

# Machine Learning

Introducción a la Inteligencia Artificial - LCC - 2024



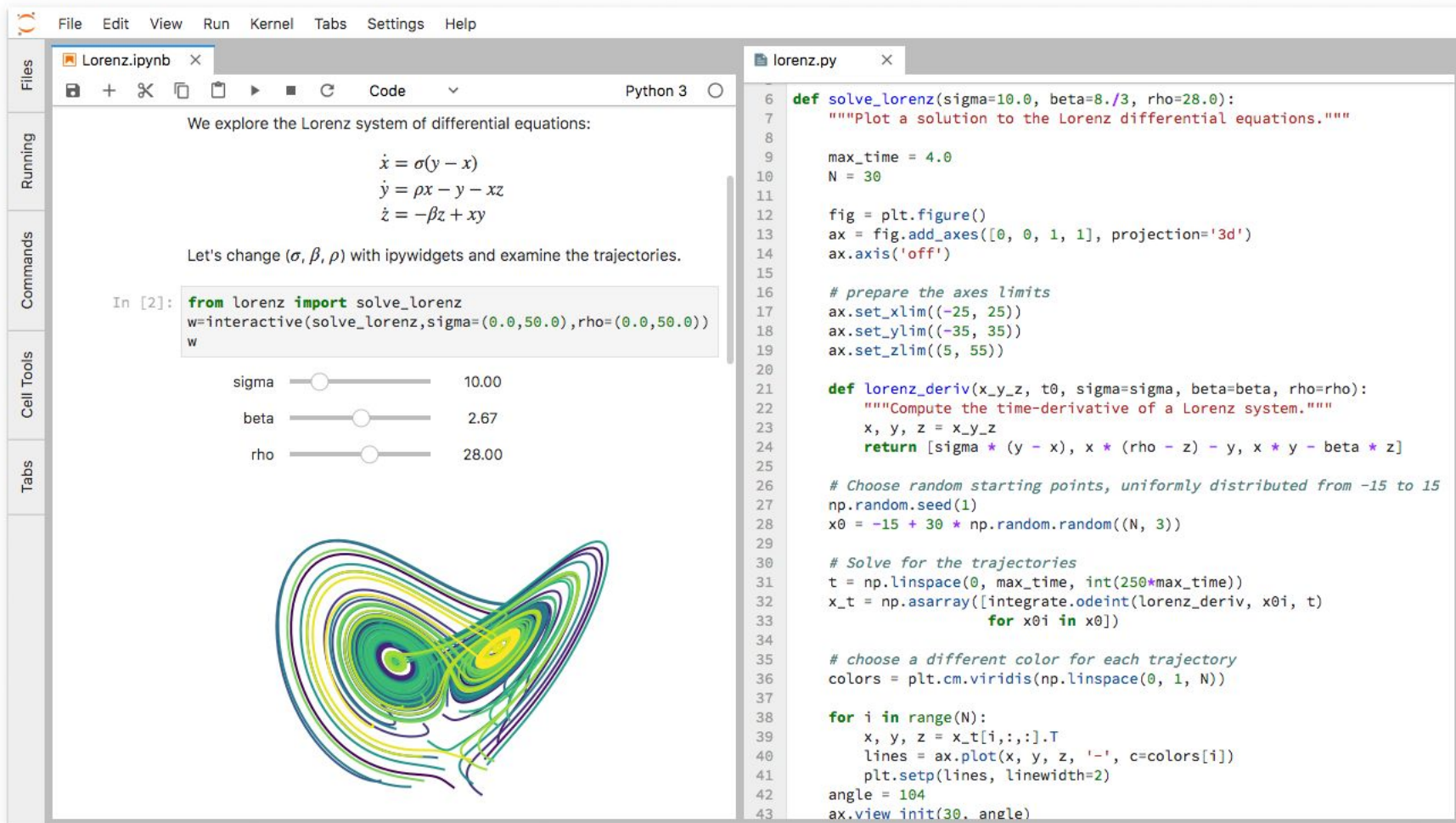
# Jupyter

Jupyter Lab + Jupyter Notebooks

[jupyter.org/](https://jupyter.org/)

- Gratuito y Open Standard
  - Desarrollo y Visualización Web
- 
- Data Science
  - Scientific Computing
  - Computational Journalism
  - Machine Learning

---



# Instalación

## Jupyter Labs

```
pip3 install jupyterlab &&\  
jupyter lab
```

[jupyterlab.readthedocs.io](https://jupyterlab.readthedocs.io)

## Jupyter Notebook

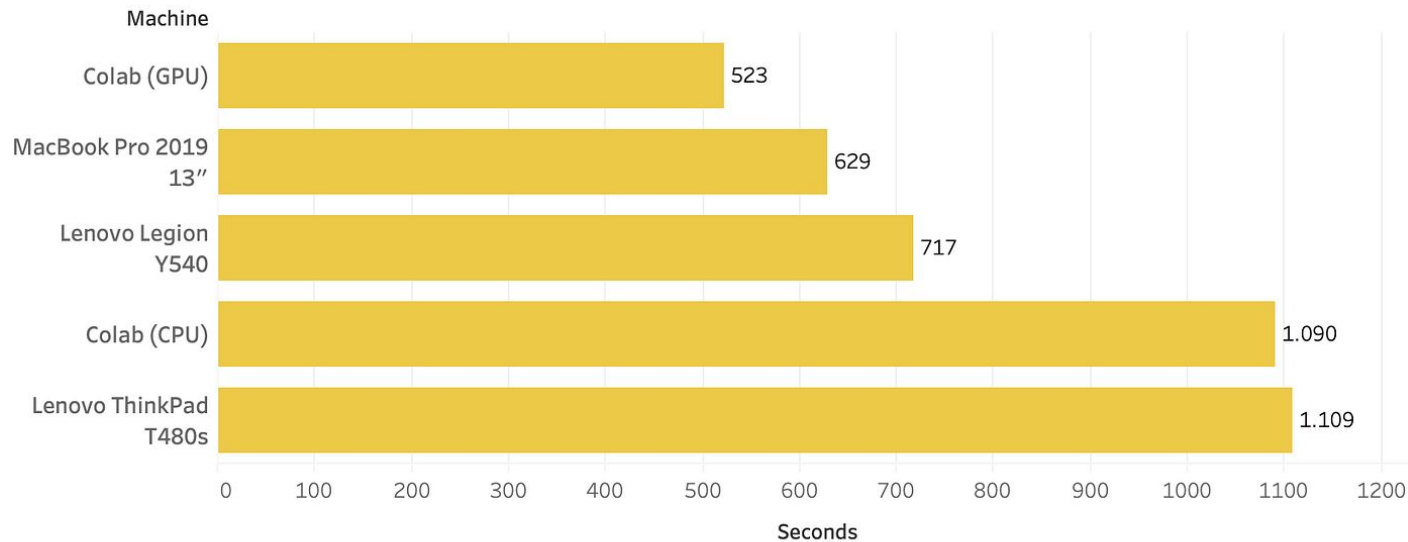
```
pip3 install notebook &&\  
jupyter notebook
```

[jupyter-notebook.readthedocs.io](https://jupyter-notebook.readthedocs.io)



[colab.google/notebooks/](https://colab.google/notebooks/)

## Fashion-MNIST - Training Time (seconds) Comparison



Sum of Seconds for each Machine.

# Python3

+ Librerías

## Necesitamos...

1. trabajar con números, vectores y matrices en formas complejas y prácticas

[numpy.org](https://numpy.org)

2. crear y manipular conjuntos de datos de forma intuitiva, y poder operar con los mismos

[pandas.pydata.org](https://pandas.pydata.org)

3. utilizar elementos básicos y avanzados de machine learning de forma de evitarnos programar tanto

[scikit-learn.org](https://scikit-learn.org)

4. graficar elementos, imágenes o métricas

[matplotlib.org](https://matplotlib.org)

# Precalentemos...

## Generando Datos:

queremos generar datos, o puntos en un espacio  $n$ -dimensional, con distribuciones específicas que nos permitan testear nuestros métodos de Machine Learning.

# Precalentemos...

## Diagonales

Programa una función que genere  $n$  puntos de  $d$  inputs (es decir  $D$ -dimensionales) de valores **reales**, cuyo output sea un valor **binario** que corresponde a la clase a la que pertenece el elemento.

Se pide que la **Clase 1** sean puntos generados al azar con distribución normal<sup>1</sup>, centrada en  $(1, 1, \dots, 1)$  y matriz de covarianza diagonal, con desviación estándar  $C * \sqrt{d}$  (donde  $C$  está dado por el usuario). La **Clase 0** debe ser igual pero centrada en  $(-1, -1, \dots, -1)$ .

Ambas clases deben contener  $n/2$  elementos.

1. <https://cs229.stanford.edu/section/gaussians.pdf>



# Precalentemos...

## Paralelos

Programa una función que genere  $n$  puntos de  $d$  inputs (es decir  $D$ -dimensionales) de valores **reales**, cuyo output sea un valor **binario** que corresponde a la clase a la que pertenece el elemento.

Se pide que la **Clase 1** sean puntos generados al azar con distribución normal<sup>1</sup>, centrada en  $(1, 0, \dots, 0)$  y matriz de covarianza diagonal, con desviación estándar  $C$  (donde  $\mathbf{C}$  está dado por el usuario).

La **Clase 0** debe ser igual pero centrada en  $(-1, 0, \dots, 0)$ .

Ambas clases deben contener  $n/2$  elementos.

1. <https://cs229.stanford.edu/section/gaussians.pdf>

# Precalentemos...

## Espirales Anidadas

Programa una función que genere **n** puntos de **2** inputs (valores *x* e *y*) de valores **reales**, cuyo output sea un valor **binario** que corresponde a la clase a la que pertenece el elemento.

Se pide que sean puntos generados al azar con distribución uniforme (en el sistema de referencia *x/y*) dentro de un círculo de radio 1.

La Clase 0 son aquellos puntos que se encuentran entre las curvas:

$$\rho = \theta / (4 * \pi) \quad \text{y} \quad \rho = (\theta + \pi) / (4 * \pi) \quad (\text{en coord. polares})$$

La **Clase 1** son los puntos restantes.

Ambas clases deben contener  $n/2$  elementos.

# Precalentemos...

## Probemos y Grafiquemos

Para verificar la generación de “Diagonales” y “Paralelas” genere y grafique conjuntos con:

- $d=2$ ,  $n=200$  y  $C=0.75$
- $d=4$ ,  $n=5000$  y  $C=2.00$

y verifique en el código que las medias y desviaciones estándar sean correctas.

Para verificar la generación de “Espirales” genere y grafique un conjunto con

- $n=2000$

compárelo con el que se muestra a continuación.

