




Inteligencia Artificial con Python y scikit-learn


[Install](#) [User Guide](#) [API](#) [Examples](#) [Community](#) [More](#)

scikit-learn

Machine Learning in Python

[Getting Started](#) [Release Highlights for 1.6](#)

- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

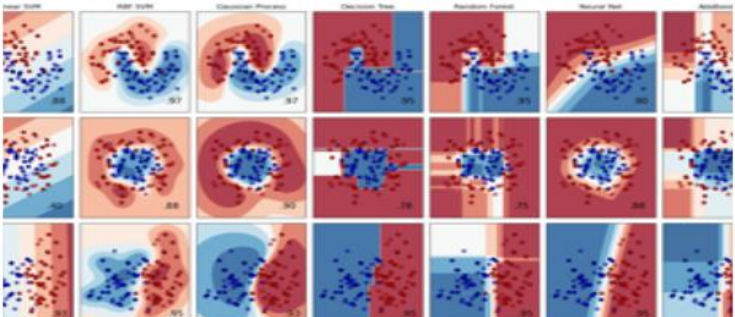


Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.

Algorithms: [Gradient boosting](#), [nearest neighbors](#), [random forest](#), [logistic regression](#), and [more...](#)

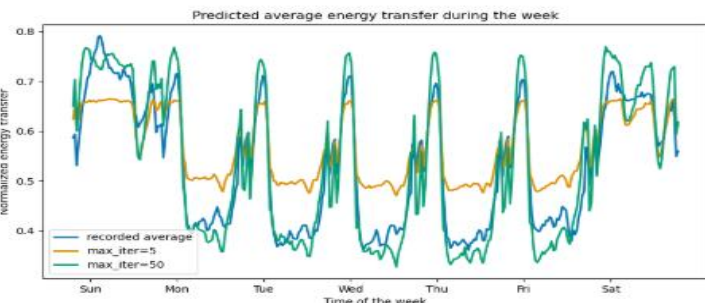


Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, stock prices.

Algorithms: [Gradient boosting](#), [nearest neighbors](#), [random forest](#), [ridge](#), and [more...](#)

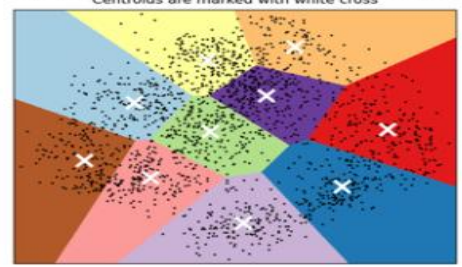


Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, grouping experiment outcomes.

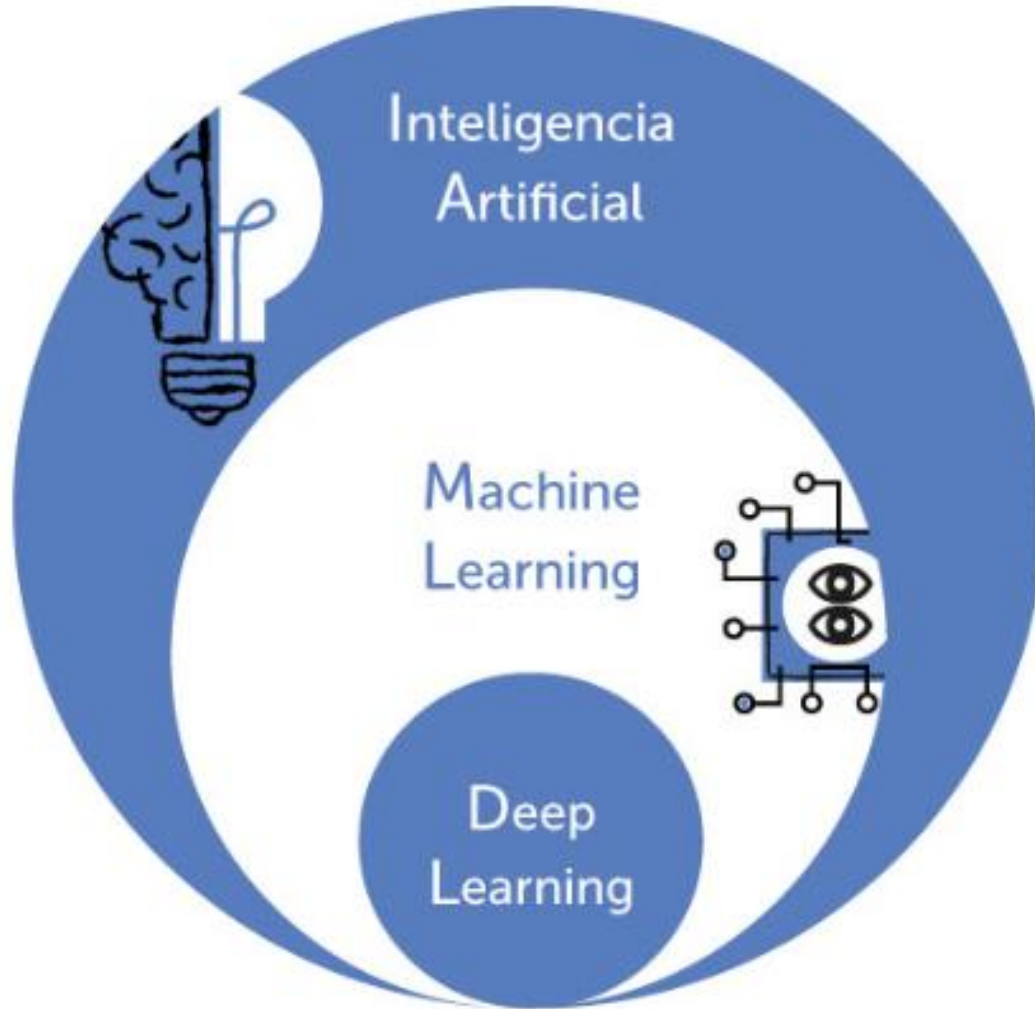
Algorithms: [k-Means](#), [HDBSCAN](#), [hierarchical clustering](#), and [more...](#)



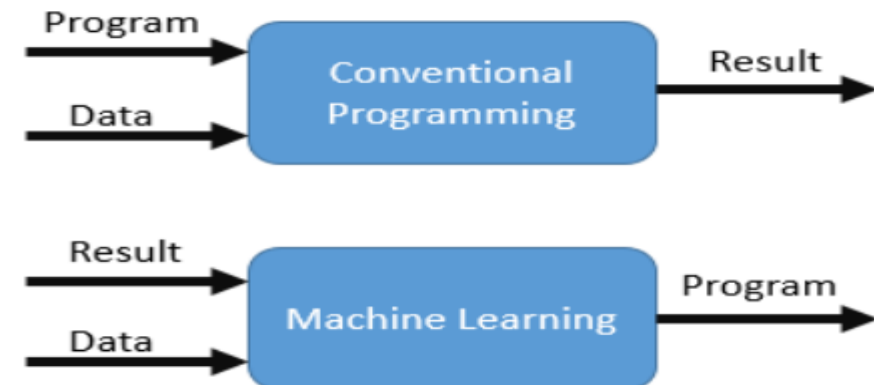
Librería scikit-learn: Aprendizaje automático con Python

- **scikit-learn Homepage**
<http://scikit-learn.org/>
- **scikit-learn User Guide**
http://scikit-learn.org/stable/user_guide.html
- **scikit-learn API reference**
<http://scikit-learn.org/stable/modules/classes.html>





- ML es un subconjunto de AI. Se utiliza en escenarios en los que necesita que las máquinas **aprendan de grandes volúmenes de datos.**
- El conocimiento adquirido se aplica a un nuevo conjunto de datos.
- ML le da a una máquina la capacidad de aprender de (o acerca de) conjuntos de datos nuevos **sin dar instrucciones explícitas.**



- El software tradicional suele estar compuesto por reglas lógicas sencillas y codificadas a mano.
- Por ejemplo, SI la condición X es THEN, realice la acción Y.
- El aprendizaje automático se basa en modelos estadísticos complejos **para descubrir patrones** en grandes conjuntos de datos.
- Tomemos como **ejemplo la aprobación de un préstamo**.
 - Dados años de historial crediticio y otra información secundaria, **un algoritmo** de aprendizaje automático podría **generar una probabilidad de que el solicitante incumpla**.
 - La lógica detrás de esta evaluación no se codificaría a mano.
 - En cambio, **el modelo se extrapolaría a partir de los registros de miles o millones de otros clientes**.



Aprendizaje automático

- Kevin Patrick Murphy define el aprendizaje automático como “...un **conjunto de métodos que pueden detectar automáticamente patrones en los datos, y luego utilizar los patrones detectados para predecir los datos futuros, o realizar otros tipos de toma de decisiones bajo incertidumbre**”.



Aprendizaje automático

- El **machine learning (ML)** y las **redes neuronales** han sido fundamentales en el **desarrollo y aplicación de la inteligencia artificial (IA)**, especialmente por su capacidad para aprender de datos y tomar decisiones sin intervención humana directa.
- El **machine learning (ML)** es una rama de la IA que permite a los sistemas **aprender** automáticamente **a partir de datos**, identificar patrones y tomar decisiones con mínima intervención humana.
- **Importancia:**
 - Permite que los sistemas se adapten a nuevas situaciones.
 - Mejora continuamente el rendimiento del sistema conforme se alimenta con más datos.
 - Es la base de muchas aplicaciones modernas de IA.

Redes Neuronales

- Las redes neuronales Inspiradas en el cerebro humano son modelos matemáticos que procesan información en capas.
- Las **redes neuronales profundas (deep learning)** son especialmente poderosas para tareas complejas como visión por computadora, procesamiento de lenguaje natural y reconocimiento de voz.
- **Importancia:**
 - Capacidad para aprender representaciones complejas de datos.
 - Se utilizan en tareas donde los algoritmos tradicionales fallan.
 - Son la base de tecnologías como ChatGPT, reconocimiento facial, y asistentes virtuales.



Aplicaciones

- **Visión por Computadora**
 - Diagnóstico médico por imágenes.
 - Reconocimiento facial y de objetos.
 - Vehículos autónomos.
- **Procesamiento de Lenguaje Natural (NLP)**
 - Traducción automática.
 - Chatbots y asistentes virtuales.
 - Análisis de sentimientos en redes sociales.
- **Sistemas de Recomendación**
 - Netflix, YouTube, Amazon usan ML para sugerir contenido personalizado.



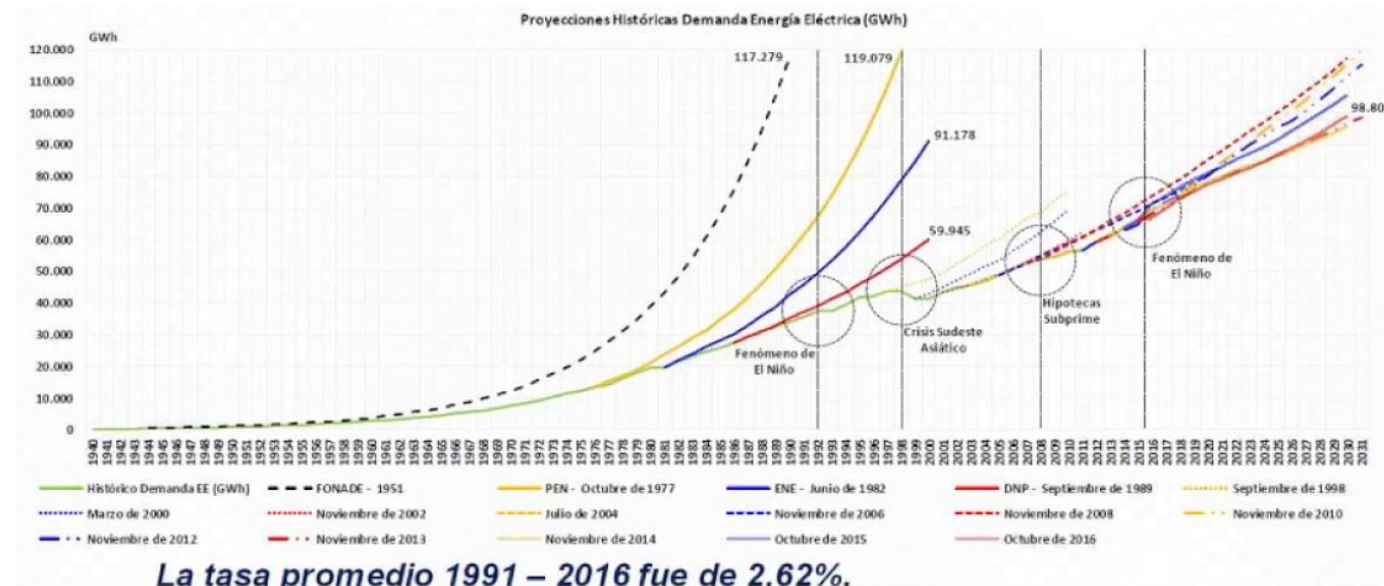
Aplicaciones

- **Predicción y Modelado**
 - Predicción de demanda energética.
 - Modelos meteorológicos.
 - Análisis financiero y bursátil.
- **Robótica Inteligente**
 - Robots que aprenden tareas complejas.
 - Navegación autónoma en entornos dinámicos.
- **Ciberseguridad**
 - Detección de anomalías y amenazas.
 - Sistemas de autenticación biométrica.

Sistemas de Energía y Redes Eléctricas Inteligentes (Smart Grids)

Predicción de demanda eléctrica:

- Se utilizan redes neuronales o modelos de regresión para **anticipar el consumo de energía por hora o por región**, optimizando la generación y distribución.
- Ejemplo:** Una compañía eléctrica entrena un modelo con datos históricos de consumo, temperatura y día de la semana para prever el pico de demanda y evitar sobrecargas.



Sistemas de Energía y Redes Eléctricas Inteligentes (Smart Grids)

Detección de fallas y mantenimiento predictivo:

- Algoritmos de machine learning supervisado (como Random Forest o SVM) detectan patrones **anómalos** en señales eléctricas **que indican posibles fallos en transformadores**, cables o motores.
- **Ejemplo:** Sensores en un transformador envían datos de vibración y temperatura; el modelo predice cuándo podría fallar y alerta antes de que ocurra.

Optimización de redes inteligentes:

- La IA ayuda a **distribuir la carga eléctrica entre fuentes de energía renovable y convencional**, maximizando la eficiencia y reduciendo pérdidas.
- **Ejemplo:** Un sistema basado en IA **ajusta en tiempo real la cantidad de energía** proveniente de paneles solares y eólicos según las condiciones climáticas y la demanda.

Máquinas Eléctricas y Electrónica de Potencia

Diagnóstico de motores eléctricos:

- El ML analiza las señales de **corriente, vibración o sonido** para detectar fallas incipientes (como desequilibrio o cortocircuitos en el bobinado).
- Ejemplo: Un modelo CNN (red neuronal convolucional) **analiza espectros de vibración** para clasificar el tipo de falla en un motor de inducción.



Optimización de convertidores y controladores:

- Algoritmos genéticos o de optimización basados en IA ajustan parámetros de convertidores DC-DC o inversores para mejorar eficiencia energética.

Energías Renovables

Predicción de generación solar y eólica:

- Modelos de machine learning (p. ej., Gradient Boosting o Redes Recurrentes) estiman la producción futura basándose en datos meteorológicos.
- Ejemplo: Predecir la energía solar generada en las próximas horas usando imágenes satelitales y datos de radiación.



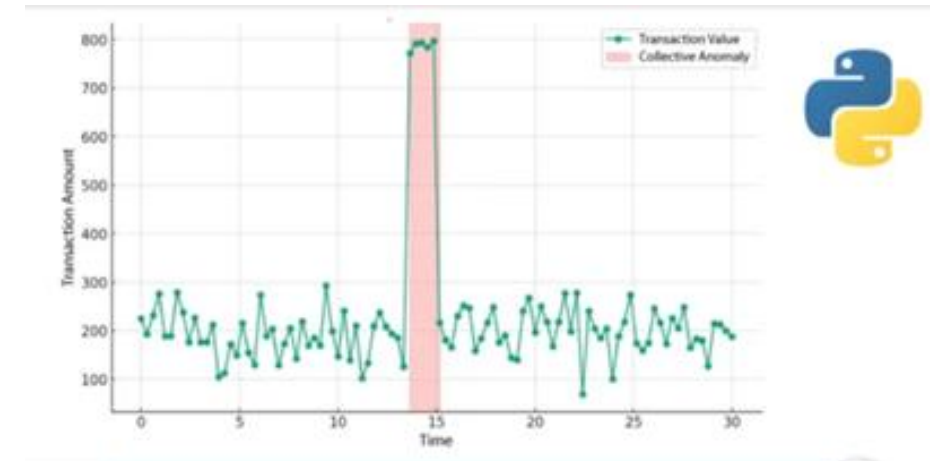
Control inteligente de microrredes:

- La IA decide cuándo almacenar o liberar energía de baterías según la predicción de generación y consumo.

Procesamiento de Señales Eléctricas

Reconocimiento de patrones en señales eléctricas:

- Redes neuronales **analizan formas de onda de corriente o voltaje** para clasificar estados del sistema o **detectar interferencias**.
- Ejemplo: Un sistema de IA identifica si una distorsión en la señal se debe a armónicos, transitorios o ruido de conmutación.



Compresión y reconstrucción de señales:

- Modelos autoencoder reducen el tamaño de los datos medidos por sensores sin perder información relevante.

Sistemas Industriales e IoT Eléctrico

Monitoreo inteligente de plantas eléctricas (Industria 4.0):

- Con el Internet de las Cosas (IoT), los datos de miles de sensores se analizan con IA para optimizar la eficiencia y detectar fallos.

Gestión de energía en edificios inteligentes:

- Algoritmos de ML aprenden patrones de consumo y controlan HVAC, iluminación y equipos para minimizar costos energéticos.



Google Colab for Python

- Google Colab es un entorno gratuito basado en la nube de Jupyter Notebook que utilizan desarrolladores de Python, científicos de datos y desarrolladores del aprendizaje automático, inteligencia artificial y redes neuronales en todo el mundo.
- Permite escribir y ejecutar código Python directamente en el navegador.
- Es una herramienta popular debido a su facilidad de uso y al acceso a potentes recursos informáticos.



Google Colab for Python

- **No requiere configuración** : no es necesario instalar Python ni ninguna biblioteca adicional en su equipo.
- **Acceso gratuito a GPU/TPU** : Google Colab ofrece **acceso gratuito a hardware potente como GPU y TPU**, lo que acelera significativamente tareas como el aprendizaje automático y el entrenamiento de aprendizaje profundo.
- **Colaboración** : Colab permite la **colaboración en tiempo real**, de forma similar a Google Docs. Varios usuarios pueden trabajar en el mismo bloc de notas y ver los cambios de los demás en tiempo real.
- **Integración de almacenamiento en la nube** : Colab se **integra con Google Drive**, lo que le permite guardar sus cuadernos y archivos en la nube y hacerlos accesibles desde cualquier lugar.
- **Admite bibliotecas populares** : las bibliotecas de Python preinstaladas como TensorFlow, Keras, PyTorch, OpenCV, NumPy, Pandas y muchas más están disponibles en Colab, lo que lo convierte en una **plataforma ideal para proyectos de IA, aprendizaje automático** y ciencia de datos.
- **Compartir y publicar fácilmente** : puedes compartir fácilmente tus cuadernos generando un enlace para compartir. El código y los resultados se pueden insertar en sitios web o blogs directamente desde Colab.
- **Uso gratuito** : Colab es gratuito, con planes pagos opcionales para usuarios que necesitan más potencia computacional o tiempos de ejecución más prolongados.



Ejercicio

Optimización de la Operación de la Red Eléctrica mediante Machine Learning: Predicción de la Demanda Máxima de una Subestación



Aplicaciones en la ingeniería eléctrica

Optimización de la Operación de la Red Eléctrica mediante Machine Learning: Predicción de la Demanda Máxima de una Subestación

Introducción

En el campo de la Ingeniería Eléctrica, la **estabilidad y la eficiencia** de la red dependen críticamente de la capacidad de los operadores para anticipar la demanda de energía. **Una predicción inexacta del consumo máximo puede llevar a fallos en el suministro** (si se subestima la demanda) **o a costos operativos innecesarios** (si se sobreestima la generación).

- Este ejercicio aborda la necesidad de predecir el **Consumo Máximo de Potencia (MW)** de una subestación de distribución en las próximas horas.
- Tradicionalmente, esta predicción se basaba en modelos estadísticos lineales o la experiencia histórica. Sin embargo, la creciente influencia de factores variables como el clima extremo y los patrones de consumo dinámicos exige el uso de herramientas más robustas.

Ejercicio

Objetivos

- Predecir el **Consumo Máximo de Potencia (MW)** de una subestación para las próximas horas, basándose en **variables ambientales y temporales** mediante el desarrollo de un modelo de **Regresión Supervisada** utilizando el algoritmo **Random Forest** para predecir el consumo máximo de la subestación.
- Desarrollar un modelo de **Random Forest Regressor** para encontrar las complejas relaciones no lineales que determinan el consumo, superando las limitaciones de los métodos lineales tradicionales.

Ejercicio

Variables

| Variable | Nombre | Descripción | Rol en el Modelo |
|------------------|---------------------|--|---|
| Variable x_1 | Hora del Día | Hora del día (0 a 23). | Factor temporal que refleja hábitos de consumo. |
| Variable x_2 | Temperatura (°C) | Temperatura ambiente pronosticada. | Factor climático clave, asociado al uso de climatización. |
| Variable x_3 | Humedad (%) | Humedad relativa del aire. | Factor climático secundario. |
| Etiqueta (y) | Consumo Máximo (MW) | La potencia máxima registrada por la subestación en ese momento. | Variable a Predecir. |

Aplicaciones en la ingeniería eléctrica

Ejemplo de los datos

| Hora_del_Dia | Temperatura_C | Humedad_Porcentaje | Consumo_Maximo_MW |
|--------------|---------------|--------------------|-------------------|
| 6 | 14.0 | 56.0 | 61.9 |
| 19 | 13.5 | 53.9 | 103.3 |
| 14 | 6.1 | 67.0 | 81.7 |
| 10 | 23.3 | 68.1 | 110.2 |
| 7 | 20.1 | 32.7 | 104.6 |



Ejercicio

Detección de sobrecalentamiento en transformadores eléctricos mediante Machine Learning



Aplicaciones en la ingeniería eléctrica

Detección de sobrecalentamiento en transformadores eléctricos mediante Machine Learning

Introducción

- En los sistemas eléctricos de potencia, los **transformadores** son componentes esenciales para la **transmisión y distribución eficiente de energía**. Sin embargo, las condiciones de operación —como la carga, la temperatura ambiente, la humedad y el nivel del aceite dieléctrico— influyen directamente en su desempeño térmico.
- El **sobrecalentamiento** es una de las principales causas de fallas en transformadores, ya que puede degradar el aislamiento, reducir la vida útil del equipo e incluso provocar interrupciones del servicio. Por ello, resulta fundamental implementar mecanismos de **monitoreo predictivo** que permitan **anticipar condiciones anómalas antes de que se produzcan fallas críticas**.

Objetivos

- Aplicar un modelo de **clasificación basado en Random Forest** para **predecir** el riesgo de **sobrecalentamiento de transformadores eléctricos**, utilizando datos simulados que representan condiciones operativas y ambientales.
- El modelo deberá:
 - Analizar variables relevantes como carga, temperatura ambiente, humedad, edad y nivel de aceite.
 - Clasificar el estado del transformador como **normal** o **sobrecalentado**.
 - Generar **alertas automáticas** cuando se detecte una alta probabilidad de riesgo térmico.
 - Interpretar la **importancia de las variables** y evaluar el desempeño del modelo mediante métricas de clasificación.



| Nombre de la variable | Tipo | Unidad / Escala | Descripción | Influencia esperada |
|-----------------------|--------------------|--------------------------------|---|---|
| load_pct | Numérica continua | Porcentaje (%) | Carga actual del transformador respecto a su capacidad nominal. | A mayor carga, mayor generación de calor → aumenta el riesgo de sobrecalentamiento. |
| ambient_C | Numérica continua | Grados Celsius (°C) | Temperatura ambiente que rodea al transformador. | A mayor temperatura ambiente, más difícil es disipar el calor → incrementa el riesgo. |
| humidity_pct | Numérica continua | Porcentaje (%) | Nivel de humedad relativa del aire en el entorno. | Puede afectar la conductividad y el enfriamiento; valores muy altos o bajos pueden alterar el comportamiento térmico. |
| age_yrs | Numérica continua | Años | Edad del transformador desde su instalación. | Transformadores más antiguos tienden a ser menos eficientes térmicamente. |
| oil_level | Numérica continua | Proporción (0.5 a 1.0) | Nivel relativo del aceite dieléctrico usado para enfriamiento y aislamiento. | Un menor nivel de aceite reduce la capacidad de disipar calor → aumenta el riesgo. |
| temp_C | Numérica continua | Grados Celsius (°C) | Temperatura interna calculada del transformador. | Se usa solo en la simulación, no como entrada del modelo de clasificación. |
| status | Categórica (0 / 1) | 0 = Normal, 1 = Sobrecalentado | Variable objetivo que indica si el transformador presenta sobrecalentamiento. | El modelo intenta predecir esta variable. |





Ejercicio

Detección temprana de riesgo de infección postoperatoria

Aplicaciones en la ingeniería Biomédica

Detección temprana de riesgo de infección postoperatoria

Introducción

- Después de una cirugía, los pacientes son monitoreados para detectar complicaciones. Una de las más comunes es la **infección del sitio quirúrgico (ISQ)**, la cual puede manifestarse en los primeros días postoperatorios y aumentar significativamente el riesgo de reintervención, sepsis o prolongación de la estancia hospitalaria.
- Tradicionalmente, el personal médico evalúa el riesgo con base en signos clínicos (temperatura, pulso, presión, recuento leucocitario, etc.), pero los sistemas automáticos basados en **machine learning** pueden analizar múltiples variables simultáneamente y emitir **alertas predictivas tempranas**, incluso antes de que la infección sea evidente clínicamente.



Objetivo del modelo

Entrenar un **modelo de clasificación supervisada (Random Forest)** para predecir si un paciente postoperado **desarrollará una infección dentro de los próximos 7 días**, a partir de sus datos clínicos y quirúrgicos de las primeras 24 horas postoperatorias.

El modelo generará:

- una probabilidad de infección,
- una clasificación automática (Sí/No),
- y una alerta de riesgo para el personal médico.

Variables

| Variable | Nombre | Descripción | Tipo | Rango/Unidades |
|------------------|----------------------|--|----------|------------------------|
| Variable x_1 | temp_24h_mean | Temperatura corporal media en las primeras 24h | Continua | 35.5 – 40.0 °C |
| Variable x_2 | heart_rate_mean | Frecuencia cardiaca media | Continua | 50 – 140 bpm |
| Variable x_3 | wbc_count | Recuento de leucocitos | Continua | 4 – 20 $\times 10^9/L$ |
| Variable x_4 | surgery_duration_min | Duración de la cirugía | Continua | 90 – 150 min |
| Variable x_5 | Age | Edad de la persona | Continua | 20-80 |
| Etiqueta (y) | infection | Variable a predecir | Binaria | 0 = No, 1 = Sí- |



Herramientas de **IA e IAG (Inteligencia Artificial Generativa)** que están ayudando a ingenieros eléctricos en tareas como diseño, revisión y optimización de circuitos

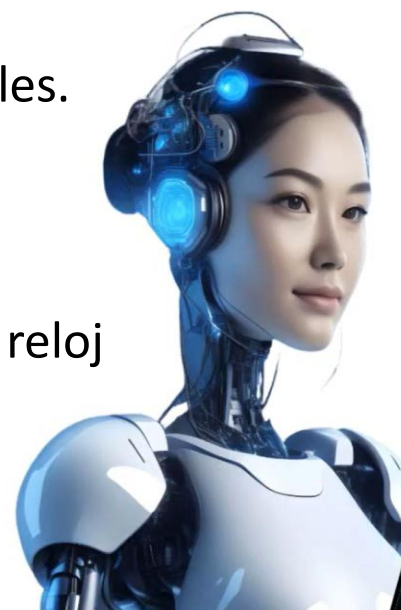


TINA y TINACloud

Software de simulación y diseño de circuitos con IA integrada.

Funciones:

- Diseño y simulación de circuitos analógicos y digitales.
- Generación de código Arduino para prototipado.
- Interacción por lenguaje natural para modificar parámetros.
- Diseño de osciladores analógicos y generadores de reloj digital
- Selección y rediseño de circuitos de evaluación de diferentes fabricantes
- Reconocimiento de imágenes con Python o MCU
- Creación de soluciones paso a paso de circuitos CC/CA simples.



offline
AI



online
AI

<https://www.tina.com/es/built-in-artificial-intelligence/>