gradiant

PyCones 24 VIGO

Pattern Busters: encontrando patrones significativos en Python con aplicaciones reales

Pablo García Santaclara

Camilo Piñón Blanco

$WHOAMI 👨‍💻

Pablo García Santaclara

- M.Sc. Teleco
- Estudiante de PhD
- Ingeniero-Investigador @GRADIANT
- IA y Optimización

Camilo Piñón Blanco

- M.Sc. Teleco
- Ingeniero-Investigador @GRADIANT
- Security Analytics

Índice

First things first ⚠️

1. Clonar / descargar el siguiente repositorio:

   github.com/Gradiant/PyConEs2024-PatternBusters

2. Seguir los pasos del README.md:

   1. **Instalar pyenv y virtualenv**
   2. **Instalar Python 3.9.19 con pyenv**
   3. **Crear un entorno virtual y activarlo**
   4. **Instalar las dependencias del proyecto**
   5. **Abrir Jupyter Lab o Jupyter Notebook**

AGENDA 2030

OBJETIVOS ✍️

✔️Presentar herramientas para análisis de datos avanzado, especialmente para **análisis exploratorio** y **descubrimiento de patrones**.

✔️El marco común a todas estas técnicas es que nos **han resultado útiles en algún momento de nuestro trabajo del día a día**.

✔️Demostrar estos conceptos con **ejemplos *hands-on* cercanos al mundo real**.

✔️Mostrar cómo implementaríamos estas técnicas en nuestro flujo de trabajo y dar ***snippets* de código** para que vosotr@s los probéis en los vuestros.

AGENDA 2030

DISCLAIMER ⚠️

Este taller **no pretende proporcionar una descripción académica** de la implementación de los algoritmos, sino una intuición detrás de su funcionamiento y su uso práctico.

Esta charla **no pretende ser una clase de estadística**.

Junto con las técnicas presentadas **proporcionaremos punteros a dónde encontrar descripciones detalladas** que nos han ayudado a comprenderlas y utilizarlas. Hemos considerado que estos punteros son el mejor punto de partida para comprender la intuición detrás de las técnicas presentadas y poder explorarlas / aplicarlas desde un primer momento.

AGENDA 2030

gradiant

Datos tabulares

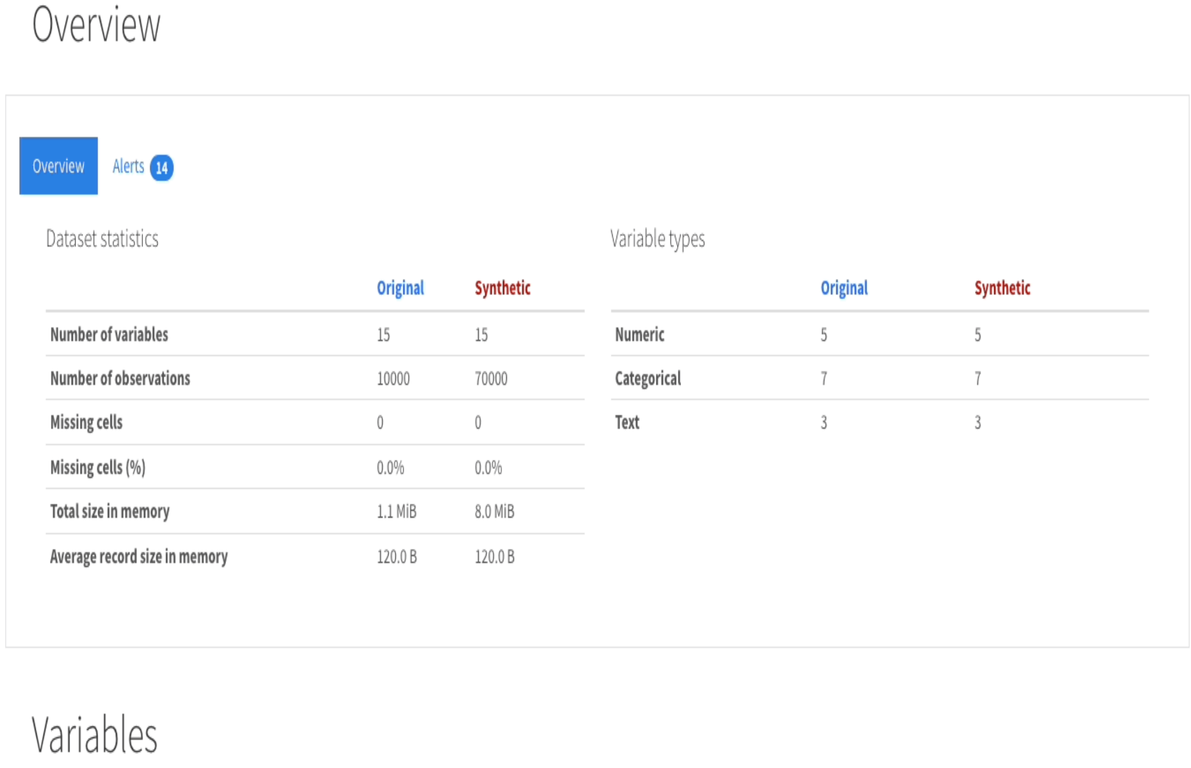
¿Por dónde empiezo? ¡Perfilado de datos!

ydata-profiling

Automatiza y **estandariza** la generación de **informes** con **estadísticas** y **visualizaciones**

Simplifica la exploración inicial de datos en una **sola línea de código:**

```
import pandas as pd
from ydata_profiling import ProfileReport
df = pd.read_csv('data.csv')
profile = ProfileReport(df, title="Profiling Report")
```

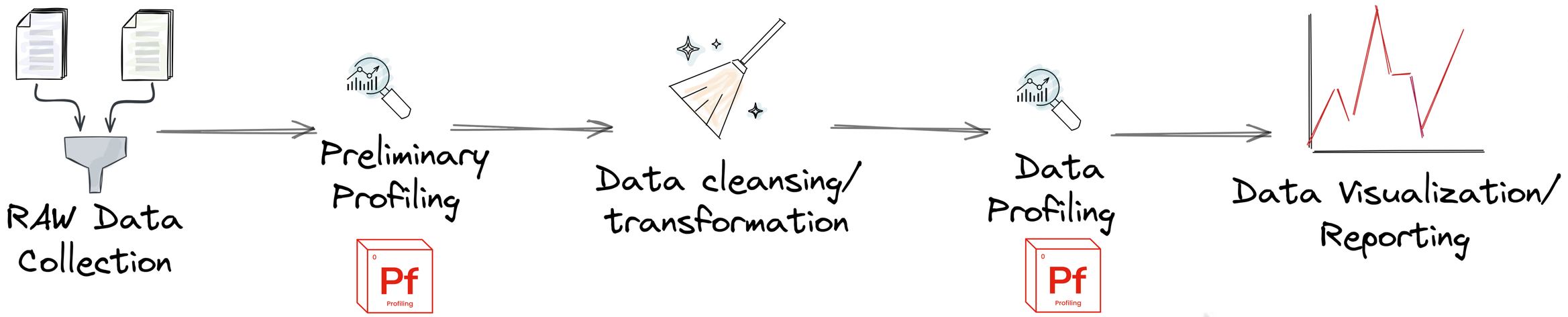


docs.profiling.ydata.ai

¿Por dónde empiezo? ¡Perfilado de datos!

`ydata-profiling`

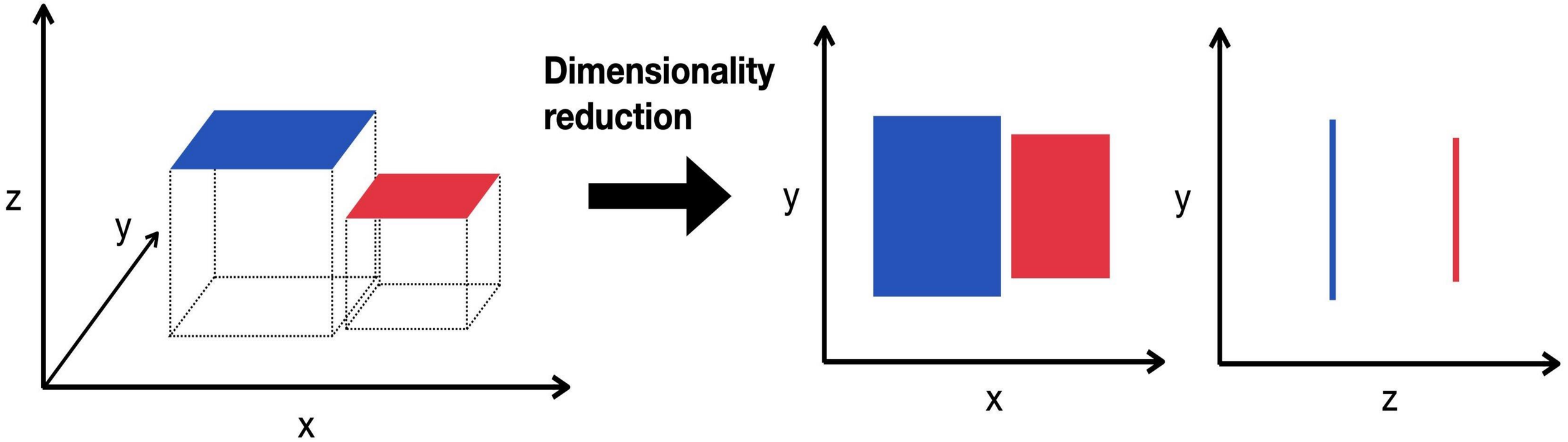
Flujo en una exploración real:



https://docs.profiling.ydata.ai/latest/features/big_data/
https://blog.dataiku.com/active-sampling-data-selection-for-efficient-model-training

Datos tabulares - Reducción de dimensionalidad avanzada

Útiles para **exploración**, **visualización** y **descubrimiento de patrones**, no tanto para análisis cuantitativos detallados.



Datos tabulares - Reducción de dimensionalidad avanzada

- PCA está bien para **casos sencillos** (datos con fuerte componente lineal).
- 2 algoritmos destacados y adoptados en la industria:

t-SNE (t-distributed Stochastic Neighbor Embedding), 2008

Popular para la visualización de datos de alta dimensión

⚠️ ***WARNING**: Realmente t-SNE **no es una técnica de reducción de dimensionalidad** en su significado más general.*

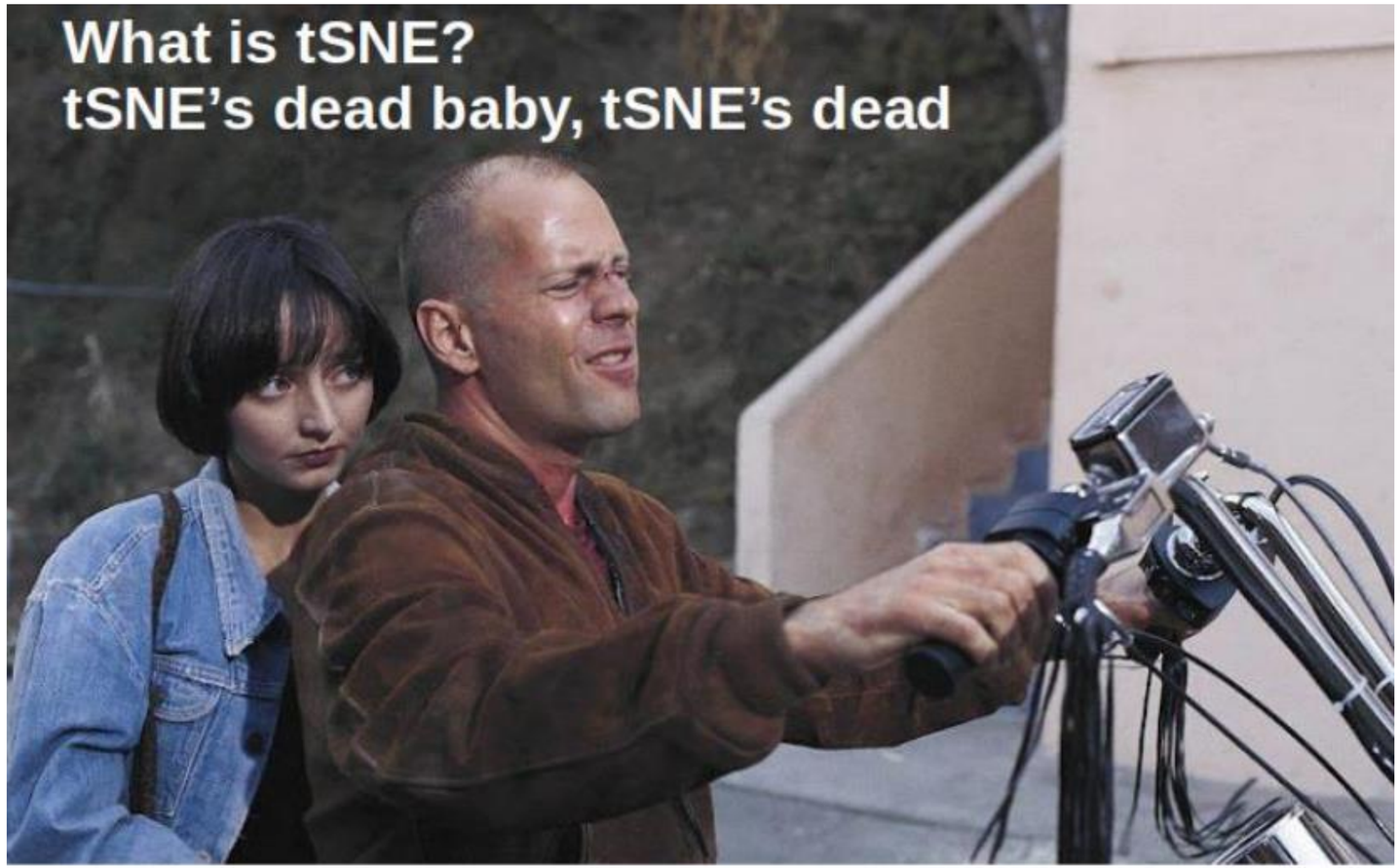
UMAP (Uniform Manifold Approximation and Projection), 2018

Está ganando popularidad en la industria debido a su velocidad y capacidad para preservar tanto las estructuras locales como globales.

L. van der Maaten and G. Hinton, "*Visualizing Data using t-SNE*," Journal of Machine Learning Research, Nov. 2008.
L. McInnes, J. Healy, and J. Melville, "*UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*," *arXiv preprint arXiv:1802.03426*, Feb. 2018.

Datos tabulares - Reducción de dimensionalidad avanzada

~~TSNE~~



CC: Nikolay Oskolkov

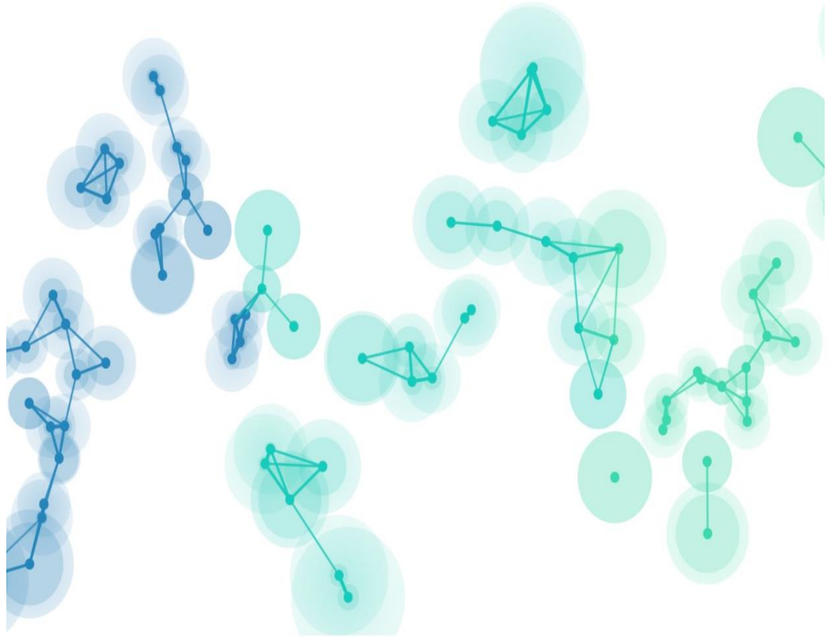
L. van der Maaten and G. Hinton, "*Visualizing Data using t-SNE*," Journal of Machine Learning Research, Nov. 2008.

Datos tabulares - Reducción de dimensionalidad avanzada

UMAP crea una representación de los datos en alta dimensión y la optimiza para una versión de estos en baja dimensión.

Intuición detrás del algoritmo

- Usa un "***complejo simplicial difuso****", que podemos ver como un grafo con aristas ponderadas entre los vértices.
- Este grafo permite identificar **qué puntos están más cercanos** y deben aparecer más cerca en la proyección final.
- Se construye extendiendo un radio desde cada punto en el espacio de alta dimensión, **conectando puntos cercanos**.
- Una vez construido el grafo en alta dimensión, UMAP optimiza la construcción de **un análogo en baja dimensión para que sea lo más similar posible**.



https://es.wikipedia.org/wiki/Complejo_simplicial
https://pair-code.github.io/understanding-umap/
La mejor explicación de UMAP

*El enfoque es "**difuso**", porque la probabilidad de conexión disminuye con la distancia, equilibrando relaciones locales y globales.

AGENDA 2030

Datos tabulares - Reducción de dimensionalidad avanzada ✂️

Intuición detrás del algoritmos

UMAP

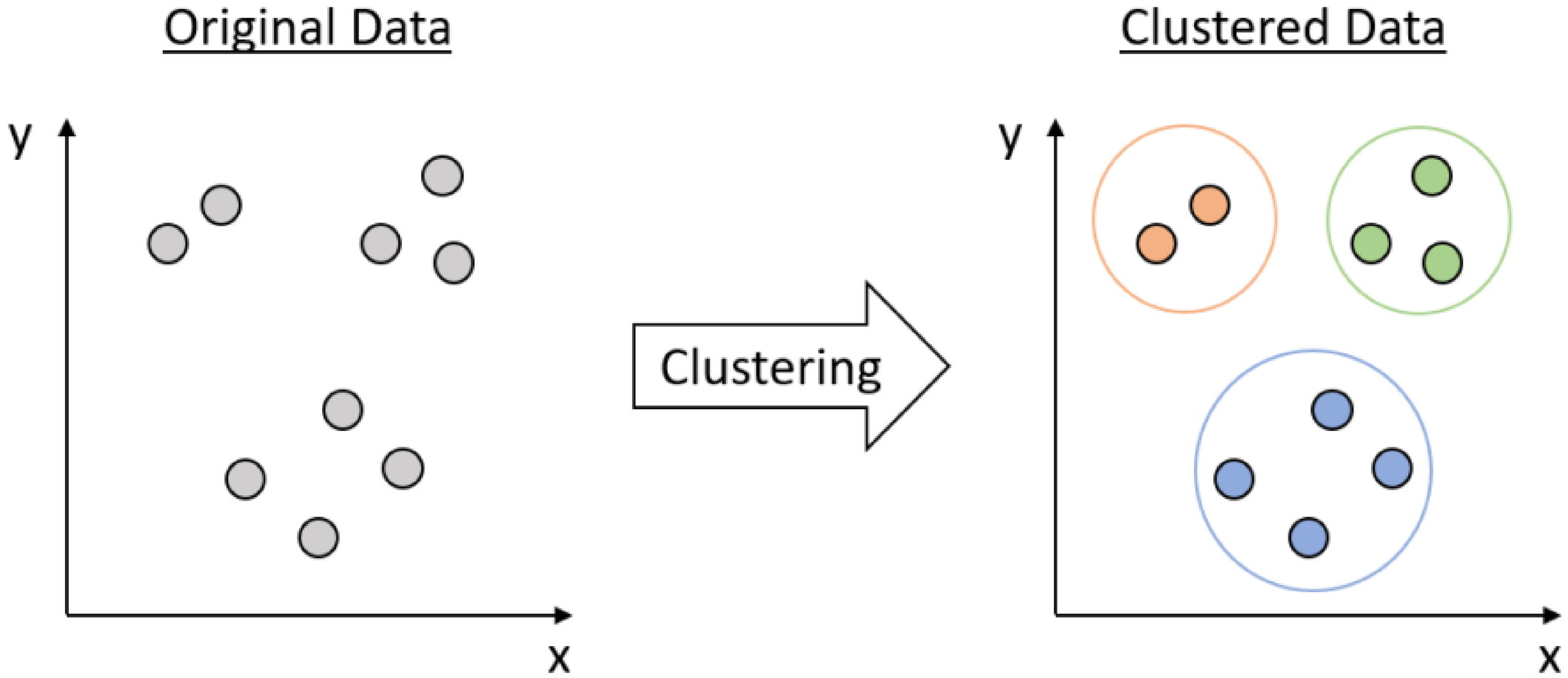
- **Más rápido** (tanto frente tamaño de dataset como en dimensionalidad)*
- **Mejor preservación** de la estructura global de los datos.
- **2 parámetros** principales:
  - `n_neighbors`: Define cuántos vecinos cercanos se consideran para cada punto en el proceso de construcción del grafo de proximidad.

    ↑↑↑ **`n_neighbors`** ⟹ ↑↑↑ prioridad a estructura global
  - `min_dist`: Controla cuán separados están los puntos en la proyección final.

    ↓↓↓ **`min_dist`** ⟹ ↑↑↑ densidad de los puntos en la proyección

AGENDA 2030

Datos tabulares – Clustering



Datos tabulares – Clustering automático con HDBSCAN

HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise)

Intuición detrás del algoritmo

Basado en **DENSIDAD**:

- Mide la densidad local alrededor de cada punto.
- **Identifica clústers encontrando regiones densas** de puntos en el espacio de datos.
- Trata puntos en áreas de muy baja densidad como ruido.

JERÁRQUICO:

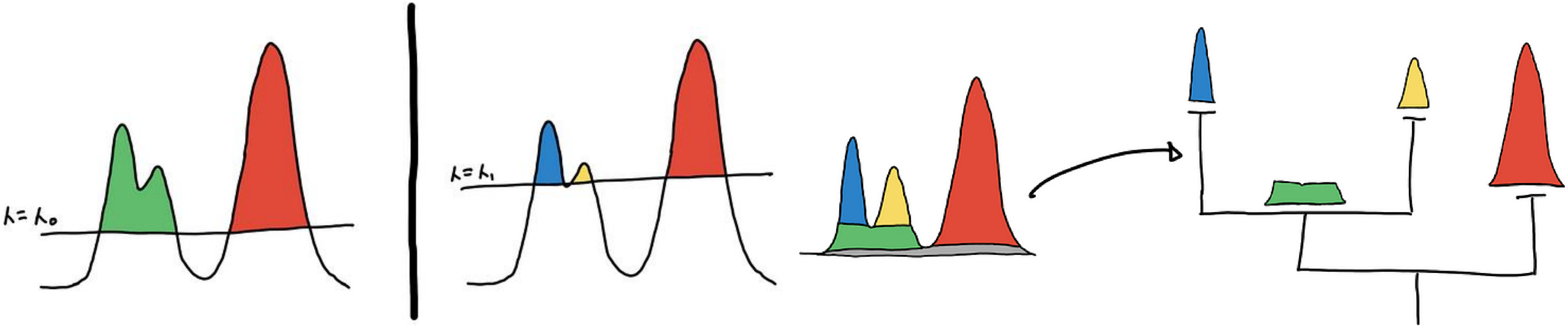
- Construye una **jerarquía de clústeres basada en diferentes niveles de densidad**.
- Selecciona automáticamente el **nivel de agrupamiento más estable** (robusto) usando una técnica de condensación de la jerarquía.

McInnes L, Healy J. *Accelerated Hierarchical Density Based Clustering* In: 2017 IEEE International Conference on Data Mining Workshops (ICDMW), IEEE. 2017
R. Campello, D. Moulavi, and J. Sander, *Density-Based Clustering Based on Hierarchical Density Estimates* In: Advances in Knowledge Discovery and Data Mining, Springer. 2013

Datos tabulares – Clustering automático con HDBSCAN 🧩

HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise)

Intuición detrás del algoritmo



Se transforma el espacio según la densidad

Se calcula la jerarquía de clústers

https://pberba.github.io/stats/2020/01/17/hdbscan/
La mejor explicación de HDBSCAN

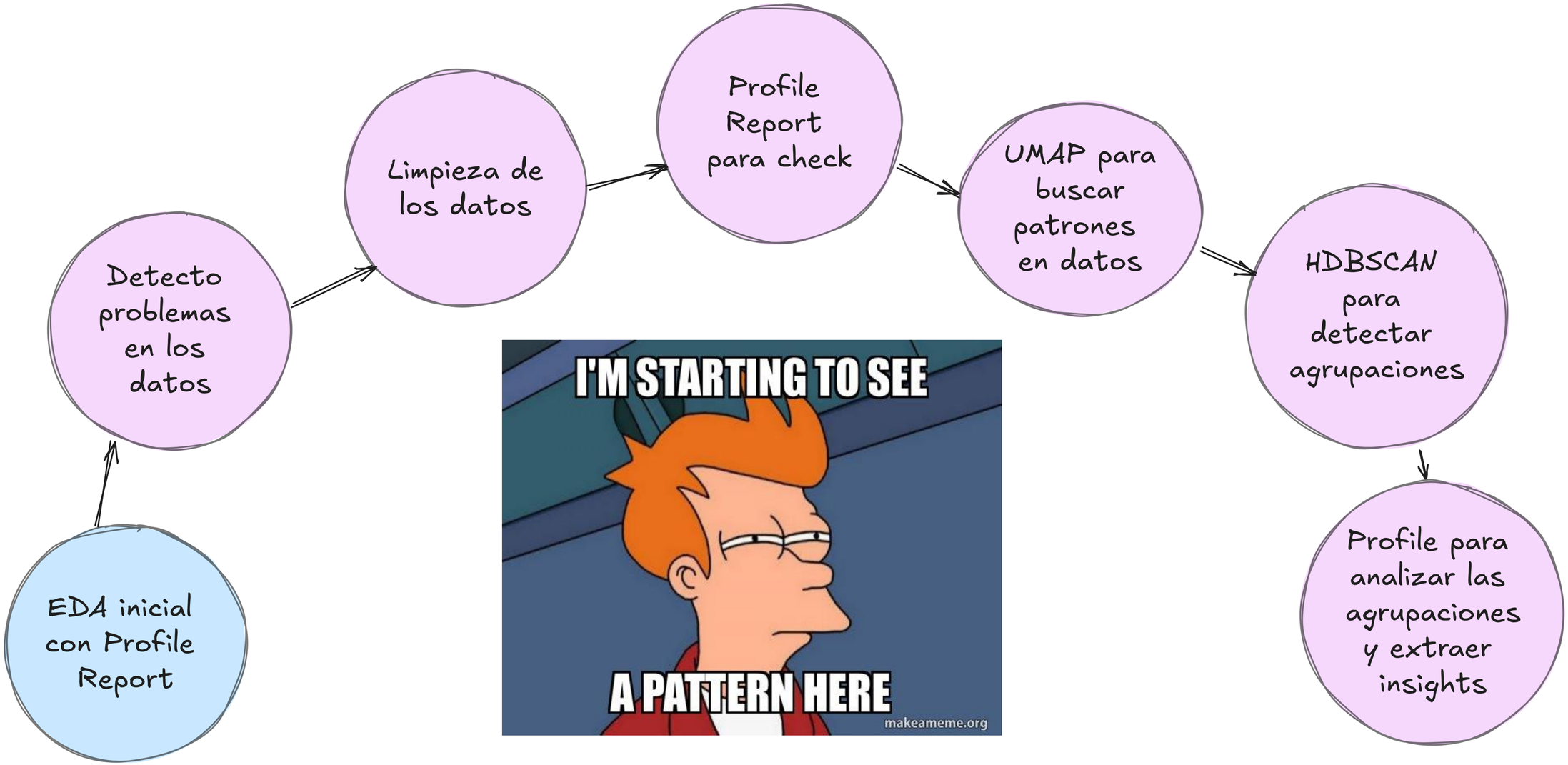
Datos tabulares – Clustering automático con HDBSCAN 🧩

2 parámetros principales:

- `min_cluster_size`: Afecta el tamaño mínimo de los clústeres.
- `min_samples`: Número de muestras en un vecindario para que un punto se considere punto central.

https://hdbscan.readthedocs.io/en/latest/how_hdbscan_works.html

Datos tabulares – Combinando técnicas en un caso real 🧰

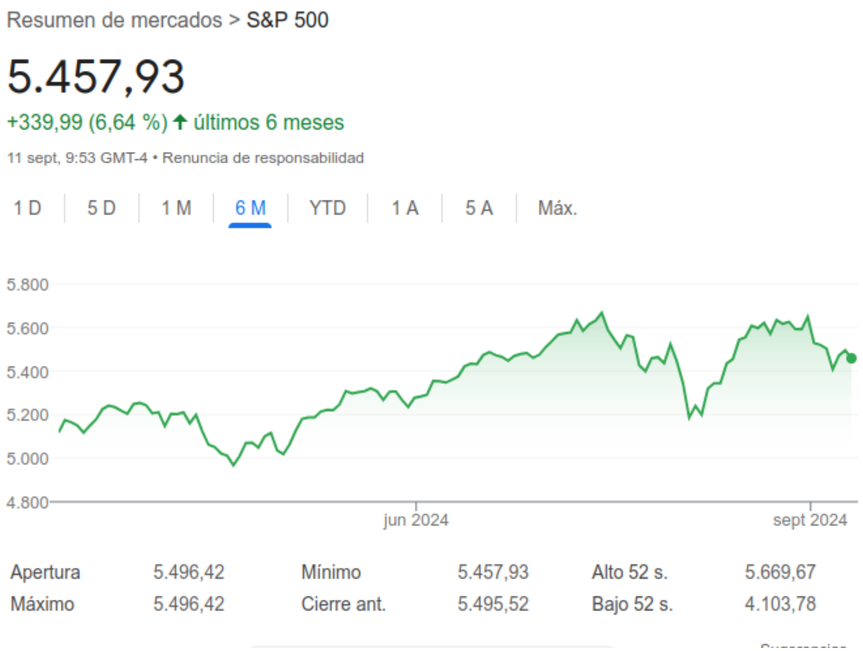


gradiant

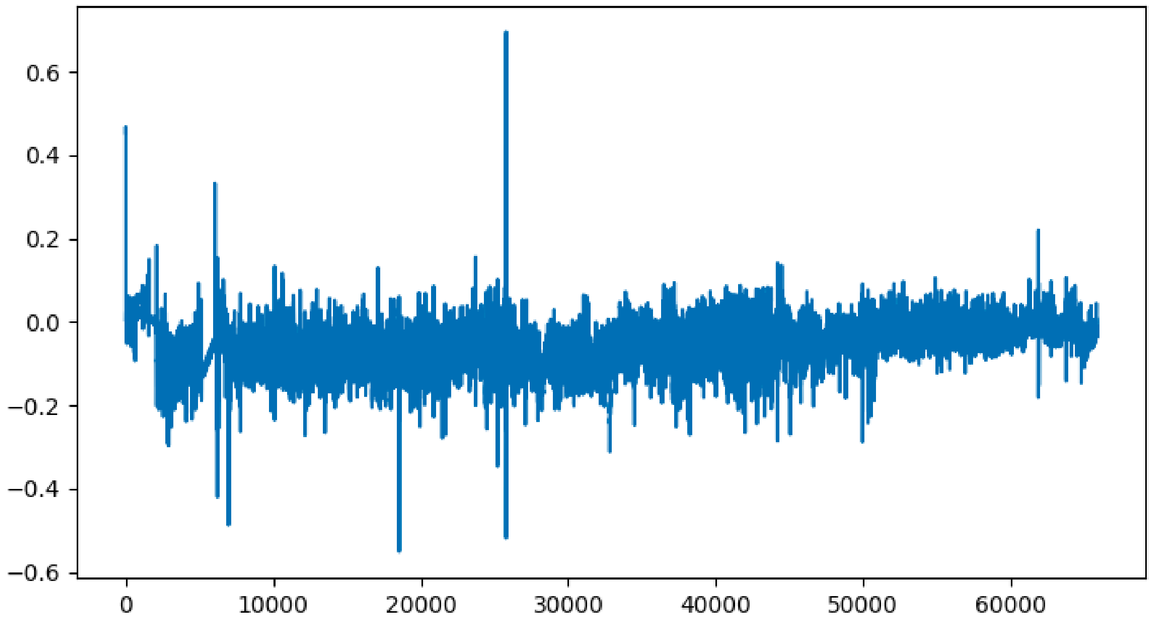
Series temporales

Series temporales 📈 🕰️

Una **serie temporal** es una secuencia de datos medidos y ordenados cronológicamente indexados por una variable temporal.



S&P 500



Gráfica de potencia en componente industrial

Series temporales 📈 🕰️ - `stumpy`

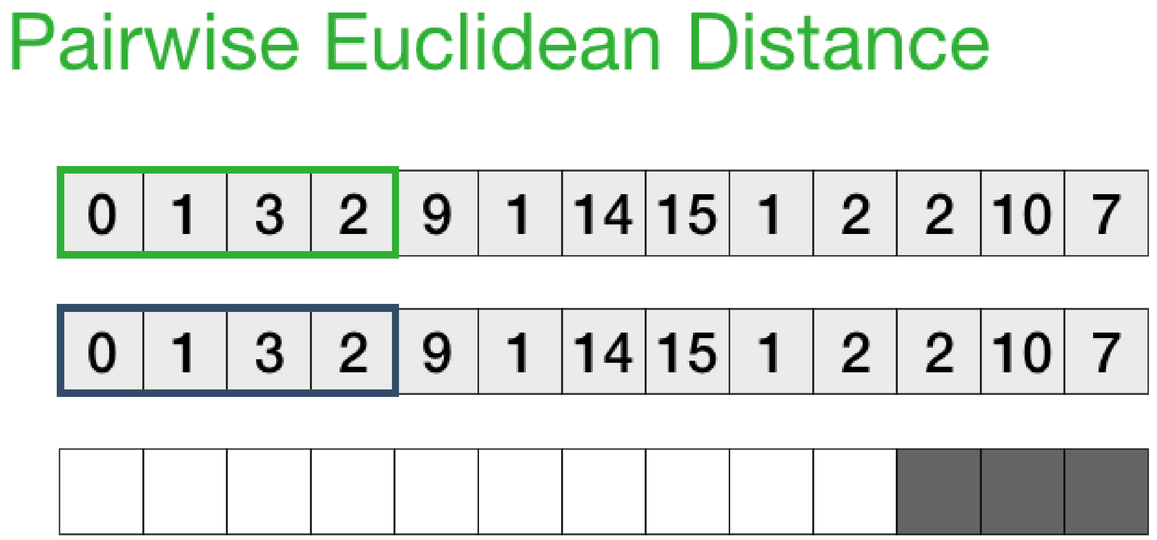
El algoritmo **STOMP** es un método de la familia de algoritmos de ***Matrix Profile***, diseñado para encontrar patrones repetitivos.

Stumpy es una librería de Python que computa la *Matrix Profile* de manera eficiente.

Series temporales 📈 🕰️ - `stumpy`

La *Matrix profile* es una matriz de distancias euclídeas z-normalizadas, entre todas las subsecuencias de una serie temporal.

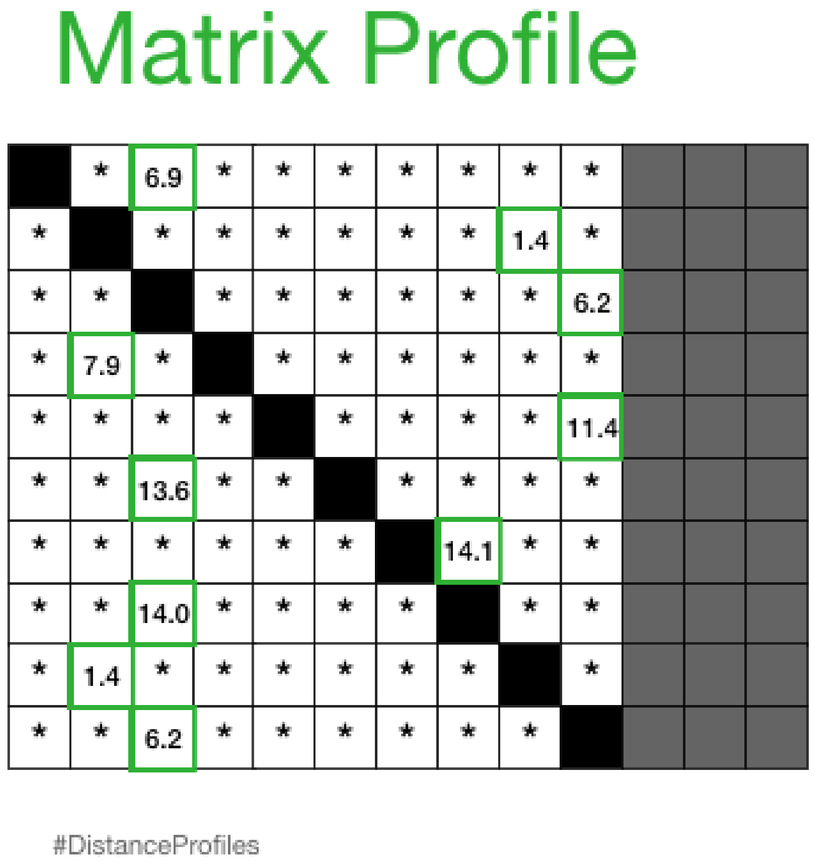
Para realizar la *Matrix Profile*, se escoge un tamaño de ventana fijo para comparar las subsecuencias.



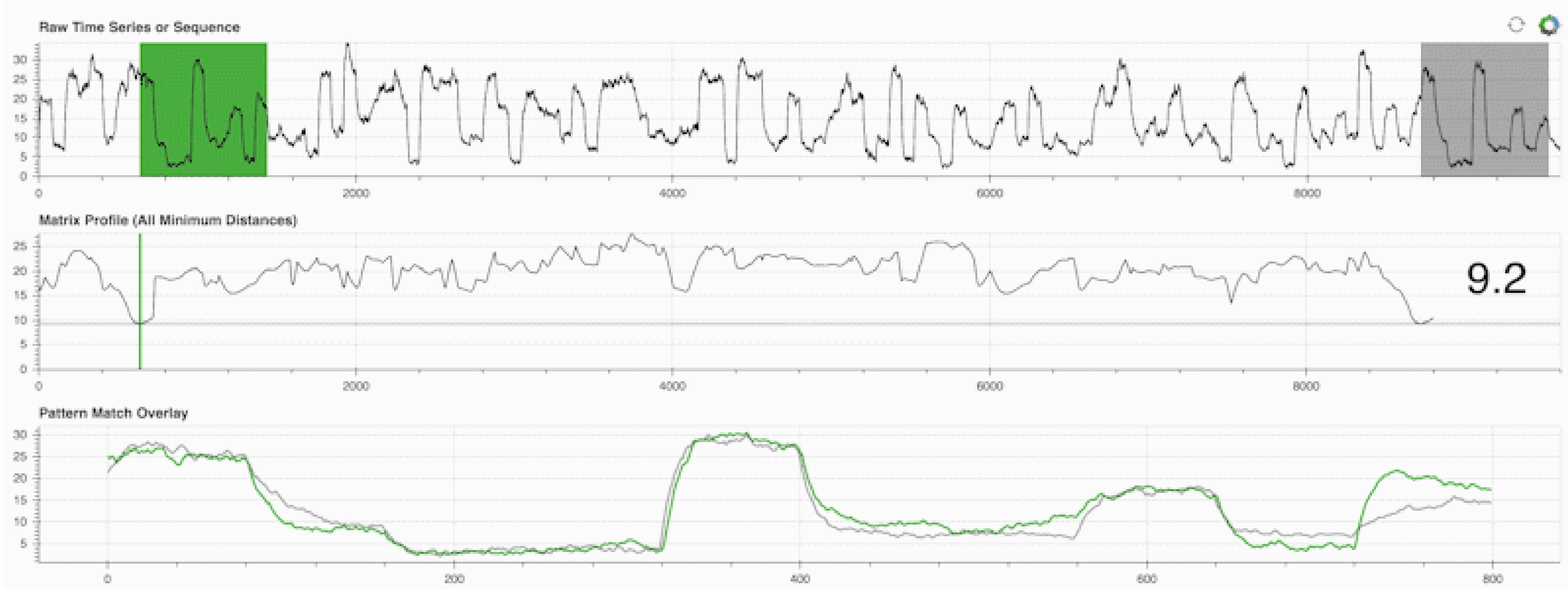
Series temporales 📈 🕰️ - `stumpy`

La *Matrix Profile* es un vector que almacena la distancia euclidiana entre toda subsecuencia dentro de la serie temporal y su vecino más cercano.

Stumpy usa un algoritmo para reducir la complejidad de estos cálculos.



Series temporales 📈 🕰️ - `stumpy`



stumpy.readthedocs.io

AGENDA 2030

Cuando debería usar stumpy ❓

Detección de patrones anómalos:

En procesos de fabricación, es posible identificar patrones fuera de lo común que, eventualmente, pueden estar vinculados a problemas en los productos.

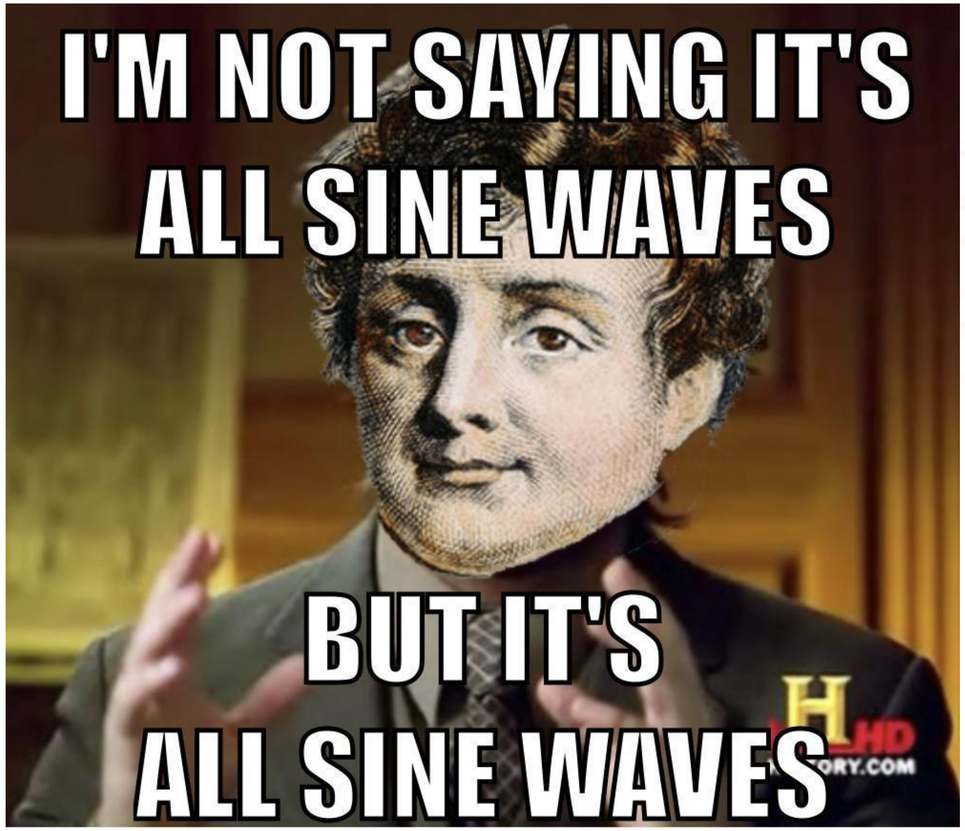
Detección de patrones recurrentes:

Por ejemplo: en un problema de series temporales de sensores, con STUMPY se pueden analizar las series temporales que registran variables como temperatura o presión. Después de detectar patrones recurrentes se pueden relacionar con métricas de calidad y, de esta forma, optimizar los procesos de producción.

Comparación de rendimiento de máquinas:

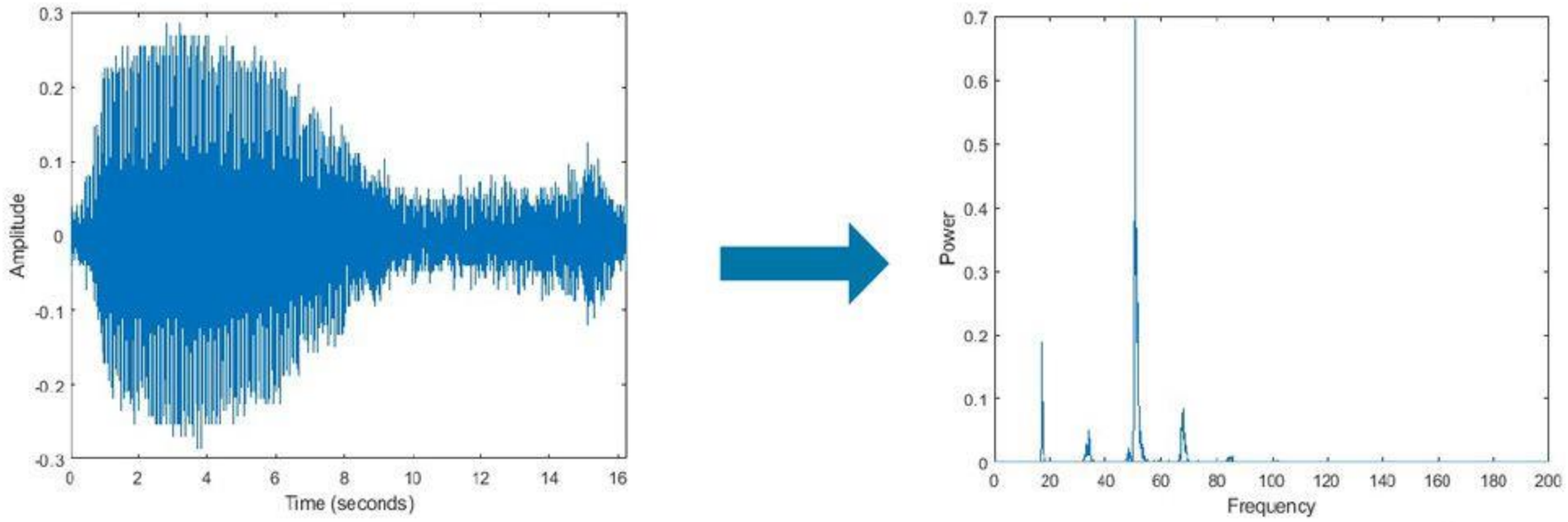
Con varias máquinas similares en una línea de producción, STUMPY te permite comparar su rendimiento. Al calcular perfiles de matriz, se pueden detectar diferencias en su funcionamiento.

Transformada de Fourier



Series temporales 📈 🕰️ – Transformada de Fourier Discreta

La **Transformada de Fourier Discreta** (**DFT**) convierte una señal discreta en el dominio del tiempo a su representación en el dominio de la frecuencia:



www.mathworks.com/discovery/fft.html

Series temporales 📈 🕰️ – Transformada de Fourier Discreta

El mayor contra es el coste computacional:

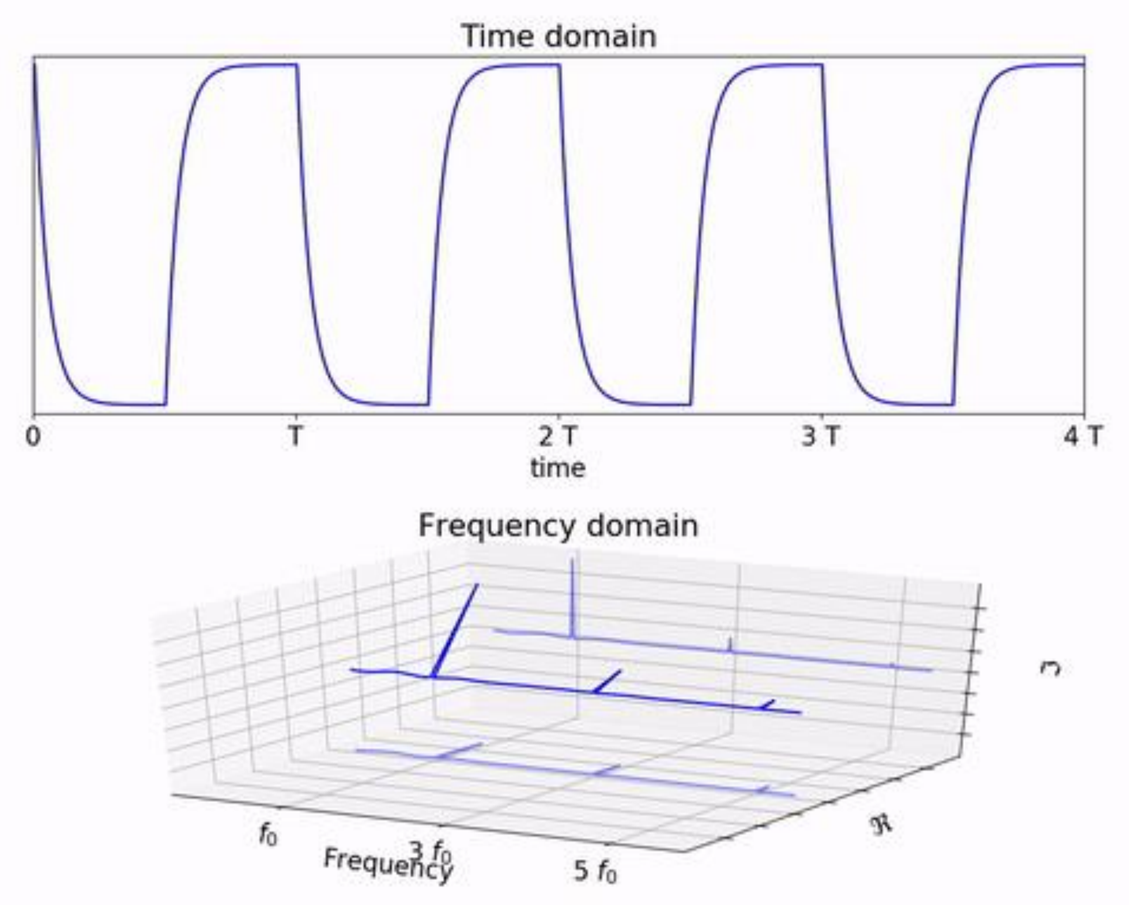
~~$O(N^2)$~~

$O(N \log N)$

Series temporales 📈 🕰️ – Fast Fourier Transform

La FFT también se puede utilizar para calcular la transformada inversa.

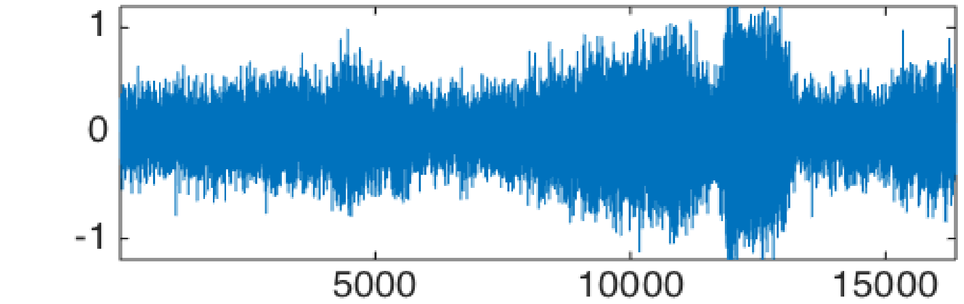
Se puede hacer detección de patrones con FFT a través de observar los picos de amplitud.

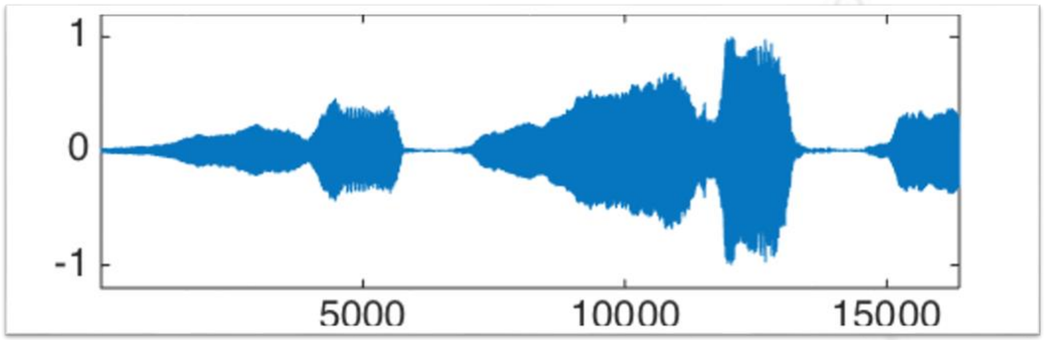


Series temporales 📈 🕰️ – Fast Fourier Transform (FFT)

La FFT también es útil para quitar ruido.

La señal se transforma al dominio de la frecuencia y luego se filtran frecuencias no deseadas.





www.numerical-tours.com/matlab/

AGENDA 2030

Cuando debería usar la FFT ❓

Análisis de vibraciones:

Examinar vibraciones en maquinaria industrial, permitiendo identificar frecuencias anómalas o detectar desgaste en las principales amplitudes. Esto facilita la detección temprana de fallos y posibilita un mantenimiento predictivo.

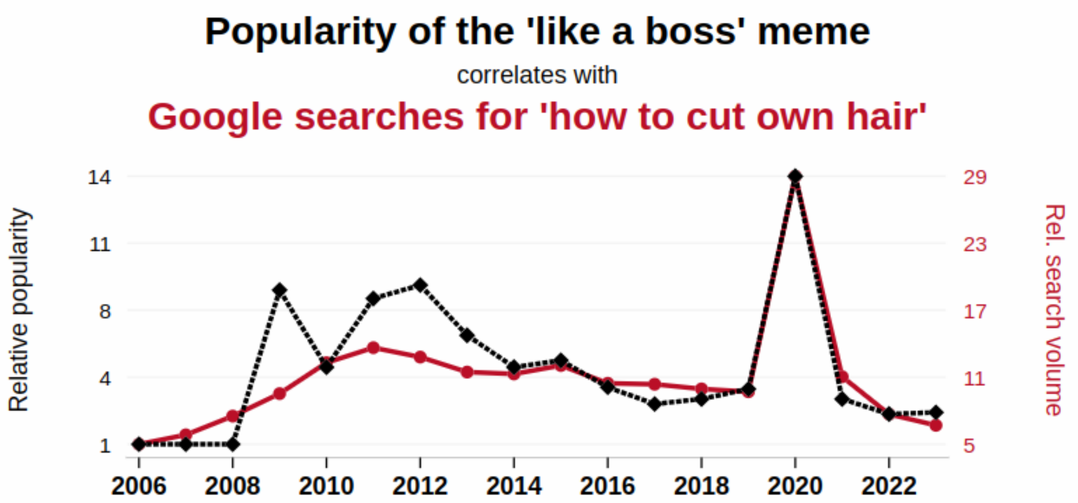
Patrones de uso energético:

La FFT permite identificar patrones cíclicos y estacionales en el consumo de energía. Esta información puede ser aprovechada para optimizar la programación de la producción y reducir costos energéticos, ajustando el uso a las fluctuaciones de demanda.

Procesamiento de señales:

Limpiando señales contaminadas por ruido y de esta manera, mejorando su calidad.

Observaciones finales 🤔



CC: Tyler Vigen

- **Correlación != Causalidad**
- Es MUY difícil (prácticamente imposible) demostrar causalidad en los patrones.
- La gran mayoría (por no decir todas) las técnicas de detección de patrones **se basan en hipótesis** o asunciones a priori sobre los datos. Estas no tienen por qué ser siempre ciertas.
- No dejarse llevar por impresiones "subjetivas" sobre los datos ⟶ Entender los procesos subyacentes en la medida de lo posible.

https://tylervigen.com/spurious-correlations

AGENDA 2030

gradiant

PyCones 24 VIGO

❤️ ¡Gracias! ❤️

pgsantaclara@gradiant.org

cpinon@gradiant.org