0
Anchura de la garganta de guía/direc	20	Distancia entre los extremos de la garganta de guía **[>38]**		38,0
Profundidad de la garganta de guía/direc	21	Se mide en el mismo punto que en la anchura de la garganta de guía **[>40]**		40,0
Sobreelevación de contracarriles/direc	22	El contracarril se sitúa más alto que el carril para el mejor guiado de los ejes de los vehículos **[15,25]**		20,0
Descuadre de la punta de las agujas/direc	23	Mide el ajuste de la punta de las agujas con el vehículo **[-2,2]**		0,0
Ancho/5/desv	24	Ancho de vía desviada 5 traviesas después de la JCA **[1666,1678] [1665,1683]**		1668,0
Ancho/10/desv	25	Ancho de vía desviada 10 traviesas después de la JCA **[1666,1678] [1665,1683]**		1668,0
Ancho/15/desv	26	Ancho de vía desviada 15 traviesas después de la JCA **[1666,1678] [1665,1683]**		1668,0
Ancho/20/desv	27	Ancho de vía desviada 20 traviesas después de la JCA **[1666,1678] [1665,1683]**		1668,0
Ancho/25/desv	28	Ancho de vía desviada 25 traviesas después de la JCA **[1666,1678] [1665,1683]**		1668,0
Ancho/30/desv	29	Ancho de vía desviada 30 traviesas después de la JCA **[1666,1678] [1665,1683]**		1668,0
Ancho/35/desv	30	Ancho de vía desviada 35 traviesas después de la JCA **[1666,1678] [1665,1683]**		1668,0
Ancho/40/desv	31	Ancho de vía desviada 40 traviesas después de la JCA **[1666,1678] [1665,1683]**		1668,0
Ancho/45/desv	32	Ancho de vía desviada 45 traviesas después de la JCA **[1666,1678] [1665,1683]**		1668,0
Entrecalle mínima/desv	33	Se medirá en el punto más próximo entre la aguja y la contraaguja **[>58] [>55]**		58,0
Cota de protección/desv	34	Distancia que debe existir entre la cara activa del contracarril y la punta del corazón. **Ancho Ibérico**: **[1626,1631] [1626,1633]**		1628,0
Entrecalle carril-contracarril/desv	35	Se mide en el mismo punto que la cota de protección **[38,45]**		40,0
Paso en la punta del cruzamiento/desv	36	Distancia entre el contracarril y la cara vertical de la pata de liebre a la altura de la punta del corazón **[<1590]**		1590,0
Paso en la entrada de pata de liebre/desv	37	Distancia entre la cara activa de la pata de liebre y la cara activa del carril opuesto **[<1620]**		1620,0
Paso en la entrada de contracarril/desv	38	Distancia entre la cara activa del contracarril y el carril del hilo opuesto de su vía **[<1620]**		1620,0
Anchura de la garganta de guía/desv	39	Distancia entre los extremos de la garganta de guía **[>38]**		38,0
Profundidad de la garganta de guía/desv	40	Se mide en el mismo punto que en la anchura de la garganta de guía **[>40]**		40,0
Sobreelevación de contracarriles/desv	41	El contracarril se sitúa más alto que el carril para el mejor guiado de los ejes de los vehículos **[15,25]**		20,0

Tabla 3. Tabla resumen del vector de datos de 29 variables

TABLA DE VARIABLES ACOTADAS				
VARIABLES	**IDENTIFICADOR**	**DEFINICIÓN [mm]**	**VALOR**	**REGLAS TOL**
Ancho/5/direc	1	Ancho de vía directa 5 traviesas antes de la JCA **[1666,1678]**		1668,0
Ancho/15cm/direc	2	Ancho de vía directa 15cms antes de la JCA **[1666,1678]**		1668,0
Ancho/paso-libre/direc	3	Ancho de vía directa en el punto donde se mide el paso libre de rueda en el cambio. **[1666,1678]**		1668,0
Ancho/cota de protección/direc	4	Ancho de vía directa en el punto donde se mide la cota de protección y la entrecalle carril-contracarril.**[1666,1678]**		1668,0
Ancho/5/desv	5	Ancho de vía desviada 5 traviesas antes de la JCA **[1666,1678]**		1668,0
Ancho/15cm/desv	6	Ancho de vía desviada 15cms antes de la JCA **[1666,1678]**		1668,0
Ancho/paso-libre/desv	7	Ancho de vía desviada en el punto donde se mide el paso libre de rueda en el cambio. **[1666,1678]**		1668,0
Ancho/cota de protección/desv	8	Ancho de vía desviada en el punto donde se mide la cota de protección y la entrecalle carril-contracarril.**[1666,1678]**		1668,0
Entrecalle mínima/direc	9	Se medirá en el punto más próximo entre la aguja y la contraaguja **[>58]**		58,0
Paso libre de rueda en el cambio/direc	10	Resultante de la resta entre el ancho(3) y la entrecalle mínima(9). **[<1618]**		1618,0
Cota de protección/direc	11	Distancia que debe existir entre la cara activa del contracarril y la punta del corazón. **Ancho Ibérico**: **[1626,1631]**		1628,0
Entrecalle carril-contracarril/direc	12	Se mide en el mismo punto que la cota de protección **[38,45]**		40,0
Entrecalle mínima/desv	13	Se medirá en el punto más próximo entre la aguja y la contraaguja **[>58]**		58,0
Paso libre de rueda en el cambio/desv	14	Resultante de la resta entre el ancho y la entrecalle mínima. **[<1618]**		1618,0
Cota de protección/desv	15	Distancia que debe existir entre la cara activa del contracarril y la punta del corazón. **Ancho Ibérico**: **[1626,1631]**		1628,0
Entrecalle carril-contracarril/desv	16	Se mide en el mismo punto que la cota de protección **[38,45]**		40,0
Paso en la punta del cruzamiento/direc	17	Distancia entre el contracarril y la cara vertical de la pata de liebre a la altura de la punta del corazón **[<1590]**		1590,0
Paso en la entrada de pata de liebre/direc	18	Distancia entre la cara activa de la pata de liebre y la cara activa del carril opuesto **[<1620]**		1620,0
Paso en la entrada de contracarril/direc	19	Distancia entre la cara activa del contracarril y el carril del hilo opuesto de su vía **[<1620]**		1620,0
Anchura de la garganta de guía/direc	20	Distancia entre los extremos de la garganta de guía **[>38]**		38,0
Profundidad de la garganta de guía/direc	21	Se mide en el mismo punto que en la anchura de la garganta de guía **[>40]**		40,0
Sobreelevación de contracarriles/direc	22	El contracarril se sitúa más alto que el carril para el mejor guiado de los ejes de los vehículos **[15,25]**		20,0
Descuadre de la punta de las agujas/direc	23	Mide el ajuste de la punta de las agujas con el vehículo **[-2,2]**		0,0
Paso en la punta del cruzamiento/desv	24	Distancia entre el contracarril y la cara vertical de la pata de liebre a la altura de la punta del corazón **[<1590]**		1590,0
Paso en la entrada de pata de liebre/desv	25	Distancia entre la cara activa de la pata de liebre y la cara activa del carril opuesto **[<1620]**		1620,0
Paso en la entrada de contracarril/desv	26	Distancia entre la cara activa del contracarril y el carril del hilo opuesto de su vía **[<1620]**		1620,0
Anchura de la garganta de guía/desv	27	Distancia entre los extremos de la garganta de guía **[>38]**		38,0
Profundidad de la garganta de guía/desv	28	Se mide en el mismo punto que en la anchura de la garganta de guía **[>40]**		40,0
Sobreelevación de contracarriles/desv	29	El contracarril se sitúa más alto que el carril para el mejor guiado de los ejes de los vehículos **[15,25]**		20,0

TABLA DE 12 VARIABLES				
VARIABLES	**IDENTIFICADOR**	**DEFINICIÓN [mm]**	**VALOR**	**REGLAS TOL**
Ancho/paso-libre/direc	0	Ancho de vía directa en el punto donde se mide el paso libre de rueda en el cambio. **[1666,1678]**		1668,0
Ancho/paso-libre/desv	1	Ancho de vía desviada en el punto donde se mide el paso libre de rueda en el cambio. **[1666,1678]**		1668,0
Ancho/cota de protección/direc	2	Ancho de vía directa en el punto donde se mide la cota de protección y la entrecalle carril-contracarril.**[1666,1678]**		1668,0
Ancho/cota de protección/desv	3	Ancho de vía desviada en el punto donde se mide la cota de protección y la entrecalle carril-contracarril.**[1666,1678]**		1668,0
Entrecalle mínima/direc	4	Se medirá en el punto más próximo entre la aguja y la contraaguja **[>58]**		58,0
Paso libre de rueda en el cambio/direc	5	Resultante de la resta entre el ancho(3) y la entrecalle mínima(9). **[<1618]**		1618,0
Cota de protección/direc	6	Distancia que debe existir entre la cara activa del contracarril y la punta del corazón. **Ancho Ibérico**: **[1626,1631]**		1628,0
Entrecalle carril-contracarril/direc	7	Se mide en el mismo punto que la cota de protección **[38,45]**		40,0
Entrecalle mínima/desv	8	Se medirá en el punto más próximo entre la aguja y la contraaguja **[>58]**		58,0
Paso libre de rueda en el cambio/desv	9	Resultante de la resta entre el ancho y la entrecalle mínima. **[<1618]**		1618,0
Cota de protección/desv	10	Distancia que debe existir entre la cara activa del contracarril y la punta del corazón. **Ancho Ibérico**: **[1626,1631]**		1628,0
Entrecalle carril-contracarril/desv	11	Se mide en el mismo punto que la cota de protección **[38,45]**		40,0

Variables Vía Directa

Variables Vía Desviada

Tabla 4. Tabla resumen del vector de datos de 12 variables

TABLA DE VARIABLES				
VARIABLES	IDENTIFICADOR	DEFINICIÓN [mm]	VALOR	REGLAS TOL
Ancho/paso-libre/direc	0	Ancho de vía directa en el punto donde se mide el paso libre de rueda en el cambio. **[1666,1678]**		1668,0
Entrecalle mínima/direc	1	Se medirá en el punto más próximo entre la aguja y la contraaguja **[>58]**		58,0
Paso libre de rueda en el cambio/direc	2	Resultante de la resta entre el ancho(3) y la entrecalle mínima(9). **[<1618]**		1618,0
Ancho/cota de protección/direc	3	Ancho de vía directa en el punto donde se mide la cota de protección y la entrecalle carril-contracarril.**[1666,1678]**		1668,0
Cota de protección/direc	4	Distancia que debe existir entre la cara activa del contracarril y la punta del corazón. **Ancho Ibérico: [1626,1631]**		1628,0
Entrecalle carril-contracarril/direc	5	Se mide en el mismo punto que la cota de protección **[38,45]**		40,0

Tabla 5. Tabla resumen del vector de datos de 6 variables

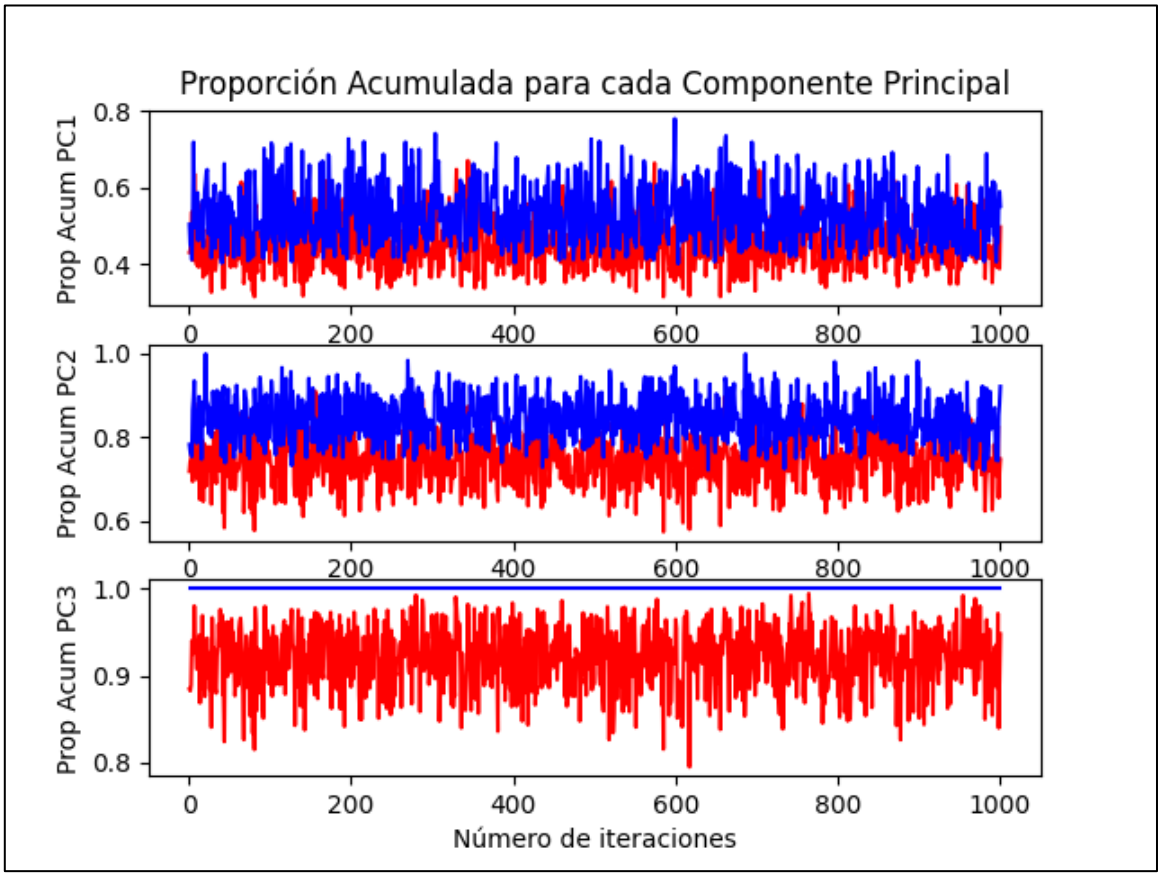


Figura 29. Prueba de validación de la variable "entrecalle mínima"

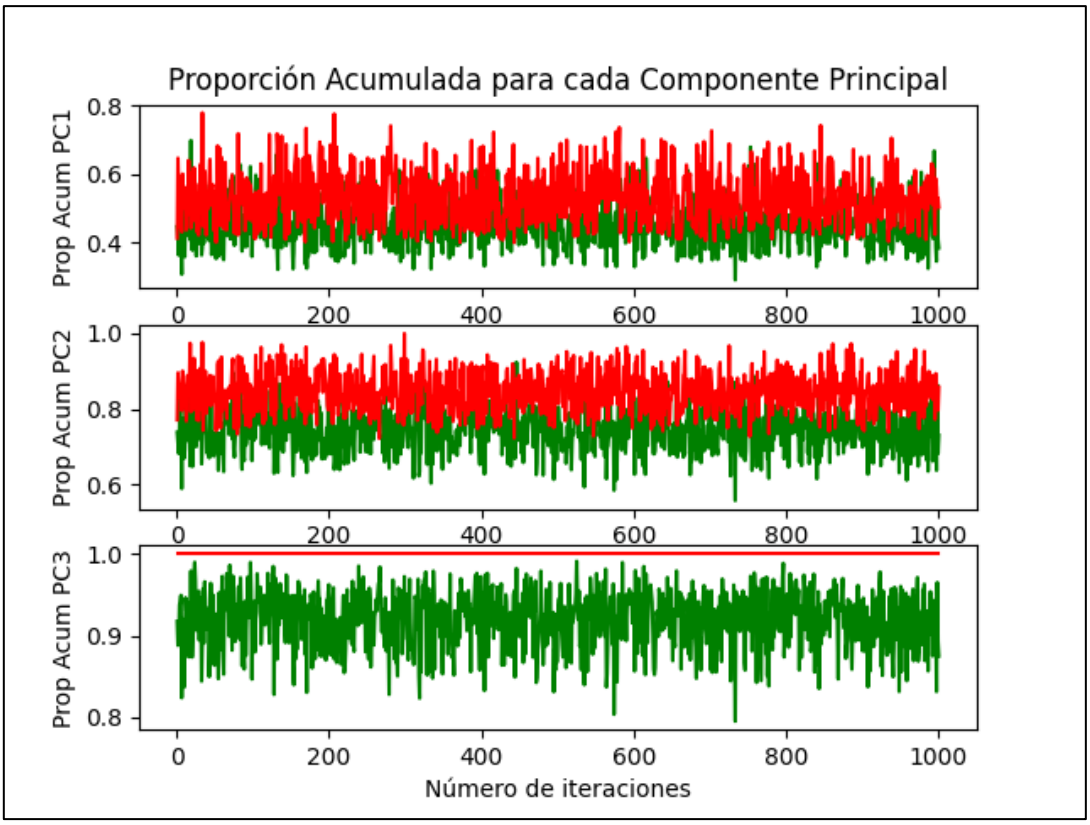


Figura 30. Prueba de validación de la variable "paso libre de rueda en el cambio"

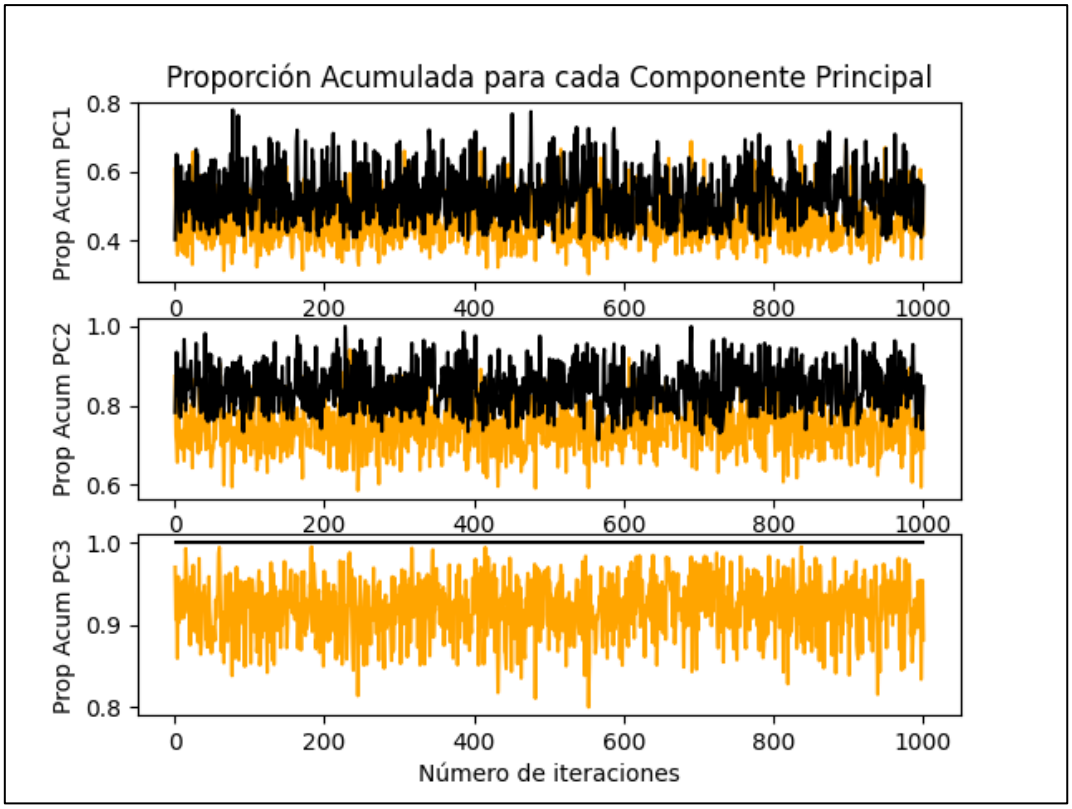


Figura 31. Prueba de validación de la variable "entrecalle carril-contracarril"

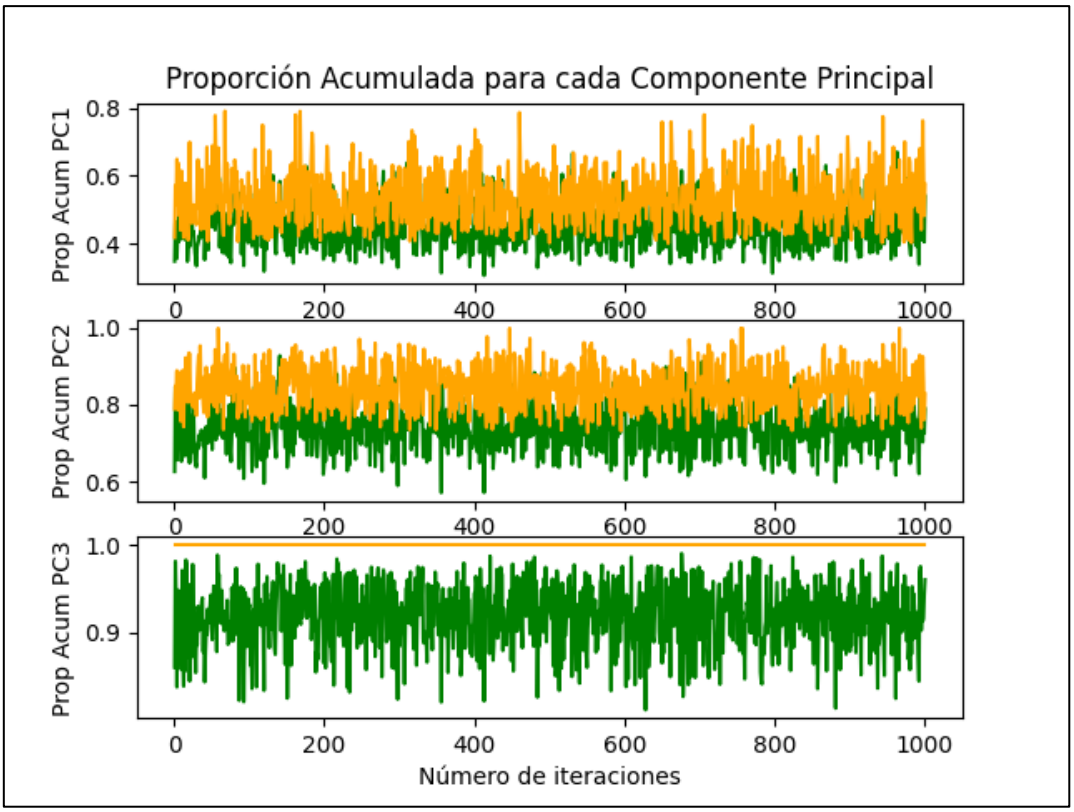


Figura 32. Prueba de validación de la variable "cota de protección"

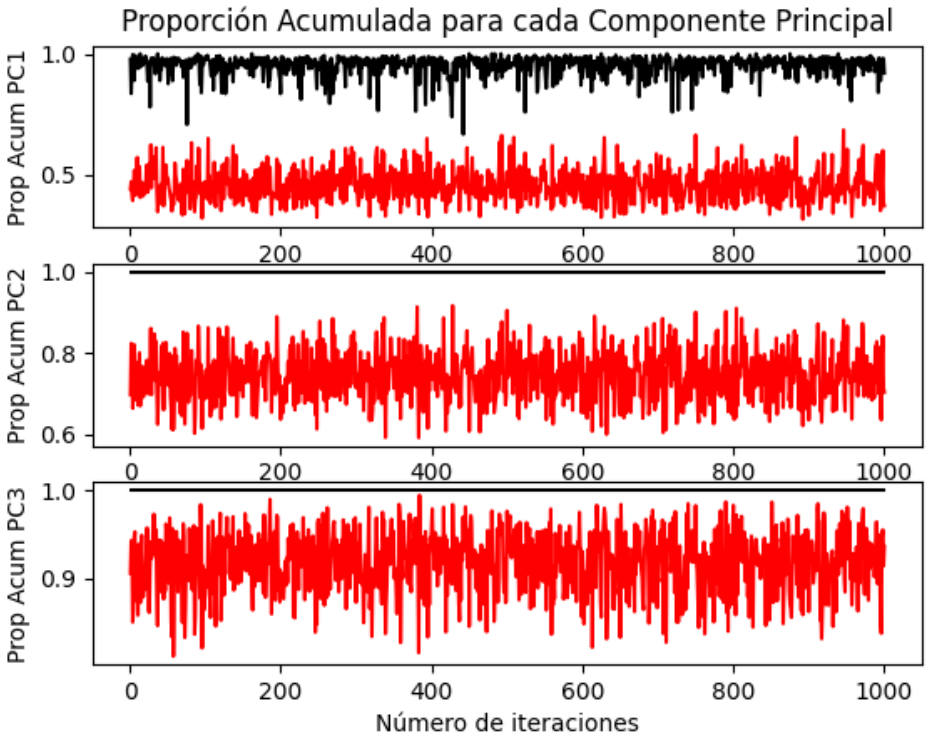


Figura 33. Prueba de validación de dos variables de zonas distinta de la estructura

ANEXO II

SCRIPT 1. Generación de datos sintéticos para el vector de datos de 29 Variables

```python
# LIBRERÍAS
import numpy as np
import pandas as pd

# nt MUESTRAS y nc VARIABLES
nt=100
nc=29

# Inicialización matriz de datos
matriz=(np.random.randint(0,1,(nc,nt)))

#Anchos de vía
for i in range(8):
    if (i!=3) and (i!=7):
        matriz[i]=np.random.randint(1666,1679,nt)

#Entrecalle mínima y Paso libre de rueda en el cambio
for j in range(nt):
    aux=np.random.randint(58,69,1)
    aux2=np.random.randint(58,69,1)
    while (matriz[2][j]-aux) > 1617 :
        aux = np.random.randint(58, 69, 1)
    matriz[8][j]=aux              #Entrecalle mínima DIREC
    matriz[9][j]=matriz[2][j]-aux #Paso libre de rueda en el cambio DIREC
    while (matriz[6][j]-aux2) > 1617 :
        aux2 = np.random.randint(58, 69, 1)
    matriz[12][j]=aux2              #Entrecalle mínima DESV
    matriz[13][j]=matriz[6][j]-aux2 #Paso libre de rueda en el cambio DESV
```

```python
#Cota de protección, Entrecalle carril-contracarril y Anchos de Vía
matriz[10]=np.random.randint(1626,1632,nt) #Cota de protección DIREC
matriz[14]=np.random.randint(1626,1632,nt) #Cota de protección DESV
for j in range(nt):
    aux=np.random.randint(38,46,1)
    aux2=np.random.randint(38,46,1)
    while ((matriz[10][j]+aux) < 1666):
        aux=np.random.randint(38,46,1)
    matriz[11][j]=aux              #Entrecalle carril-contracarril DIREC
    matriz[3][j]=matriz[10][j]+aux #Ancho de vía DIREC
    while ((matriz[14][j]+aux2)<1666):
        aux2=np.random.randint(38,46,1)
    matriz[15][j]=aux2              #Entrecalle carril-contracarril DESV
    matriz[7][j]=matriz[14][j]+aux2 #Ancho de vía DESV
```

```
#Resto de variables
for j in range(nt):
    matriz[16][j]=np.random.randint(1570, 1590)
    matriz[17][j]=np.random.randint(1590, 1620)
    matriz[18][j]=np.random.randint(1590, 1620)
    matriz[19][j]=np.random.randint(39, 65)
    matriz[20][j]=np.random.randint(41, 49)
    matriz[21][j]=np.random.randint(15, 26)
    matriz[22][j]=np.random.randint(-2, 3)
    matriz[23][j]=np.random.randint(1570, 1590)
    matriz[24][j]=np.random.randint(1590, 1620)
    matriz[25][j]=np.random.randint(1590, 1620)
    matriz[26][j]=np.random.randint(39, 65)
    matriz[27][j]=np.random.randint(41, 49)
    matriz[28][j]=np.random.randint(15, 26)
```

```
#Verificación
for j in range(nc):
    print(j,min(matriz[j]),max(matriz[j]))

print(matriz)

#Trasponemos para trabajar con la matriz de datos (NTxNC)
data=matriz.T

# Pasar matriz de datos a DataFrame
dfaux = pd.DataFrame(data=data)

# Añadir nombres a las columnas
indices=[]
for i in range(1,nc+1):
    indices.append([])
    index='Var'+str(i)
    indices[i-1]=index

df=dfaux.set_axis(indices, axis=1)

#VISUALIZACIÓN DEL DATA FRAME
print("\n")
print(df)
```

SCRIPT 2. Implementación de método PCA a 29 variables

```
# LIBRERÍAS
import numpy as np
import pandas as pd
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
import warnings
warnings.filterwarnings('ignore')

# nt MUESTRAS y nc VARIABLES
nt=100
nc=29

# Inicialización matriz de datos
matriz=(np.random.randint(0,1,(nc,nt)))

#Anchos de vía
for i in range(8):
    if (i!=3) and (i!=7):
        matriz[i]=np.random.randint(1666,1679,nt)

#Entrecalle mínima y Paso libre de rueda en el cambio
for j in range(nt):
    aux=np.random.randint(58,69,1)
    aux2=np.random.randint(58,69,1)
    while (matriz[2][j]-aux) > 1617 :
        aux = np.random.randint(58, 69, 1)
    matriz[8][j]=aux              #Entrecalle mínima DIREC
    matriz[9][j]=matriz[2][j]-aux #Paso libre de rueda en el cambio DIREC
```

```
    while (matriz[6][j]-aux2) > 1617 :
        aux2 = np.random.randint(58, 69, 1)
    matriz[12][j]=aux2              #Entrecalle mínima DESV
    matriz[13][j]=matriz[6][j]-aux2 #Paso libre de rueda en el cambio DESV

#Cota de protección, Entrecalle carril-contracarril y Anchos de Vía
matriz[10]=np.random.randint(1626,1632,nt) #Cota de protección DIREC
matriz[14]=np.random.randint(1626,1632,nt) #Cota de protección DESV
for j in range(nt):
    aux=np.random.randint(38,46,1)
    aux2=np.random.randint(38,46,1)
    while ((matriz[10][j]+aux) < 1666):
        aux=np.random.randint(38,46,1)
    matriz[11][j]=aux              #Entrecalle carril-contracarril DIREC
    matriz[3][j]=matriz[10][j]+aux #Ancho de vía DIREC
    while ((matriz[14][j]+aux2)<1666):
        aux2=np.random.randint(38,46,1)
    matriz[15][j]=aux2              #Entrecalle carril-contracarril DESV
    matriz[7][j]=matriz[14][j]+aux2 #Ancho de vía DESV
```

```
#Resto de variables
for j in range(nt):
    matriz[16][j]=np.random.randint(1570, 1590)
    matriz[17][j]=np.random.randint(1590, 1620)
    matriz[18][j]=np.random.randint(1590, 1620)
    matriz[19][j]=np.random.randint(39, 65)
    matriz[20][j]=np.random.randint(41, 49)
    matriz[21][j]=np.random.randint(15, 26)
    matriz[22][j]=np.random.randint(-2, 3)
```

```
    matriz[23][j]=np.random.randint(1570, 1590)
    matriz[24][j]=np.random.randint(1590, 1620)
    matriz[25][j]=np.random.randint(1590, 1620)
    matriz[26][j]=np.random.randint(39, 65)
    matriz[27][j]=np.random.randint(41, 49)
    matriz[28][j]=np.random.randint(15, 26)

#Verificación
for j in range(nc):
    print(j,min(matriz[j]),max(matriz[j]))

print(matriz)

#Trasponemos para trabajar con la matriz de datos (NTxNC)
data=matriz.T

# Pasar matriz de datos a DataFrame
dfaux = pd.DataFrame(data=data)

# Añadir nombres a las columnas
indices=[]
for i in range(1,nc+1):
    indices.append([])
    index='Var'+str(i)
    indices[i-1]=index

df=dfaux.set_axis(indices, axis=1)
#print(df)
```

```
# PCA
# Entrenamiento modelo PCA con escalado de los datos
estandar = pd.DataFrame(StandardScaler().fit_transform(data))
modelo_pca=PCA()
datos=modelo_pca.fit_transform(estandar)
# Mostrar varianzas
PCS=[]
for i in range(1,nc+1):
    PCS.append([])
    PCx='PC'+str(i)
    PCS[i-1]=PCx
varianzas=modelo_pca.explained_variance_ratio_
auxiliar = 0
VarianzasAcum=[]
for i in range(nc):
    VarianzasAcum.append([])
    VarianzasAcum[i] = auxiliar + varianzas[i]
    auxiliar = VarianzasAcum[i]

# PARTICIÓN DEL DATA FRAME
varianzas2=[]
PCS2=[]
varianzasacum2=[]
for i in range (15):
    varianzas2.append([])
    varianzasacum2.append([])
    PCS2.append([])
    PCS2[i]=PCS[i]
    varianzas2[i]=varianzas[i]
    varianzasacum2[i] = VarianzasAcum[i]
```

```
FV2=pd.DataFrame(varianzas2, columns=['PROP. VARIANZA'], index=PCS2)
FV2['PROPORCIÓN_ACUM']=varianzasacum2

varianzas3=[]
PCS3=[]
varianzasacum3=[]
for i in range (15,29):
    varianzas3.append([])
    varianzasacum3.append([])
    PCS3.append([])
    PCS3[i-15]=PCS[i]
    varianzas3[i-15]=varianzas[i]
    varianzasacum3[i-15] = VarianzasAcum[i]
FV3=pd.DataFrame(varianzas3, columns=['PROP. VARIANZA'], index=PCS3)
FV3['PROPORCIÓN_ACUM']=varianzasacum3

print("\n")
print(FV2)
print("\n")
print(FV3)
print("\n")
```

SCRIPT 3. Implementación de método PCA a 12 variables

```
# LIBRERÍAS
import numpy as np
import pandas as pd
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
import seaborn as sns
from IPython.display import display
import warnings
warnings.filterwarnings('ignore')

# nt DATASETS y nc VARIABLES
nt=100
nc=12

#Inicialización matriz de datos
matriz=(np.random.randint(0,1,(nc,nt)))

#Anchos de vía
for i in range(2):
    matriz[i]=np.random.randint(1666,1679,nt)

#Entrecalle mínima y Paso libre de rueda en el cambio
for j in range(nt):
    aux=np.random.randint(58,69,1)
    aux2=np.random.randint(58,69,1)
    while (matriz[0][j]-aux) > 1617 :
        aux = np.random.randint(58, 69, 1)
    matriz[4][j]=aux              #Entrecalle mínima DIREC
    matriz[5][j]=matriz[0][j]-aux #Paso libre de rueda en el cambio DIREC (LD)
    while (matriz[1][j]-aux2) > 1617 :
        aux2 = np.random.randint(58, 69, 1)
    matriz[8][j]=aux2              #Entrecalle mínima DESV
    matriz[9][j]=matriz[1][j]-aux2 #Paso libre de rueda en el cambio DESV (LD)

#Cota de protección, Entrecalle carril-contracarril y Anchos de Vía
matriz[6]=np.random.randint(1626,1632,nt) #Cota de protección DIREC
matriz[10]=np.random.randint(1626,1632,nt) #Cota de protección DESV
for j in range(nt):
    aux=np.random.randint(38,46,1)
    aux2=np.random.randint(38,46,1)
    while ((matriz[6][j]+aux) < 1666):
        aux=np.random.randint(38,46,1)
    matriz[7][j]=aux              #Entrecalle carril-contracarril DIREC
    matriz[2][j]=matriz[6][j]+aux #Ancho de vía DIREC (LD)
    while ((matriz[10][j]+aux2)<1666):
        aux2=np.random.randint(38,46,1)
    matriz[11][j]=aux2             #Entrecalle carril-contracarril DESV
    matriz[3][j]=matriz[10][j]+aux2 #Ancho de vía DESV (LD)
```

```
#Verificación
print("VERIFICACIÓN\n" )
for j in range(nc):
    print(j,min(matriz[j]),max(matriz[j]))

print("\n")
#Trasponemos para trabajar con la matriz de datos (NTxNC)
data=matriz.T

# Pasar matriz de datos a DataFrame
dfaux = pd.DataFrame(data=data)

# Añadir nombres a las columnas
indices=[]
for i in range(1,nc+1):
    indices.append([])
    index='Var'+str(i)
    indices[i-1]=index

df=dfaux.set_axis(indices, axis=1)

# MOSTRAR DATAFRAME
#print(df)
```

```
# PCA
# Entrenamiento modelo PCA con escalado de los datos
estandar = pd.DataFrame(StandardScaler().fit_transform(data))
modelo_pca=PCA()
datos=modelo_pca.fit_transform(estandar)
# Mostrar varianzas
PCS=[]
for i in range(1,nc+1):
    PCS.append([])
    PCx='PC'+str(i)
    PCS[i-1]=PCx
varianzas=modelo_pca.explained_variance_ratio_
auxiliar = 0
VarianzasAcum=[]
for i in range(nc):

    VarianzasAcum.append([])
    VarianzasAcum[i] = round((auxiliar + varianzas[i]),4)
    auxiliar = VarianzasAcum[i]
    varianzas[i] = round(varianzas[i], 4)

FV=pd.DataFrame(varianzas, columns=['PROP. VARIANZA'], index=PCS)
FV['PROPORCIÓN_ACUM']=VarianzasAcum
```

```
# MOSTRAR VARIANZAS
print(FV)

# AÑADIR COLUMNA PCs
ejex = []
ejex = np.arange(1, nc+1)
FV.insert(0, "PCs", ejex, True)

# DIAGRAMA DE BARRAS VARIANZAS
ejex = []
ejex = np.arange(1, nc+1)

fig, (ax1, ax2) = plt.subplots(1, 2)

ax1.bar(ejex, varianzas,color='skyblue', edgecolor='black')
ax1.set_title('Diagrama de las Proporciones de Varianza')
ax1.set(xlabel='Componentes Principales',ylabel='Proporciones de Varianza')
ax2.bar(ejex, VarianzasAcum,color='skyblue', edgecolor='black')
ax2.set_title('Diagrama de Proporciones Acumuladas')
ax2.set(xlabel='Componentes Principales',ylabel='Proporción Acumuladas')
plt.show()
```

SCRIPT 4. Implementación de método PCA a varias pruebas con 6 variables

```python
# LIBRERÍAS
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from IPython.display import display
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
import warnings
warnings.filterwarnings('ignore')

# nt DATASETS, nc VARIABLES y X MUESTRAS
nt = 100
nc = 6

# DATA SET ALEATORIO
print("DATA SET ALEATORIO\n")

# Inicialización matriz de datos aleatorios
matriz = (np.random.randint(0, 1, (nc, nt)))

# Entrecalle mínima y Paso libre de rueda en el cambio
for j in range(nt):
    matriz[0][j] = np.random.randint(1666, 1679, 1)  # Ancho de vía paso libre
    aux = np.random.randint(58, 69, 1)
    while (matriz[0][j] - aux) > 1617:
        aux = np.random.randint(58, 69, 1)
    matriz[1][j] = aux  # Entrecalle mínima
```

```python
    matriz[2][j] = matriz[0][j] - aux  # Paso libre de rueda en el cambio (LD)

# Cota de protección, Entrecalle carril-contracarril y Anchos de Vía

for j in range(nt):
    matriz[4][j] = np.random.randint(1626, 1632, 1)  # Cota de protección DIREC
    aux = np.random.randint(38, 46, 1)
    while ((matriz[4][j] + aux) < 1666):
        aux = np.random.randint(38, 46, 1)
    matriz[5][j] = aux  # Entrecalle carril-contracarril DIREC
    matriz[3][j] = matriz[4][j] + aux  # Ancho de vía DIREC (LD)

#Trasponemos para trabajar con la matriz de datos (NTxNC)
data=matriz.T

# Pasar matriz de datos a DataFrame
df = pd.DataFrame(data)

print("DATA SET ALEATORIO\n")
print(df)
print("\n")

# PCA
# Entrenamiento modelo PCA con escalado de los datos
estandar = pd.DataFrame(StandardScaler().fit_transform(data))
modelo_pca=PCA()
datos=modelo_pca.fit_transform(estandar)
```

```
#ACUMULACIÓN DE VARIANZAS
varianzas=modelo_pca.explained_variance_ratio_
aux=0
VarianzasAcum=[]
for i in range(nc):
    VarianzasAcum.append([])
    VarianzasAcum[i]= aux+varianzas[i]
    aux=VarianzasAcum[i]


# PRUEBA 1: VALORES ANTERIORES DE ANCHO DE VÍA PASO LIBRE
print("PRUEBA 1: VALORES ANTERIORES DE ANCHO DE VÍA PASO LIBRE\n")

# Inicialización matriz de datos modificados
matriz2 = (np.random.randint(0, 1, (nc, nt)))
#matriz2=np.ones((nc,nt))
matriz2[0] = np.random.randint(1666, 1679, nt)  # Ancho de vía paso libre
for j in range(0,2):
    aux = np.random.randint(58, 69, 1)
    while (matriz2[0][j] - aux) > 1617:
        aux = np.random.randint(58, 69, 1)
    matriz2[1][j] = aux  # Entrecalle mínima
    matriz2[2][j] = matriz2[0][j] - aux  # Paso libre de rueda en el cambio (LD)
for j in range (2,nt):

    X=2
    matriz2[1][j] = matriz2[1][j-1] + X*((matriz2[0][j-1]-matriz2[0][j-2])/2)
    matriz2[2][j] = matriz2[0][j] - matriz2[1][j]
```

```
for j in range(nt):
    matriz2[4][j] = np.random.randint(1626, 1632, 1)  # Cota de protección DIREC
    aux = np.random.randint(38, 46, 1)
    while ((matriz2[4][j] + aux) < 1666):
        aux = np.random.randint(38, 46, 1)
    matriz2[5][j] = aux  # Entrecalle carril-contracarril DIREC
    matriz2[3][j] = matriz2[4][j] + aux  # Ancho de vía DIREC (LD)


#Trasponemos para trabajar con la matriz de datos (NTxNC)
data2=matriz2.T

# Pasar matriz de datos a DataFrame
df2 = pd.DataFrame(data2)

print("DATA FRAME PRUEBA_1\n")
print(df2)
print("\n")

# PCA PRUEBA 1
# Entrenamiento modelo PCA con escalado de los datos
estandar2 = pd.DataFrame(StandardScaler().fit_transform(data2))
modelo_pca2=PCA()
datos2=modelo_pca2.fit_transform(estandar2)

#ACUMULACIÓN DE VARIANZAS
varianzas2=modelo_pca2.explained_variance_ratio_
aux=0
VarianzasAcum2=[]
```

```
for i in range(nc):
    VarianzasAcum2.append([])
    VarianzasAcum2[i]= aux+varianzas2[i]
    aux=VarianzasAcum2[i]


# PRUEBA 2: VALORES ANTERIORES DE ANCHO DE VÍA PASO LIBRE
print("PRUEBA 2: VALORES ANTERIORES DE ANCHO DE VÍA PASO LIBRE\n")

# Inicialización matriz de datos modificados
matriz3 = (np.random.randint(0, 1, (nc, nt)))

for j in range(nt):
    matriz3[4][j] = np.random.randint(1626, 1632, 1)  # Cota de protección DIREC
    aux = np.random.randint(38, 46, 1)
    while ((matriz3[4][j] + aux) < 1666):
        aux = np.random.randint(38, 46, 1)
    matriz3[5][j] = aux  # Entrecalle carril-contracarril DIREC
    matriz3[3][j] = matriz3[4][j] + aux  # Ancho de vía DIREC (LD)

matriz3[0] = np.random.randint(1666, 1679, nt)  # Ancho de vía paso libre

for j in range(0,1):
    aux = np.random.randint(58, 69, 1)
    while (matriz3[0][j] - aux) > 1617:
        aux = np.random.randint(58, 69, 1)
    matriz3[1][j] = aux  # Entrecalle mínima
    matriz3[2][j] = matriz3[0][j] - aux  # Paso libre de rueda en el cambio (LD)
```

```
for j in range (1,nt):

    X=2.1
    matriz3[2][j] = matriz3[2][j-1] + X*((matriz3[3][j-1]-matriz3[3][j-2])/2)
    matriz3[1][j] = matriz3[0][j] - matriz3[2][j]
#Trasponemos para trabajar con la matriz de datos (NTxNC)
data3=matriz3.T

# Pasar matriz de datos a DataFrame
df3 = pd.DataFrame(data3)

print("DATA FRAME PRUEBA_2\n")
print(df3)
print("\n")

# PCA PRUEBA 2
# Entrenamiento modelo PCA con escalado de los datos
estandar3 = pd.DataFrame(StandardScaler().fit_transform(data3))
modelo_pca3=PCA()
datos3=modelo_pca3.fit_transform(estandar3)

#ACUMULACIÓN DE VARIANZAS
varianzas3=modelo_pca3.explained_variance_ratio_
aux=0
VarianzasAcum3=[]
for i in range(nc):
    VarianzasAcum3.append([])
    VarianzasAcum3[i]= aux+varianzas3[i]
    aux=VarianzasAcum3[i]
```

```
# PRUEBA 3: VALORES ANTERIORES DE ANCHO DE VÍA PASO LIBRE
print("PRUEBA 3: VALORES ANTERIORES DE ANCHO DE VÍA PASO LIBRE\n")

# Inicialización matriz de datos modificados
matriz4 = (np.random.randint(0, 1, (nc, nt)))

for j in range(nt):
    matriz4[4][j] = np.random.randint(1626, 1632, 1)  # Cota de protección DIREC
    aux = np.random.randint(38, 46, 1)
    while ((matriz4[4][j] + aux) < 1666):
        aux = np.random.randint(38, 46, 1)
    matriz4[5][j] = aux  # Entrecalle carril-contracarril DIREC
    matriz4[3][j] = matriz4[4][j] + aux  # Ancho de vía DIREC (LD)

matriz4[0] = np.random.randint(1666, 1679, nt)  # Ancho de vía paso libre
for j in range(0,1):
    aux = np.random.randint(58, 69, 1)
    while (matriz4[0][j] - aux) > 1617:
        aux = np.random.randint(58, 69, 1)
    matriz4[1][j] = aux  # Entrecalle mínima
    matriz4[2][j] = matriz4[0][j] - aux  # Paso libre de rueda en el cambio (LD)

X=2

for j in range(1,nt):
    matriz4[1][j] = matriz4[1][j-1] + X*((matriz4[5][j-1]-matriz4[5][j-2])/2)
    matriz4[2][j] = matriz4[0][j] - matriz4[1][j]
```

```
#Trasponemos para trabajar con la matriz de datos (NTxNC)
data4=matriz4.T

# Pasar matriz de datos a DataFrame
df4 = pd.DataFrame(data4)

print("DATA FRAME PRUEBA_3\n")
print(df4)
print("\n")

# PCA PRUEBA 3
# Entrenamiento modelo PCA con escalado de los datos
estandar4 = pd.DataFrame(StandardScaler().fit_transform(data4))
modelo_pca4=PCA()
datos4=modelo_pca4.fit_transform(estandar4)

#ACUMULACIÓN DE VARIANZAS
varianzas4=modelo_pca4.explained_variance_ratio_
aux=0
VarianzasAcum4=[]
for i in range(nc):
    VarianzasAcum4.append([])
    VarianzasAcum4[i]= aux+varianzas4[i]
    aux=VarianzasAcum4[i]
```

```
# GRÁFICAS
ejex = []
ejex = np.arange(1, 7)

plt.figure(0)
plt.plot(ejex,VarianzasAcum,'r', ejex, VarianzasAcum2,'b', ejex, VarianzasAcum3,'g', ejex,
VarianzasAcum4, 'black')
plt.ylim(0, 1.1)
plt.xlabel("Componentes Principales")
plt.ylabel("Proporción Acumulada")
plt.legend(["Dataset Aleatorio","Prueba 1","Prueba 2","Prueba3"], loc = "lower right")
plt.title("Análisis Gráfico")
plt.show()
```

SCRIPT 5. Análisis de la influencia del tamaño de la muestra en la precisión de la técnica PCA

```
# LIBRERÍAS
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from IPython.display import display
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
import warnings

warnings.filterwarnings('ignore')

# nc VARIABLES y X veces

nc = 6

Varianza1 = []
Varianza2 = []
Varianza3 = []
Varianza4 = []


for x in range(1000):

    Varianza1.append([])
    Varianza2.append([])
    Varianza3.append([])
    Varianza4.append([])
```

```
# Prueba 1: nt=10
nt = 10
matriz = (np.random.randint(0, 1, (nc, nt)))

# Entrecalle mínima y Paso libre de rueda en el cambio
for j in range(nt):
    matriz[0][j] = np.random.randint(1666, 1679, 1)  # Ancho de vía paso libre
    aux = np.random.randint(58, 69, 1)
    while (matriz[0][j] - aux) > 1617:
        aux = np.random.randint(58, 69, 1)
    matriz[1][j] = aux  # Entrecalle mínima
    matriz[2][j] = matriz[0][j] - aux  # Paso libre de rueda en el cambio (LD)

# Cota de protección, Entrecalle carril-contracarril y Anchos de Vía

for j in range(nt):
    matriz[4][j] = np.random.randint(1626, 1632, 1)  # Cota de protección DIREC
    aux = np.random.randint(38, 46, 1)
    while ((matriz[4][j] + aux) < 1666):
        aux = np.random.randint(38, 46, 1)
    matriz[5][j] = aux  # Entrecalle carril-contracarril DIREC
    matriz[3][j] = matriz[4][j] + aux  # Ancho de vía DIREC (LD)

# Trasponemos para trabajar con la matriz de datos (NTxNC)
data = matriz.T

# Pasar matriz de datos a DataFrame
df = pd.DataFrame(data)
```

```
# PCA
# Entrenamiento modelo PCA con escalado de los datos
estandar = pd.DataFrame(StandardScaler().fit_transform(data))
modelo_pca = PCA()
datos = modelo_pca.fit_transform(estandar)

# ACUMULACIÓN DE VARIANZAS
varianzas = modelo_pca.explained_variance_ratio_
aux = 0
VarianzasAcum = []
for i in range(nc):
    VarianzasAcum.append([])
    VarianzasAcum[i] = aux + varianzas[i]
    aux = VarianzasAcum[i]
Varianza1[x]=VarianzasAcum[2]
# PRUEBA 2: nt=100
nt = 100
# Inicialización matriz de datos modificados
matriz2 = (np.random.randint(0, 1, (nc, nt)))
# Entrecalle mínima y Paso libre de rueda en el cambio
for j in range(nt):
    matriz2[0][j] = np.random.randint(1666, 1679, 1)  # Ancho de vía paso libre
    aux = np.random.randint(58, 69, 1)
    while (matriz2[0][j] - aux) > 1617:
        aux = np.random.randint(58, 69, 1)
    matriz2[1][j] = aux  # Entrecalle mínima
    matriz2[2][j] = matriz2[0][j] - aux  # Paso libre de rueda en el cambio (LD)
```

```
# Cota de protección, Entrecalle carril-contracarril y Anchos de Vía

for j in range(nt):
    matriz2[4][j] = np.random.randint(1626, 1632, 1)  # Cota de protección DIREC
    aux = np.random.randint(38, 46, 1)
    while ((matriz2[4][j] + aux) < 1666):
        aux = np.random.randint(38, 46, 1)
    matriz2[5][j] = aux  # Entrecalle carril-contracarril DIREC
    matriz2[3][j] = matriz2[4][j] + aux  # Ancho de vía DIREC (LD)

# Trasponemos para trabajar con la matriz de datos (NTxNC)
data2 = matriz2.T

# Pasar matriz de datos a DataFrame
df2 = pd.DataFrame(data2)


# PCA PRUEBA 1
# Entrenamiento modelo PCA con escalado de los datos
estandar2 = pd.DataFrame(StandardScaler().fit_transform(data2))
modelo_pca2 = PCA()
datos2 = modelo_pca2.fit_transform(estandar2)

# ACUMULACIÓN DE VARIANZAS
varianzas2 = modelo_pca2.explained_variance_ratio_
aux = 0
VarianzasAcum2 = []
for i in range(nc):
    VarianzasAcum2.append([])
    VarianzasAcum2[i] = aux + varianzas2[i]
```

```
    aux = VarianzasAcum2[i]
Varianza2[x] = VarianzasAcum2[2]

# PRUEBA 3: nt=1000
nt = 1000
# Inicialización matriz de datos modificados
matriz3 = (np.random.randint(0, 1, (nc, nt)))
# Entrecalle mínima y Paso libre de rueda en el cambio
for j in range(nt):
    matriz3[0][j] = np.random.randint(1666, 1679, 1)  # Ancho de vía paso libre
    aux = np.random.randint(58, 69, 1)
    while (matriz3[0][j] - aux) > 1617:
        aux = np.random.randint(58, 69, 1)
    matriz3[1][j] = aux  # Entrecalle mínima
    matriz3[2][j] = matriz3[0][j] - aux  # Paso libre de rueda en el cambio (LD)

# Cota de protección, Entrecalle carril-contracarril y Anchos de Vía

for j in range(nt):
    matriz3[4][j] = np.random.randint(1626, 1632, 1)  # Cota de protección DIREC
    aux = np.random.randint(38, 46, 1)
    while ((matriz3[4][j] + aux) < 1666):
        aux = np.random.randint(38, 46, 1)
    matriz3[5][j] = aux  # Entrecalle carril-contracarril DIREC
    matriz3[3][j] = matriz3[4][j] + aux  # Ancho de vía DIREC (LD)

# Trasponemos para trabajar con la matriz de datos (NTxNC)
data3 = matriz3.T
```

```
# Pasar matriz de datos a DataFrame
df3 = pd.DataFrame(data3)

# PCA PRUEBA 2
# Entrenamiento modelo PCA con escalado de los datos
estandar3 = pd.DataFrame(StandardScaler().fit_transform(data3))
modelo_pca3 = PCA()
datos3 = modelo_pca3.fit_transform(estandar3)

# ACUMULACIÓN DE VARIANZAS
varianzas3 = modelo_pca3.explained_variance_ratio_
aux = 0
VarianzasAcum3 = []
for i in range(nc):
    VarianzasAcum3.append([])
    VarianzasAcum3[i] = aux + varianzas3[i]
    aux = VarianzasAcum3[i]
Varianza3[x] = VarianzasAcum3[2]

# PRUEBA 4: nt=10000
nt = 10000
# Inicialización matriz de datos modificados
matriz4 = (np.random.randint(0, 1, (nc, nt)))
# Entrecalle mínima y Paso libre de rueda en el cambio
for j in range(nt):
    matriz4[0][j] = np.random.randint(1666, 1679, 1)  # Ancho de vía paso libre
    aux = np.random.randint(58, 69, 1)
    while (matriz4[0][j] - aux) > 1617:
        aux = np.random.randint(58, 69, 1)
    matriz4[1][j] = aux  # Entrecalle mínima
```

```
    matriz4[2][j] = matriz4[0][j] - aux  # Paso libre de rueda en el cambio (LD)

# Cota de protección, Entrecalle carril-contracarril y Anchos de Vía

for j in range(nt):
    matriz4[4][j] = np.random.randint(1626, 1632, 1)  # Cota de protección DIREC
    aux = np.random.randint(38, 46, 1)
    while ((matriz4[4][j] + aux) < 1666):
        aux = np.random.randint(38, 46, 1)
    matriz4[5][j] = aux  # Entrecalle carril-contracarril DIREC
    matriz4[3][j] = matriz4[4][j] + aux  # Ancho de vía DIREC (LD)

# Trasponemos para trabajar con la matriz de datos (NTxNC)
data4 = matriz4.T

# Pasar matriz de datos a DataFrame
df4 = pd.DataFrame(data4)

# PCA PRUEBA 3
# Entrenamiento modelo PCA con escalado de los datos
estandar4 = pd.DataFrame(StandardScaler().fit_transform(data4))
modelo_pca4 = PCA()
datos4 = modelo_pca4.fit_transform(estandar4)

# ACUMULACIÓN DE VARIANZAS
varianzas4 = modelo_pca4.explained_variance_ratio_
aux = 0
VarianzasAcum4 = []
for i in range(nc):
    VarianzasAcum4.append([])
```

```
        VarianzasAcum4[i] = aux + varianzas4[i]
        aux = VarianzasAcum4[i]
    Varianza4[x] = VarianzasAcum4[2]

# GRÁFICAS
ejex = []
ejex = np.arange(1, 1001)

plt.figure(0)
plt.plot(ejex, Varianza1, 'r', ejex, Varianza2, 'b', ejex, Varianza3, 'g', ejex, Varianza4, 'black')
plt.ylim(0, 1.1)
plt.xlabel("Iteraciones")
plt.ylabel("Proporción Acumulada Componente 3")
plt.legend(["Nt=10", "Nt=100", "Nt=1000", "Nt=10000"], loc="lower right")
plt.title("Análisis Gráfico")
plt.show()
```

SCRIPT 6. Algoritmo de validación con datos sintéticos

```
# LIBRERÍAS
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from IPython.display import display
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
import warnings
warnings.filterwarnings('ignore')

# nt DATASETS, nc VARIABLES y X MUESTRAS
nt = 10
nc = 6
X = 1000

VarianzasAle1 = []
VarianzasAle2 = []
VarianzasAle3 = []
VarianzasMod1 = []
VarianzasMod2 = []
VarianzasMod3 = []
# Ciclo de X muestras
for y in range(X):

    VarianzasAle1.append([])
    VarianzasAle2.append([])
    VarianzasAle3.append([])
```

```
    VarianzasMod1.append([])
    VarianzasMod2.append([])
    VarianzasMod3.append([])

    # Inicialización matriz de datos aleatorios
    matriz = (np.random.randint(0, 1, (nc, nt)))

    # Entrecalle mínima y Paso libre de rueda en el cambio
    for j in range(nt):
        matriz[0][j] = np.random.randint(1666, 1679, 1)  # Ancho de vía paso libre
        aux = np.random.randint(58, 69, 1)
        while (matriz[0][j] - aux) > 1617:
            aux = np.random.randint(58, 69, 1)
        matriz[1][j] = aux  # Entrecalle mínima
        matriz[2][j] = matriz[0][j] - aux  # Paso libre de rueda en el cambio (LD)

    # Cota de protección, Entrecalle carril-contracarril y Anchos de Vía

    for j in range(nt):
        matriz[4][j] = np.random.randint(1626, 1632, 1)  # Cota de protección DIREC
        aux = np.random.randint(38, 46, 1)
        while ((matriz[4][j] + aux) < 1666):
            aux = np.random.randint(38, 46, 1)
        matriz[5][j] = aux  # Entrecalle carril-contracarril DIREC
        matriz[3][j] = matriz[4][j] + aux  # Ancho de vía DIREC (LD)

    # Prueba 1: Ancho de vía(entrecalle min) aumenta progresivamente
    matriz2 = (np.random.randint(0, 1, (nc, nt)))
```

```
    # Ancho de vía paso libre
    matriz2[0][0] = 1668
    aux = np.random.randint(58, 69, 1)
    while (matriz2[0][0] - aux) > 1617:
        aux = np.random.randint(58, 69, 1)
    matriz2[1][0] = aux  # Entrecalle mínima
    matriz2[2][0] = matriz2[0][0] - aux  # Paso libre de rueda en el cambio (LD)

    for j in range(1, nt):
        matriz2[0][j] = matriz2[0][j - 1] + np.random.randint(0, 2, 1)
        matriz2[1][j] = matriz2[0][j] - matriz2[0][j - 1] + matriz2[1][j - 1]
        matriz2[2][j] = matriz2[0][j] - matriz2[1][j]

    # Cota de protección, Entrecalle carril-contracarril y Anchos de Vía
    matriz2[4] = np.random.randint(1626, 1632, nt)  # Cota de protección DIREC
    for j in range(nt):
        aux = np.random.randint(38, 46, 1)
        while ((matriz2[4][j] + aux) < 1666):
            aux = np.random.randint(38, 46, 1)
        matriz2[5][j] = aux  # Entrecalle carril-contracarril DIREC
        matriz2[3][j] = matriz2[4][j] + aux  # Ancho de vía DIREC (LD)

    # Prueba 2: Entrecalle mínima aumenta progresivamente
    matriz3 = (np.random.randint(0, 1, (nc, nt)))

    # Entrecalle mínima aumenta de manera progresiva
    matriz3[1][0] = 60
    aux = np.random.randint(1666,1679, 1)
    while (aux - matriz3[1][0] ) > 1617:
        aux = np.random.randint(1666, 1679, 1)
```

```
    matriz3[0][0] = aux  # Ancho de vía
    matriz3[2][0] = aux - matriz3[1][0]   # Paso libre de rueda en el cambio (LD)

    for j in range(1, nt):
        matriz3[1][j] = matriz3[1][j - 1] + np.random.randint(0, 2, 1)
        matriz3[0][j] = matriz3[1][j] - matriz3[1][j - 1] + matriz3[0][j - 1]
        matriz3[2][j] = matriz3[0][j] - matriz3[1][j]

    # Cota de protección, Entrecalle carril-contracarril y Anchos de Vía
    matriz3[4] = np.random.randint(1626, 1632, nt)  # Cota de protección DIREC
    for j in range(nt):
        aux = np.random.randint(38, 46, 1)
        while ((matriz3[4][j] + aux) < 1666):
            aux = np.random.randint(38, 46, 1)
        matriz3[5][j] = aux  # Entrecalle carril-contracarril DIREC
        matriz3[3][j] = matriz3[4][j] + aux  # Ancho de vía DIREC (LD)

    # Prueba 3:Paso libre de rueda aumenta progresivamente debido al desgaste de la aguja
    matriz4 = (np.random.randint(0, 1, (nc, nt)))

    # Paso libre de rueda aumenta progresivamente debido al desgaste de la aguja
    matriz4[2][0] = 1614
    aux = np.random.randint(58, 69, 1)
    while (aux + matriz4[2][0]) < 1679:
        aux = np.random.randint(58, 69, 1)
    matriz4[1][0] = aux  # Entrecalle mínima (LD)
    matriz4[0][0] = aux + matriz4[2][0]  # Ancho de vía
```

```
    for j in range(1, nt):
        matriz4[2][j] = matriz4[2][j - 1] + np.random.randint(0, 2, 1)
        matriz4[1][j] = -(matriz4[2][j] - matriz4[2][j - 1]) + matriz4[1][j - 1]
        matriz4[0][j] = matriz4[1][j] + matriz4[2][j]

    # Cota de protección, Entrecalle carril-contracarril y Anchos de Vía
    matriz4[4] = np.random.randint(1626, 1632, nt)  # Cota de protección DIREC
    for j in range(nt):
        aux = np.random.randint(38, 46, 1)
        while ((matriz4[4][j] + aux) < 1666):
            aux = np.random.randint(38, 46, 1)
        matriz4[5][j] = aux  # Entrecalle carril-contracarril DIREC
        matriz4[3][j] = matriz4[4][j] + aux  # Ancho de vía DIREC (LD)

    # Prueba 4: Cota de protección aumenta de manera progresiva y entrecalle carril-contracarril
disminunye
    matriz5 = (np.random.randint(0, 1, (nc, nt)))
    matriz5[0] = np.random.randint(1666, 1679, nt)

    # Entrecalle mínima y Paso libre de rueda en el cambio
    for j in range(nt):
        aux = np.random.randint(58, 69, 1)
        while (matriz5[0][j] - aux) > 1617:
            aux = np.random.randint(58, 69, 1)
        matriz5[1][j] = aux  # Entrecalle mínima
        matriz5[2][j] = matriz5[0][j] - aux  # Paso libre de rueda en el cambio (LD)

    # Cota de protección, Entrecalle carril-contracarril y Ancho de Vía
    matriz5[4][0] = 1628  # Cota de protección DIREC
    aux = np.random.randint(38, 46, 1)
```

```
    while ((matriz5[4][0] + aux) < 1666):
        aux = np.random.randint(38, 46, 1)
    matriz5[5][0] = aux  # Entrecalle carril-contracarril DIREC
    matriz5[3][0] = matriz5[4][0] + matriz5[5][0]  # Ancho de vía DIREC (LD)

    for j in range(1, nt):
        matriz5[4][j] = matriz5[4][j - 1] + np.random.randint(0, 2, 1)
        matriz5[5][j] = matriz5[5][j - 1] - (matriz5[4][j] - matriz5[4][j - 1])
        matriz5[3][j] = matriz5[4][j] + matriz5[5][j]

    #Prueba 5: Entrecalle carril-contracarril aumenta progresivamente
    matriz6 = (np.random.randint(0, 1, (nc, nt)))
    matriz6[0] = np.random.randint(1666, 1679, nt)

    # Entrecalle mínima y Paso libre de rueda en el cambio
    for j in range(nt):
        aux = np.random.randint(58, 69, 1)
        while (matriz6[0][j] - aux) > 1617:
            aux = np.random.randint(58, 69, 1)
        matriz6[1][j] = aux  # Entrecalle mínima
        matriz6[2][j] = matriz6[0][j] - aux  # Paso libre de rueda en el cambio (LD)

    # Cota de protección, Entrecalle carril-contracarril y Ancho de Vía
    matriz6[4][0] = 1628  # Cota de protección DIREC
    aux = np.random.randint(38, 46, 1)
    while ((matriz6[4][0] + aux) < 1666):
        aux = np.random.randint(38, 46, 1)
    matriz6[5][0] = aux  # Entrecalle carril-contracarril DIREC
    matriz6[3][0] = matriz6[4][0] + matriz6[5][0]  # Ancho de vía DIREC (LD)
```

```
for j in range(1, nt):
    matriz6[5][j] = matriz6[5][j - 1] + np.random.randint(0, 2, 1)
    matriz6[4][j] = matriz6[4][j - 1] - (matriz6[5][j] - matriz6[5][j - 1])
    matriz6[3][j] = matriz6[4][j] + matriz6[5][j]


#Prueba 6: Paso libre y entrecalle carril-contracarril aumentan progresivamente
matriz7 = (np.random.randint(0, 1, (nc, nt)))
#Paso libre aumenta y entrecalle min disminuye
matriz7[2][0] = 1614
aux = np.random.randint(58, 69, 1)
while (aux + matriz7[2][0]) < 1679:
    aux = np.random.randint(58, 69, 1)
matriz7[1][0] = aux  # Entrecalle mínima (LD)
matriz7[0][0] = aux + matriz7[2][0]  # Ancho de vía

for j in range(1, nt):
    matriz7[2][j] = matriz7[2][j - 1] + np.random.randint(0, 2, 1)
    matriz7[1][j] = -(matriz7[2][j] - matriz7[2][j - 1]) + matriz7[1][j - 1]
    matriz7[0][j] = matriz7[1][j] + matriz7[2][j]

# Cota de protección, Entrecalle carril-contracarril y Ancho de Vía
matriz7[4][0] = 1628  # Cota de protección DIREC
aux = np.random.randint(38, 46, 1)
while ((matriz7[4][0] + aux) < 1666):
    aux = np.random.randint(38, 46, 1)
matriz7[5][0] = aux  # Entrecalle carril-contracarril DIREC
matriz7[3][0] = matriz7[4][0] + matriz7[5][0]  # Ancho de vía DIREC (LD)
```

```
for j in range(1, nt):
    matriz7[5][j] = matriz7[5][j - 1] + np.random.randint(0, 2, 1)
    matriz7[3][j] = matriz7[3][j - 1] + (matriz7[5][j] - matriz7[5][j - 1])
    matriz7[4][j] = matriz7[3][j] - matriz7[5][j]

# IMPLEMENTACIÓN PCA
# Trasponemos para trabajar con la matriz de datos (NTxNC)
data = matriz.T

# Pasar matriz de datos a DataFrame
df = pd.DataFrame(data)

# PCA
# Entrenamiento modelo PCA con escalado de los datos
estandar = pd.DataFrame(StandardScaler().fit_transform(data))
modelo_pca = PCA()
datos = modelo_pca.fit_transform(estandar)

# ACUMULACIÓN DE Proporciones
vector_varianza = modelo_pca.explained_variance_ratio_
aux = 0
for i in range(nc):
    vector_varianza[i] = aux + vector_varianza[i]
    aux = vector_varianza[i]
VarianzasAle1[y].append(vector_varianza[0])
VarianzasAle2[y].append(vector_varianza[1])
VarianzasAle3[y].append(vector_varianza[2])

# Trasponemos para trabajar con la matriz de datos (NTxNC)
data2 = matriz7.T
```

```
    # Pasar matriz de datos a DataFrame
    df2 = pd.DataFrame(data2)

    # PCA
    # Entrenamiento modelo PCA con escalado de los datos
    estandar2 = pd.DataFrame(StandardScaler().fit_transform(data2))
    modelo_pca2 = PCA()
    datos2 = modelo_pca2.fit_transform(estandar2)

    # ACUMULACIÓN DE Proporciones
    vector_varianza2 = modelo_pca2.explained_variance_ratio_
    auxiliar = 0
    for i in range(nc):
        vector_varianza2[i] = auxiliar + vector_varianza2[i]
        auxiliar = vector_varianza2[i]
    VarianzasMod1[y].append(vector_varianza2[0])
    VarianzasMod2[y].append(vector_varianza2[1])
    VarianzasMod3[y].append(vector_varianza2[2])


    if y >= X - 2:
        print("VERIFICACIÓN\n")
        print(f"Muestra ALEATORIA número: {y}\n")
        print(f"{df}\n")
        print(f"Vector de Proporciones Acumuladas:\n {vector_varianza}\n")
        print(f"Muestra MODIFICADA número: {y}\n")
        print(f"{df2}\n")
        print(f"Vector de Proporciones Acumuladas:\n {vector_varianza2}\n")
```

```
print("Mínimos y Máximos de la proporción acumulada hasta la PC3 para los Data Sets ALEATORIOS:")
print(min(VarianzasAle3), max(VarianzasAle3))
print("\nMínimos y Máximos de la proporción acumulada hasta la PC3 para los Data Sets MODIFICADOS:")
print(min(VarianzasMod3), max(VarianzasMod3))

# GRÁFICAS
# Gráfica de la varianza acumulada de la tercera componente principal
ejex=[]
ejex=np.arange(1, X+1)

# Subplots con las tres primeras Componentes Principales
fig, (ax1, ax2, ax3) = plt.subplots(3)

ax1.plot(ejex,VarianzasAle1, 'r',ejex,VarianzasMod1,'black')
ax1.set_title('Proporción Acumulada para cada Componente Principal')
ax1.set(ylabel='Prop Acum PC1')
ax2.plot(ejex,VarianzasAle2,'r',ejex,VarianzasMod2,'black')
ax2.set(ylabel='Prop Acum PC2')
ax3.plot(ejex,VarianzasAle3,'r',ejex,VarianzasMod3,'black')
ax3.set(xlabel='Número de iteraciones',ylabel='Prop Acum PC3')
plt.show()
```