

# Aplicaciones del Aprendizaje automático con Python

Dra. Jessica Beltrán Márquez

[www.jessicabeltran.net](http://www.jessicabeltran.net)

# Objetivo

El objetivo de esta presentación y las actividades que la acompañan es brindar una introducción y conceptos básicos relacionados con el aprendizaje automático en el idioma español.

También se busca que se conozcan herramientas útiles para usar algoritmos de clasificación. Varias de las actividades han sido basadas en trabajos de otras personas que han sido referenciadas, como tutoriales del software Orange data mining y Tensorflow.

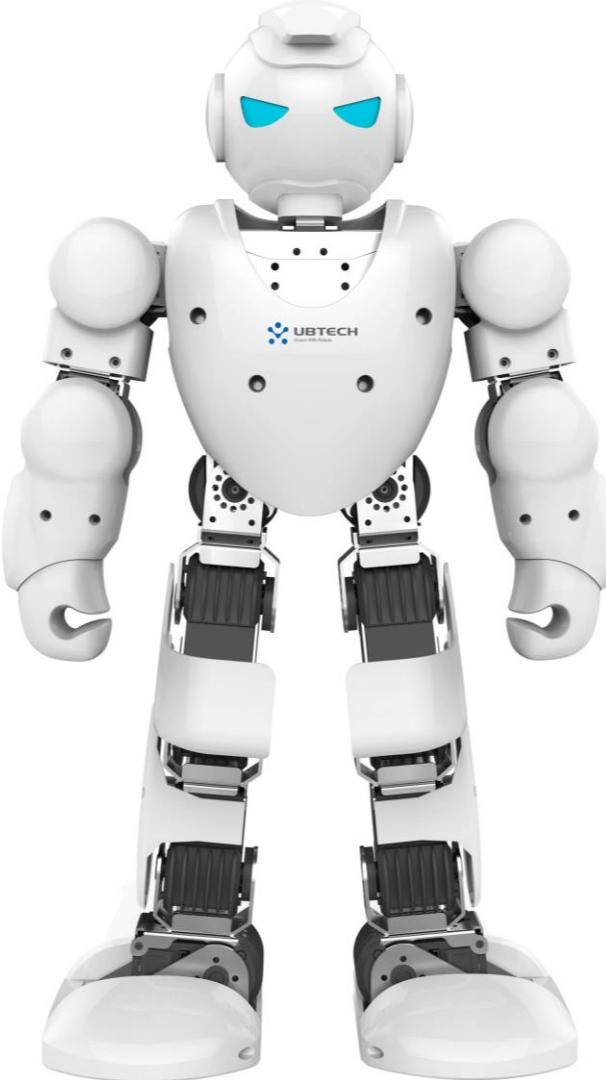
Con este curso no serás experto, pero espero que sirva para que tengas mayor interés en esta área y conozcas que existen herramientas para que puedas prepararte, como cuadernos de jupyter, etc.

Otra motivación de esta presentación es el tener más información en español, ya que la mayoría está en inglés.

Cualquier sugerencia, corrección o comentario por favor envíame un correo.

-Jessica

# Inteligencia artificial



- La inteligencia artificial (IA) se refiere a inteligencia en las máquinas.
- En ciencias computacionales, La IA enfatiza que las máquinas imiten funciones cognitivas de los seres humanos como aprender o resolver problemas.

# Aprendizaje automático

- El aprendizaje automático es una rama de la inteligencia artificial que provee a los sistemas la capacidad de aprender automáticamente y aprender de la experiencia sin ser específicamente programados para ello.
- De forma más concreta, se trata de crear programas capaces de generalizar comportamientos a partir de una información suministrada en forma de ejemplos.



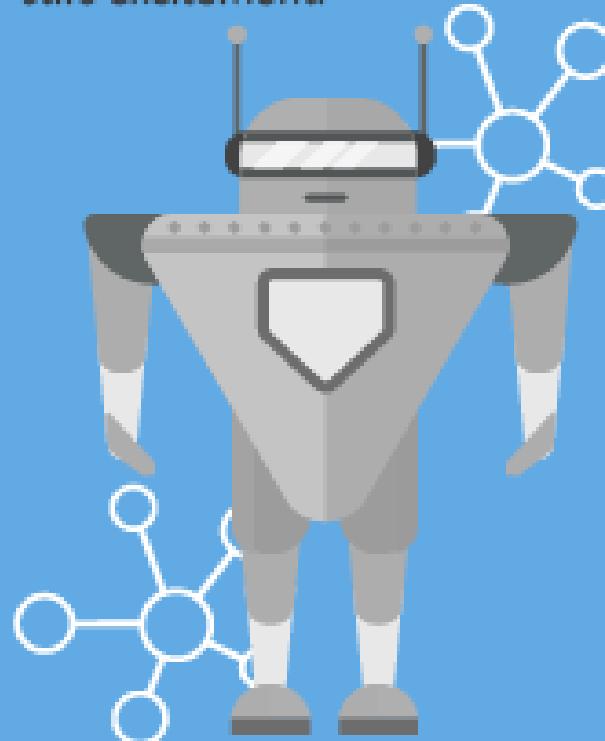
# Inteligencia artificial y Aprendizaje automático

**La inteligencia artificial es un concepto amplio sobre máquinas que son capaces de realizar tareas que consideramos “inteligentes”.**

**El aprendizaje automático es una aplicación de la IA que se basa en la idea de que les proveeamos de datos a las máquinas y dejar que aprendan por si mismas.**

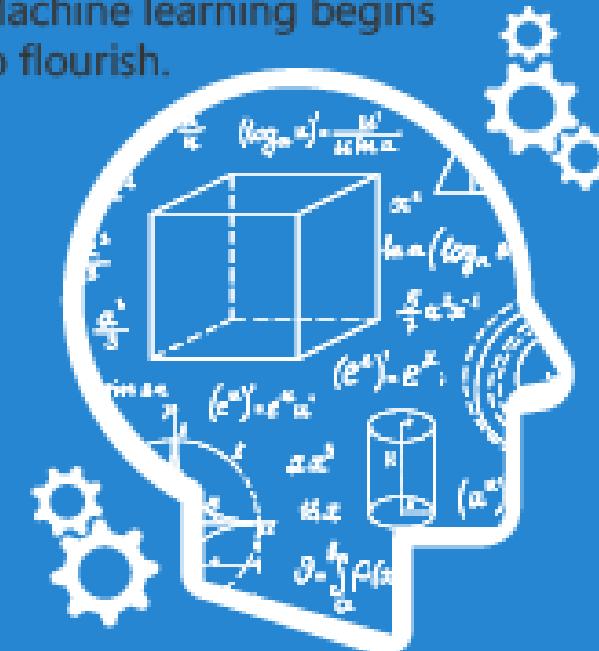
# ARTIFICIAL INTELLIGENCE

Early artificial intelligence stirs excitement.



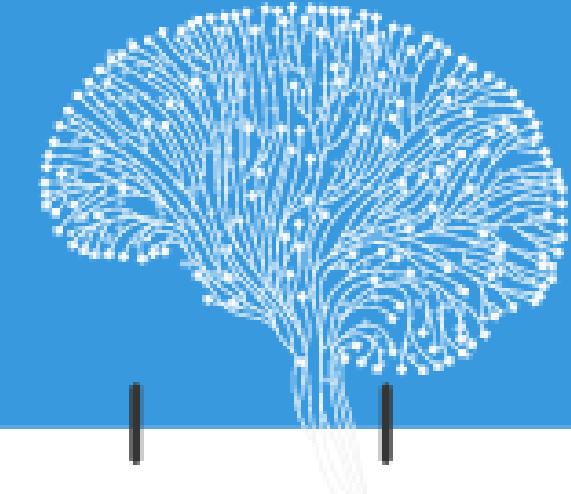
## MACHINE LEARNING

Machine learning begins to flourish.



## DEEP LEARNING

Deep learning breakthroughs drive AI boom.



1950's

1960's

1970's

1980's

1990's

2000's

2010's

Since an early flush of optimism in the 1950's, smaller subsets of artificial intelligence - first machine learning, then deep learning, a subset of machine learning - have created ever larger disruptions.

<https://www.alltechbuzz.net/impact-of-ai-and-ml/>

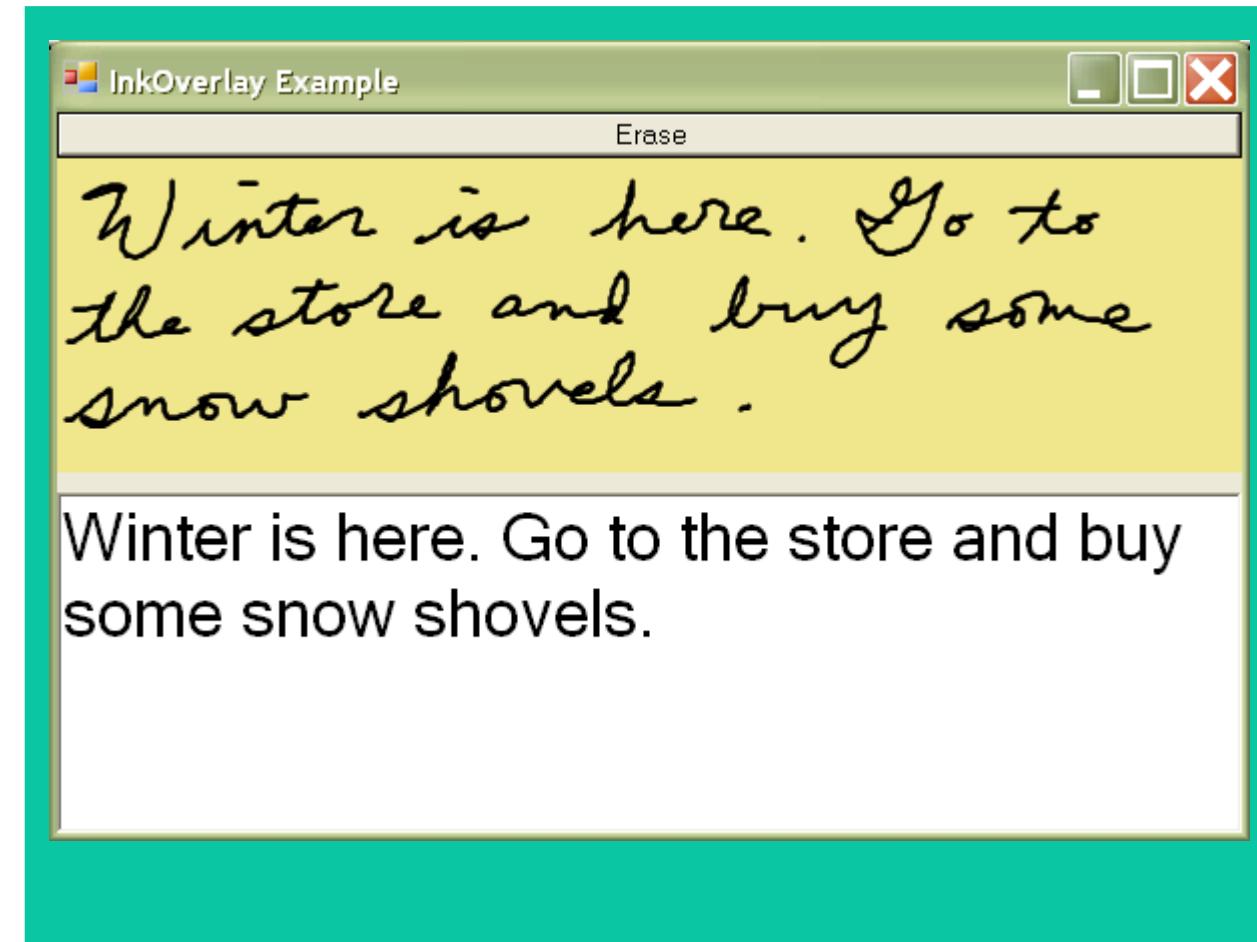
# Escritura a mano (Handwritting)

A screenshot of a software application window titled "History Options Help". The top panel contains a mathematical formula:  $f(z) = \frac{1}{2\pi} \int_0^{2\pi} u(e^{i\psi}) \frac{e^{i\psi} + z}{e^{i\psi} - z} d\psi, |z| < 1$ . Below it, a yellow notepad-like area shows the same formula, but with "f(z)" written as "f(z) =". The "Insert" button is visible at the bottom right of the notepad area.

# Escritura a mano (Handwriting)

Diversidad en las letras escritas a mano

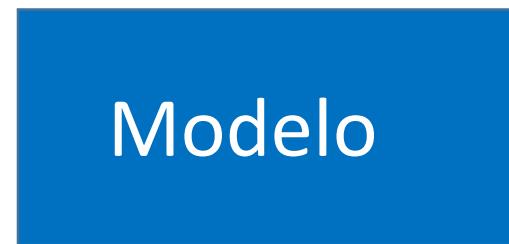
0 0 0 0 0 0 0 0 0 0 0 0 0  
1 1 1 1 1 1 1 1 1 1 1 1 1  
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  
3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  
5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5  
6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6  
7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7  
8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8  
9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9



# Escritura a mano (Handwritting)

Entrada

0 4 3 1



Salida

0 4 3 1

# Reconocimiento del habla (Speech recognition)

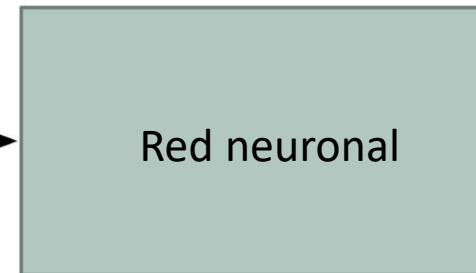


Entrada

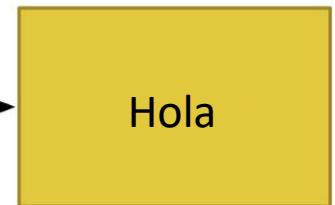


Onda de sonido de una persona diciendo “Hola”

Salida



Red neuronal

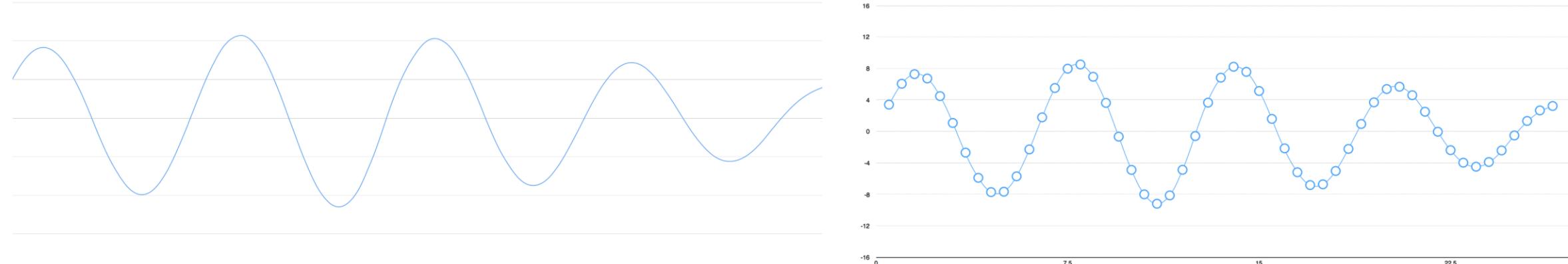


Hola

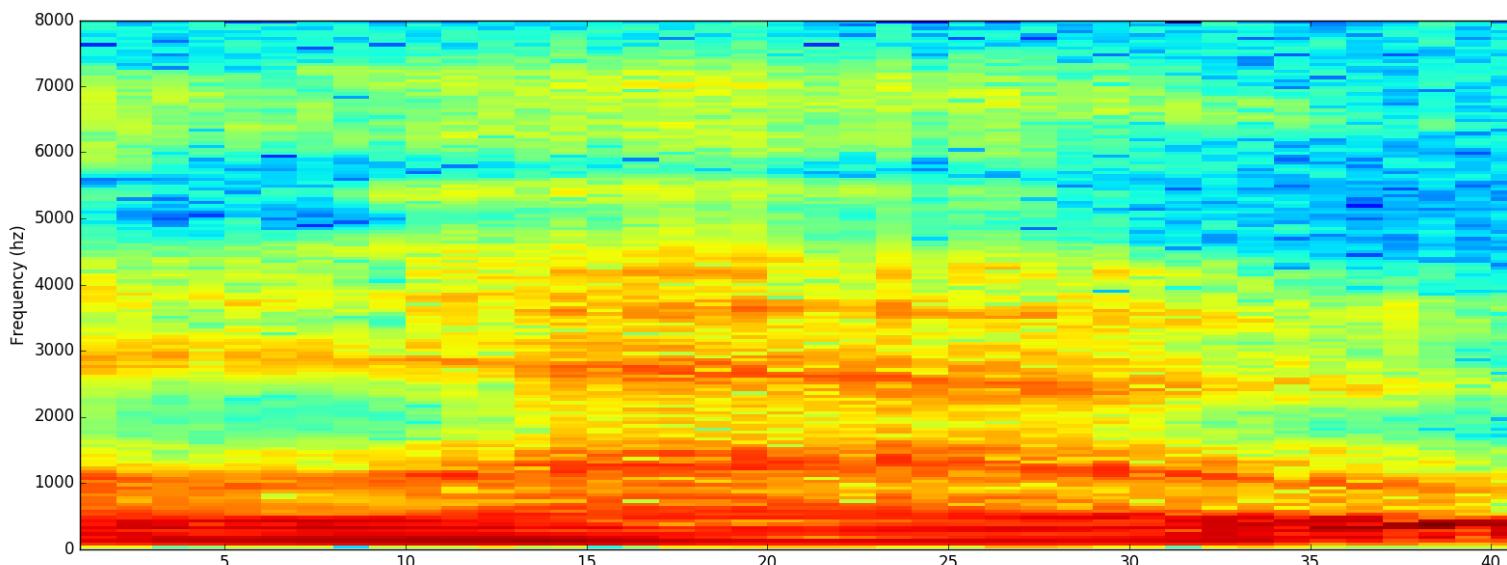
Palabra identificada

**Alexa**  
**Toca música de los Beatles**

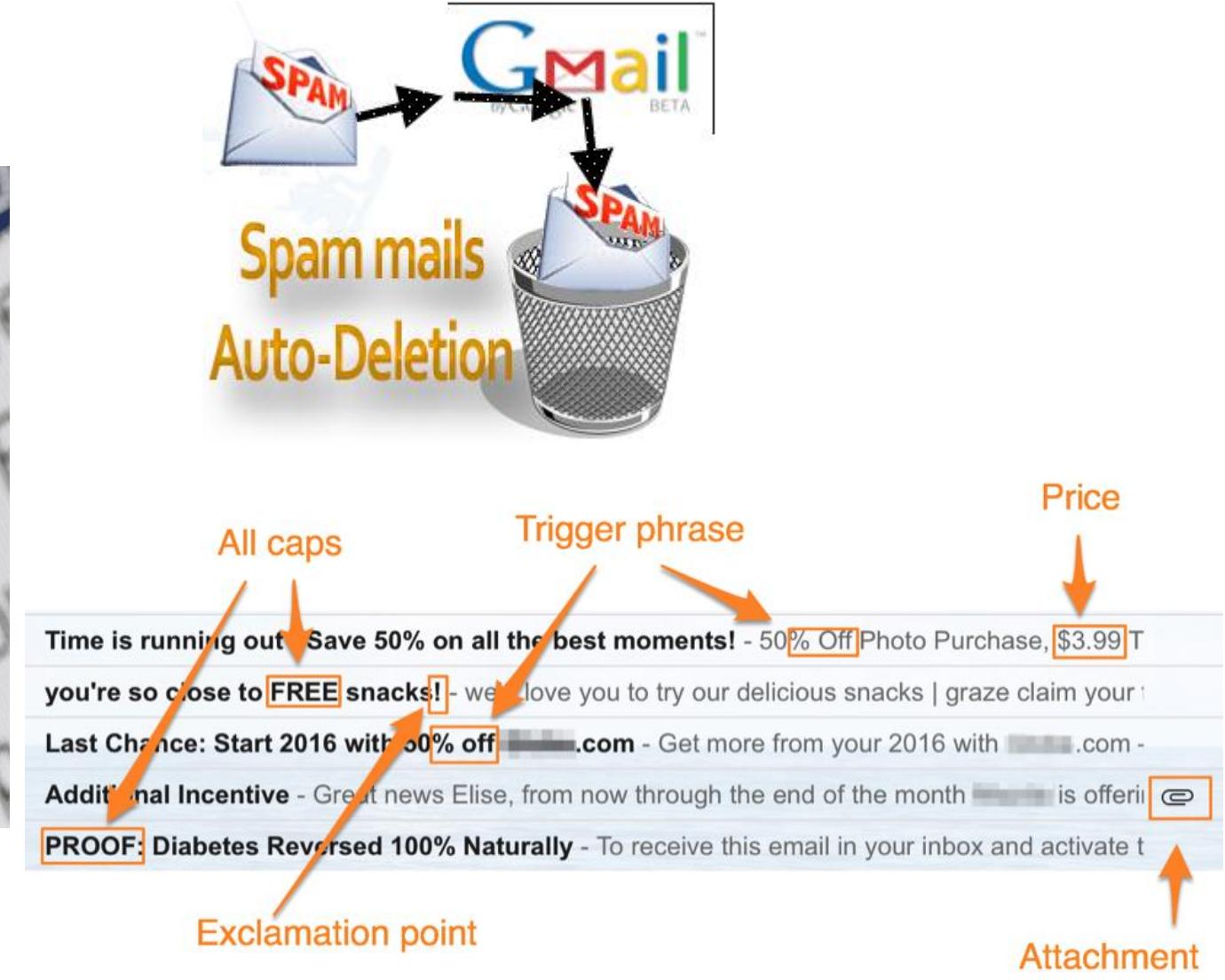
# Reconocimiento del habla (Speech recognition)



```
[-1274, -1252, -1160, -986, -792, -692, -614, -429, -286, -134, -57, -41, -169, -456, -450, -541, -761, -1067, -1231, -1047, -952, -645, -489, -- , -397, -212, 193, 114, -17, -110, 128, 261, 198, 390, 461, 772, 948, 1451, 1974, 2624, 3793, 4968, 5939, 6057, 6581, 7302, 7640, 7223, 6119, 544 4820, 4353, 3611, 2740, 2004, 1349, 1178, 1085, 901, 301, -262, -499, -488, -707, -1406, -1997, -2377, -2494, -2605, -2675, -2627, -2500, -2148 1648, -970, -364, 13, 260, 494, 788, 1011, 938, 717, 507, 323, 324, 325, 350, 103, -113, 64, 176, 93, -249, -461, -606, -909, -1159, -1307, -1544]
```



# Detección de correo basura



# HERRAMIENTAS

# Anaconda

- Anaconda es una distribución de Python y R de código libre usada en cómputo científico.



# Numpy

- Es una biblioteca para Python que soporta el manejo de arreglos y matrices multidimensionales de gran tamaño. También incluye una colección de funciones matemáticas para operar sobre los arreglos y matrices.



<http://www.numpy.org/>

Para instalar en Python:  
`pip install numpy`

Esta biblioteca ya está  
incluida en Anaconda

# Jupyter

- Es una aplicación web de código libre que permite crear y compartir documentos con código vivo, ecuaciones, visualizaciones y texto explicatorio.



<http://jupyter.org/>

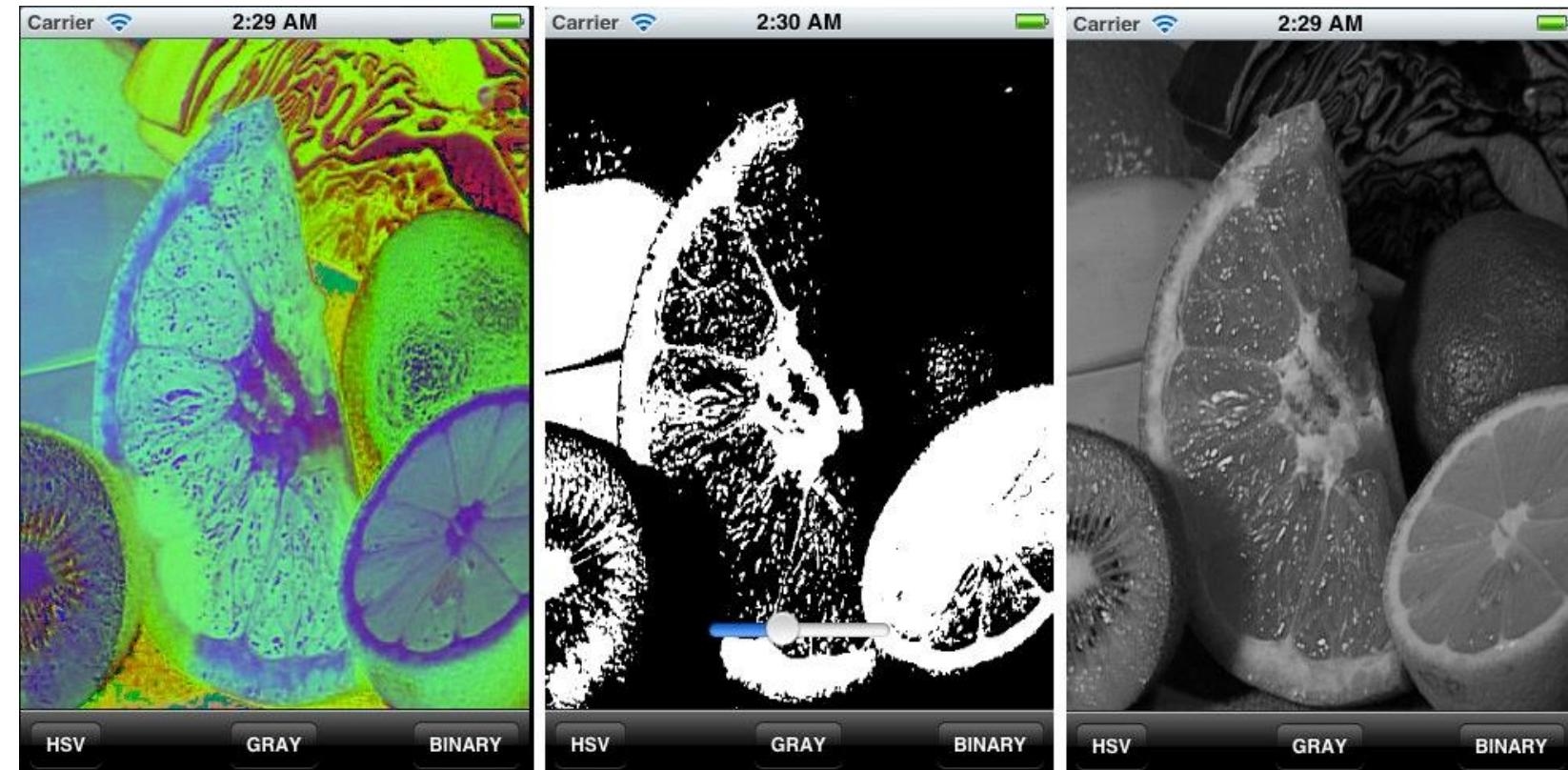
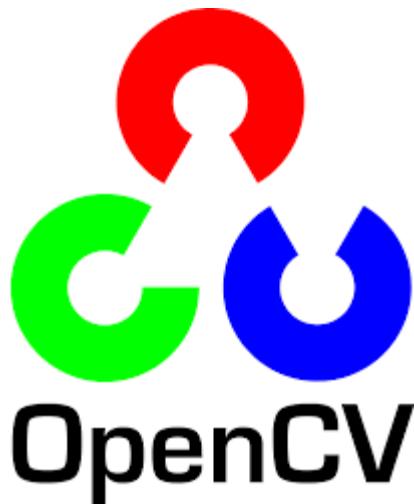
Para instalar en Python:  
`pip install jupyter`

Está incluido en Anaconda

# OpenCV

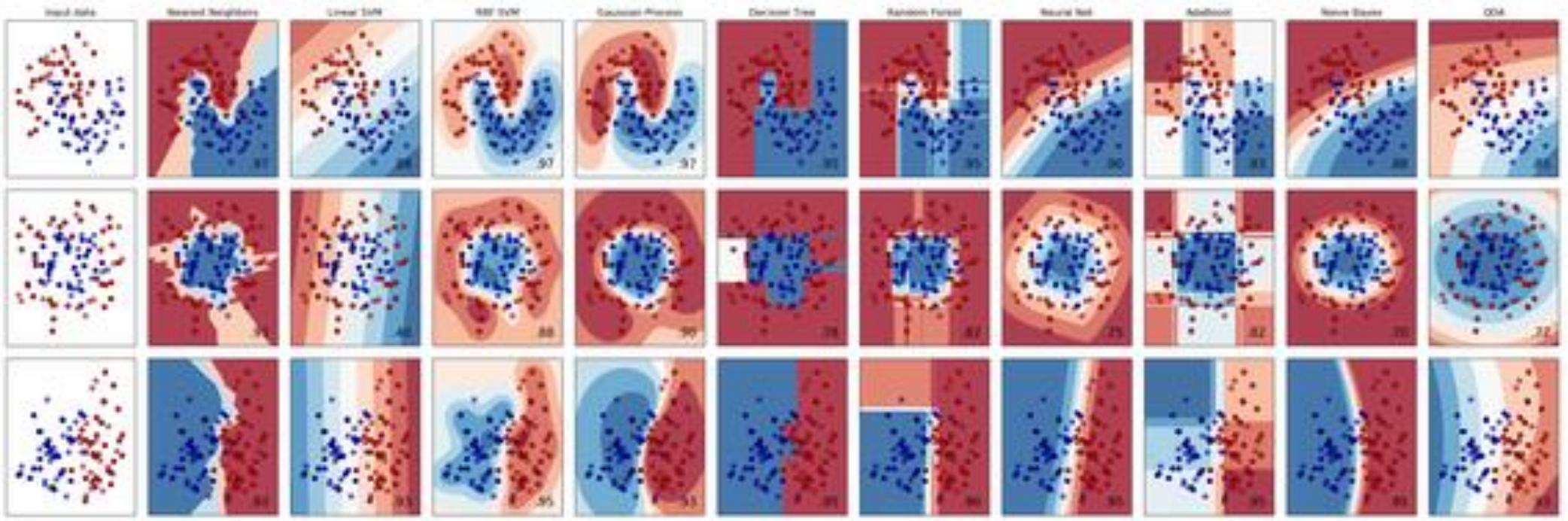
- OpenCV (Open Source Computer Vision Library) es una biblioteca de visión artificial originalmente desarrollada por Intel con licencia BSD (libre para uso académico y comercial).

<https://opencv.org/>



# Scikit

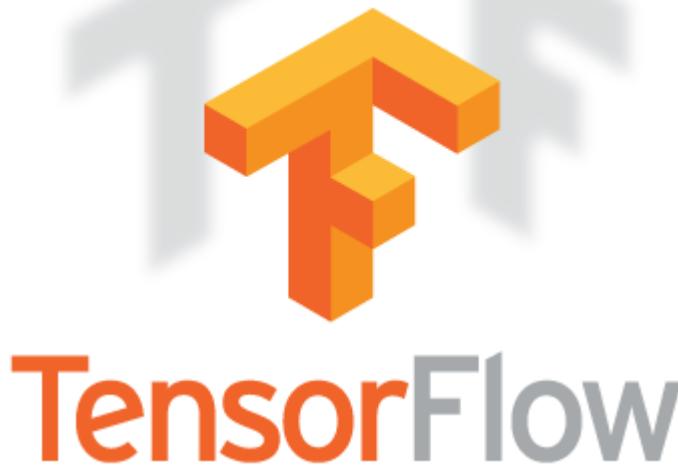
- Es una biblioteca libre en Python para machine learning.



Para instalar en Python:  
pip install sklearn

# Tensor flow

- Es una biblioteca de Código libre para el aprendizaje automático desarrollado por Google.



<https://www.tensorflow.org/>

Para instalar en Python:

`pip install --ignore-installed --upgrade tensorflow`

# Matplotlib

- Es una biblioteca para la generación de gráficos a partir de datos contenidos en listas o arrays en el lenguaje de programación Python y su extensión matemática NumPy.

<https://matplotlib.org/>

Para instalar en Python:  
pip install matplotlib



# Orange

- Herramienta interactiva para machine learning y visualización de datos.

<https://orange.biolab.si>



Está incluído en Anaconda Navigator

# Instalación de Anaconda

# Anaconda

- Anaconda es una distribución de Python y R de código libre usada en cómputo científico.



# Anaconda Distribution

The World's Most Popular Python/R Data Science Platform

[Download](#)

The open-source [Anaconda Distribution](#) is the easiest way to perform Python/R data science and machine learning on Linux, Windows, and Mac OS X. With over 15 million users worldwide, it is the industry standard for developing, testing, and training on a single machine, enabling *individual data scientists* to:

- Quickly download 1,500+ Python/R data science packages
- Manage libraries, dependencies, and environments with Conda
- Develop and train machine learning and deep learning models with scikit-learn, TensorFlow, and Theano
- Analyze data with scalability and performance with Dask, NumPy, pandas, and Numba
- Visualize results with Matplotlib, Bokeh, Datashader, and Holoviews



Windows



macOS



Linux

## Anaconda 2019.07 for macOS Installer

## Anaconda 2019.07 for macOS Installer

Descargar la versión 3.7

### Python 3.7 version

Download

64-Bit Graphical Installer (653 MB)

64-Bit Command Line Installer (435 MB)

### Python 2.7 version

Download

64-Bit Graphical Installer (634 MB)

64-Bit Command Line Installer (408 MB )

## Get Started with Anaconda Distribution

### Documentation

Installation and user  
guide for Anaconda  
Distribution 5

[Read More](#)

### Anaconda Blog

News, software  
releases, and  
developer best  
practices [Read More](#)

### Community Support

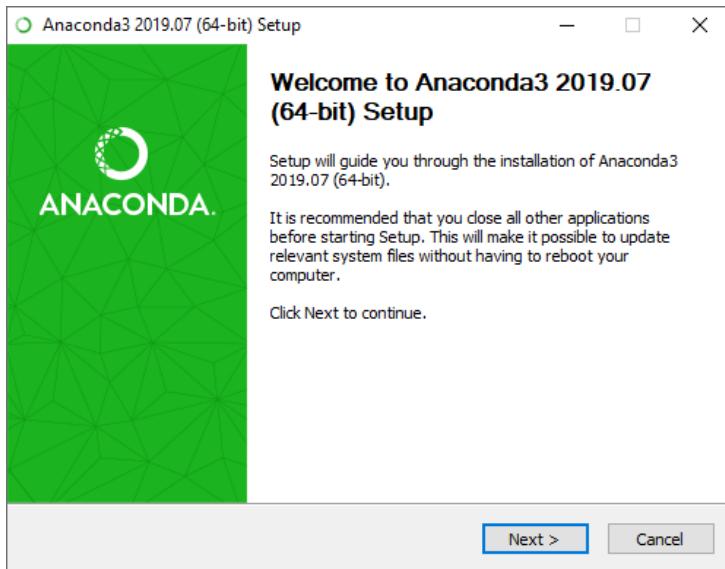
Solutions and  
knowledge from the  
community [Read More](#)

### Anaconda Webinars

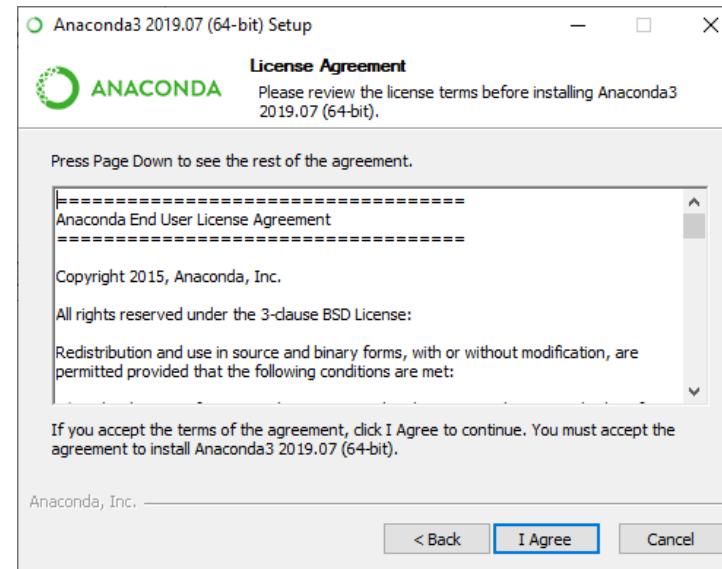
Industry trends and  
tutorials from  
Anaconda [Read More](#)

### Anaconda Training

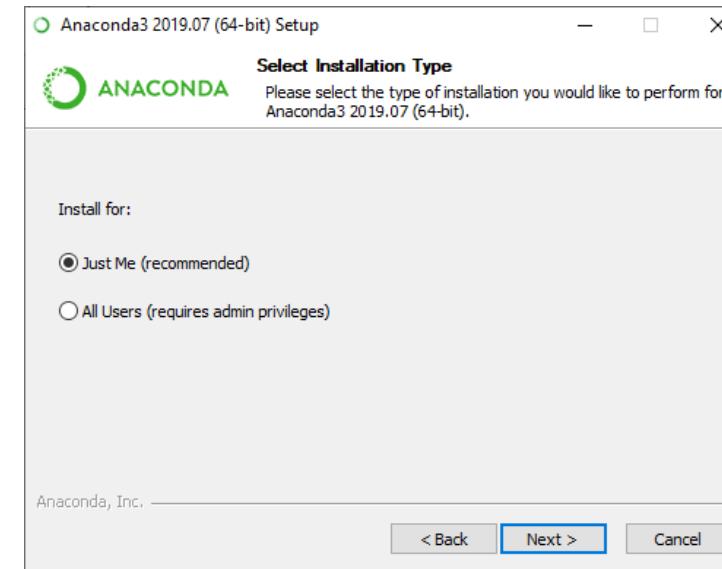
Learn Python for Data  
Science with  
DataCamp  
  
[Start Learning](#)



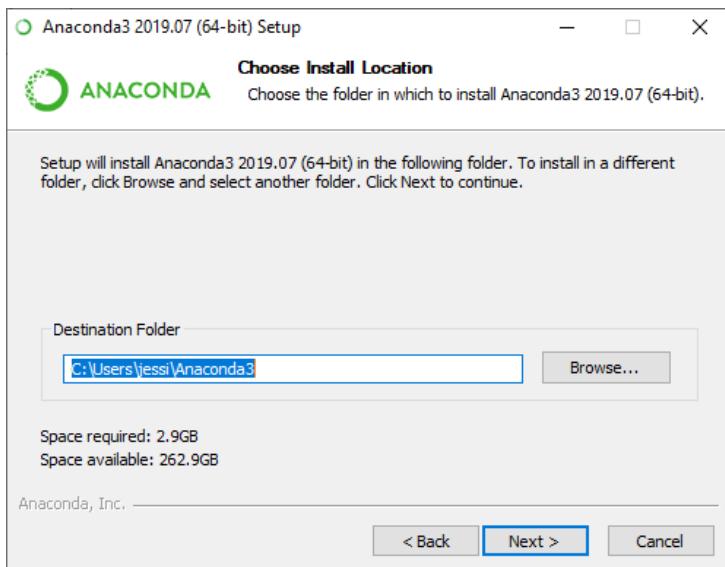
Paso 1



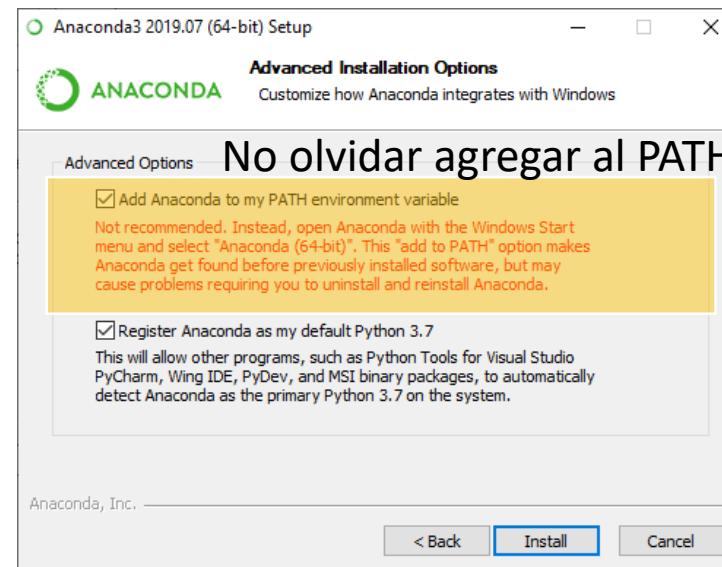
Paso 2



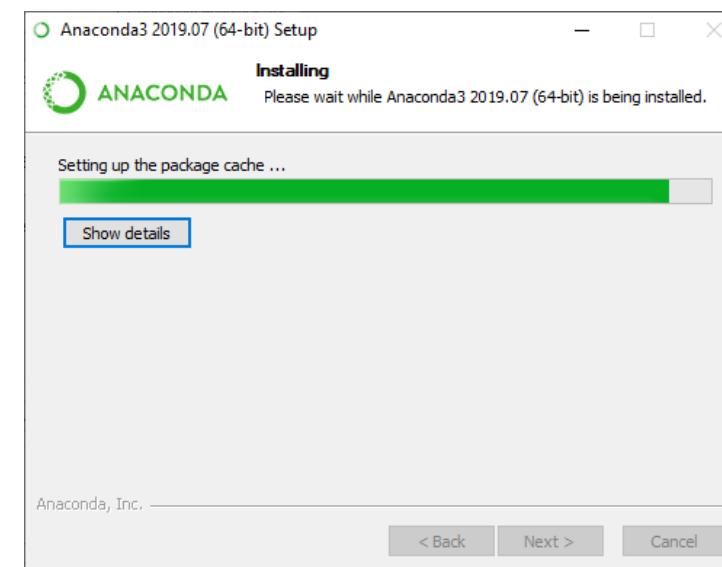
Paso 3

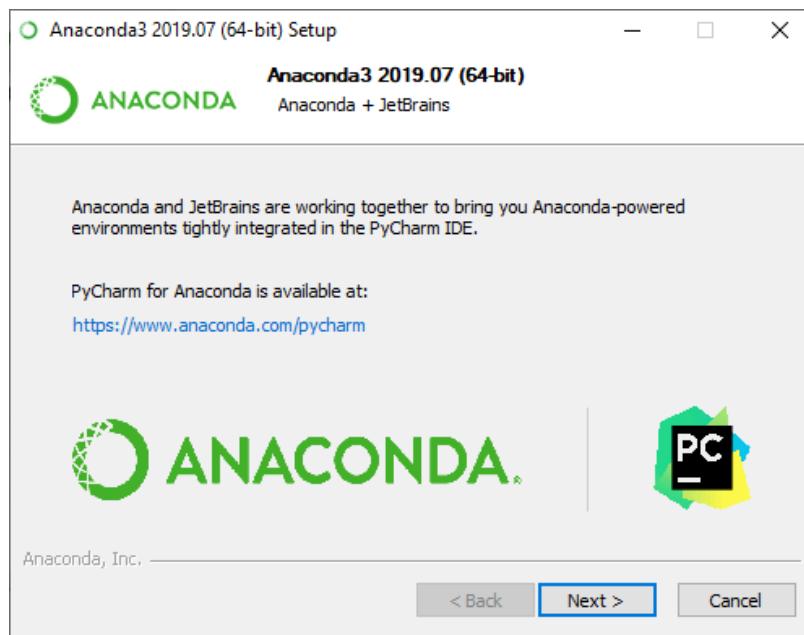


Paso 4

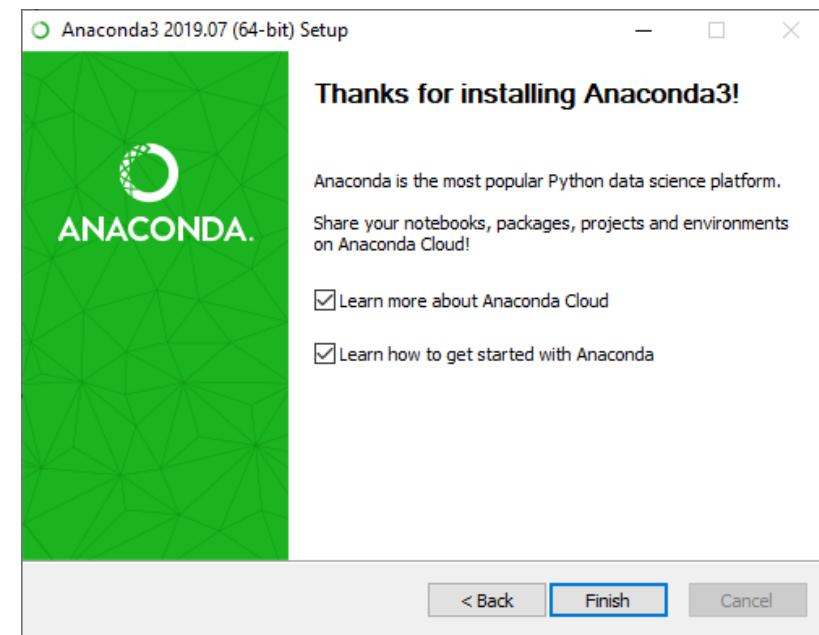


Paso 5





Paso 7

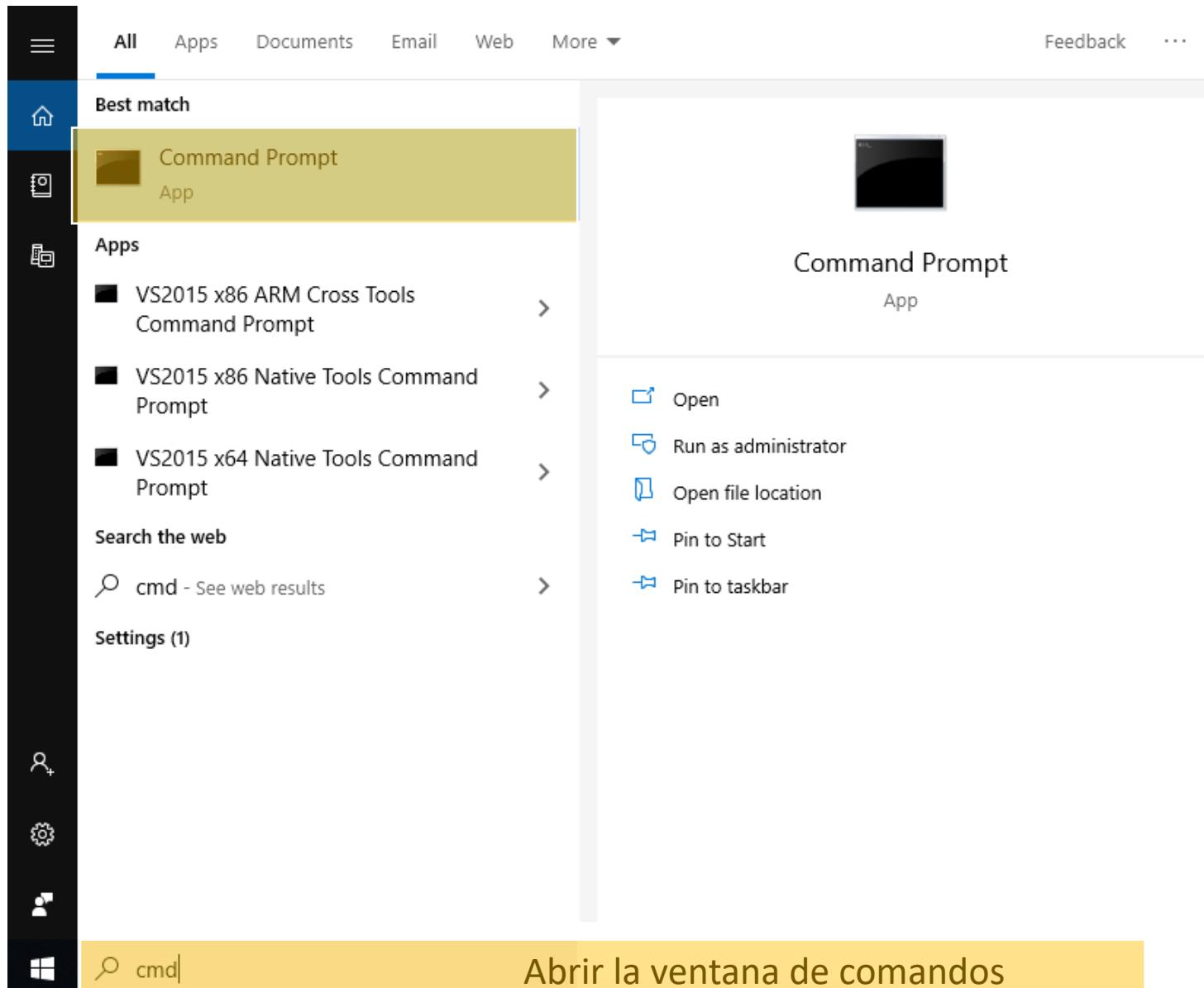


Paso 8

# Probar instalación de Python

## Abrir la ventana de comandos de Windows

Paso 2



Paso 1

# Crear un nuevo ambiente

Este directorio será distinto en tu computadora



```
Command Prompt
Microsoft Windows [Version 10.0.17763.678]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\jessi> Directorio actual
```

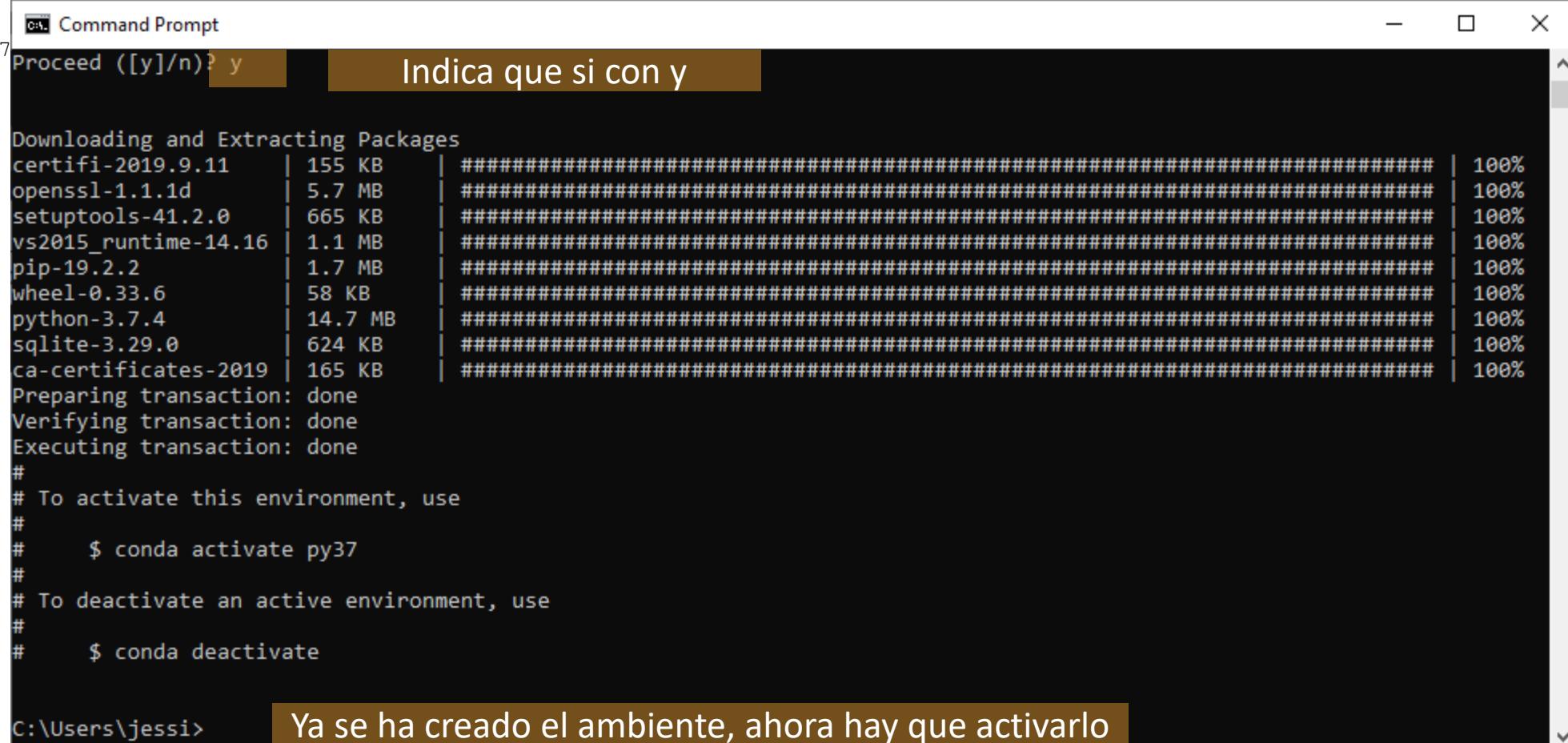
## Paso 1

Puedes elegir otro nombre  
Por ejemplo:  
--name cursoML

```
conda create --name py37 python=3.7
```



```
Command Prompt
C:\Users\jessi> conda create --name py37 python=3.7 - Create an environment
```



```
Command Prompt
Proceed ([y]/n)? y Indica que si con y

Downloading and Extracting Packages
certifi-2019.9.11 | 155 KB | #####| 100%
openssl-1.1.1d | 5.7 MB | #####| 100%
setuptools-41.2.0 | 665 KB | #####| 100%
vs2015_runtime-14.16 | 1.1 MB | #####| 100%
pip-19.2.2 | 1.7 MB | #####| 100%
wheel-0.33.6 | 58 KB | #####| 100%
python-3.7.4 | 14.7 MB | #####| 100%
sqlite-3.29.0 | 624 KB | #####| 100%
ca-certificates-2019 | 165 KB | #####| 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate py37
#
# To deactivate an active environment, use
#
#     $ conda deactivate

C:\Users\jessi> Ya se ha creado el ambiente, ahora hay que activarlo
```

## Activar el nuevo ambiente

### Paso 1

Si elegiste el nombre cursoML sería  
conda activar cursoML



Command Prompt

```
C:\Users\jessi>conda activate py37
(py37) C:\Users\jessi>
```

Activar ambiente py37

The screenshot shows a Windows Command Prompt window titled "Command Prompt". The command "conda activate py37" is entered in the prompt. The output "(py37) C:\Users\jessi>" indicates that the "py37" environment has been activated. A callout box labeled "Activar ambiente py37" points to the output line.

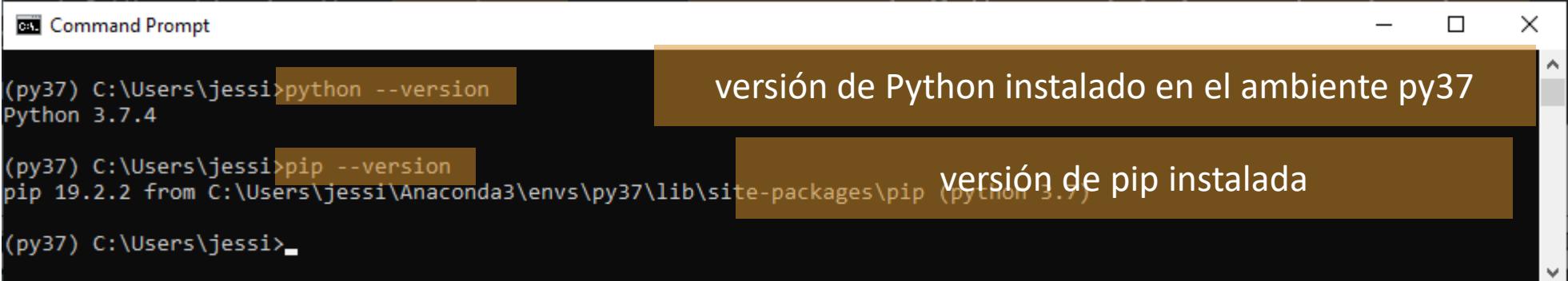
conda activate py37

Anota las siguientes versiones instaladas en tu computadora

### Paso 2

Revisar la versión de Python  
instalada en el ambiente

python --version



Command Prompt

```
(py37) C:\Users\jessi>python --version
Python 3.7.4
```

versión de Python instalado en el ambiente py37

```
(py37) C:\Users\jessi>pip --version
pip 19.2.2 from C:\Users\jessi\Anaconda3\envs\py37\lib\site-packages\pip (python 3.7)
```

versión de pip instalada

```
(py37) C:\Users\jessi>
```

The screenshot shows a Windows Command Prompt window titled "Command Prompt". The command "python --version" is entered, and the output "Python 3.7.4" is displayed. Another command, "pip --version", is also shown, with its output "pip 19.2.2 from C:\Users\jessi\Anaconda3\envs\py37\lib\site-packages\pip (python 3.7)" highlighted. Callout boxes labeled "versión de Python instalado en el ambiente py37" and "versión de pip instalada" point to the respective outputs.

### Paso 3

Revisar la versión de pip  
instalada en el ambiente

pip --version

## Entrar al interprete de Python y probarlo

### Paso 1

Entrar al interprete de Python

```
c:\ Command Prompt - python
(py37) C:\Users\jessi>python
Python 3.7.4 (default, Aug 9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> -
```

Entramos al interprete de Python  
Nótese que cambia el *prompt*

Información de Python instalado en el ambiente base

### Paso 2

Probar sumas

```
c:\ Command Prompt - python
>>> 5+2
7
>>>
```

Calculadora

### Paso 3

Probar restas

```
c:\ Command Prompt - python
>>> 89-7
82
>>> -
```

Calculadora

### Paso 4

Salir

```
c:\ Command Prompt
>>> exit()
```

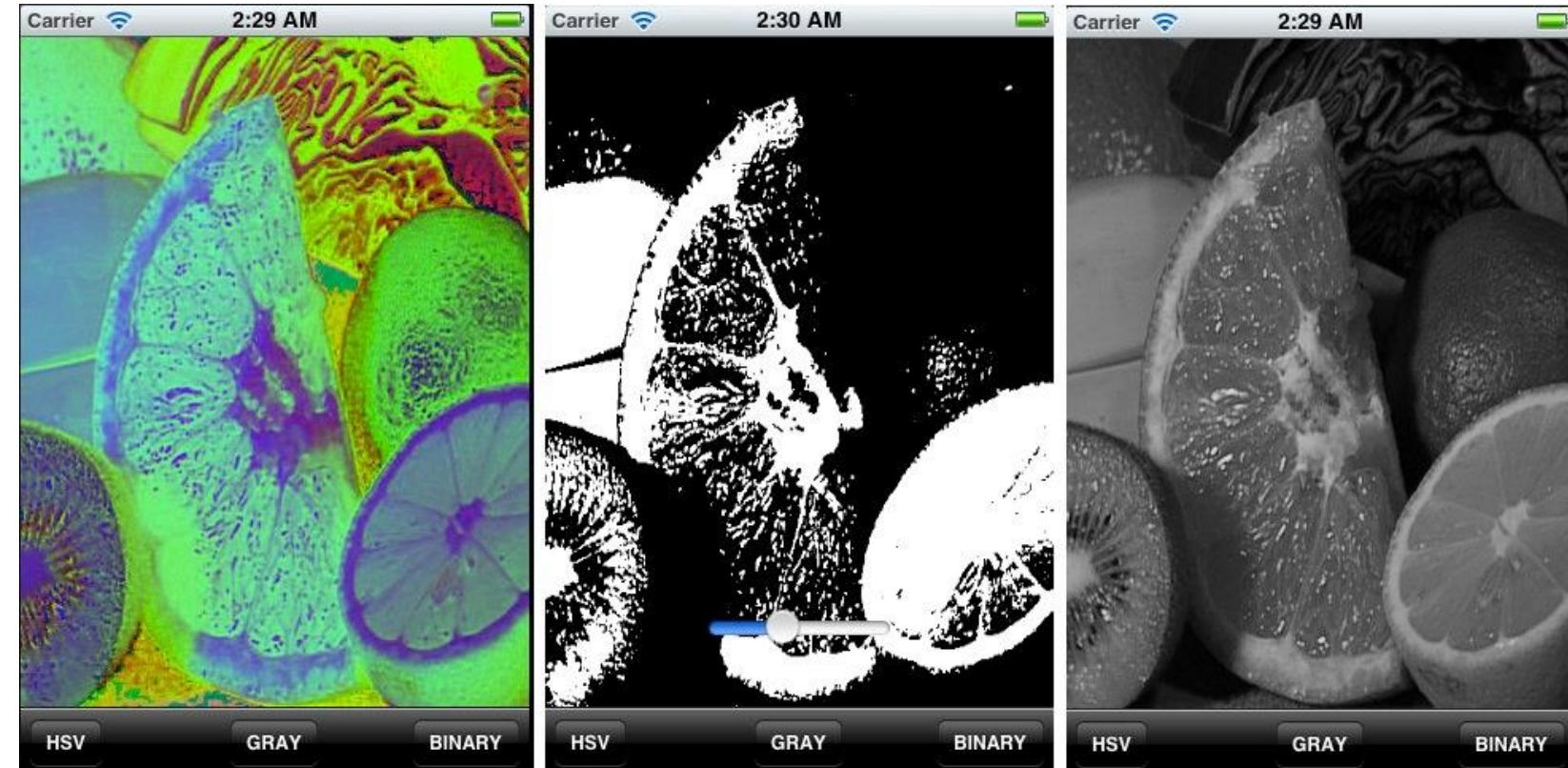
Salirse del interprete de Python

# Instalar openCV

# OpenCV

- OpenCV (Open Source Computer Vision Library) es una biblioteca de visión artificial originalmente desarrollada por Intel con licencia BSD (libre para uso académico y comercial).

<https://opencv.org/>



# Instalar e importar la biblioteca openCV

## Paso 1

Instalar openCV

conda install -c conda-forge opencv

## Paso 2

Proceder con la instalación con y

## Paso 3

Entrar al interprete de Python

## Paso 4

Importar la biblioteca y revisar si no hay errores

## Paso 5

Revisar versión instalada

cv2.\_\_version\_\_

## Paso 6

Salirse del interprete

Command Prompt  
(py37) C:\Users\jessi>conda install -c conda-forge opencv

Instalar OpenCV

Command Prompt - conda install -c conda-forge opencv  
Proceed ([y]/n)?

Command Prompt - conda install -c conda-forge opencv - python  
(py37) C:\Users\jessi>python

Entrar a Python  
Python 3.7.4 (default, Aug 9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32  
Type "help", "copyright", "credits" or "license" for more information.  
>>> import cv2

Dentro del interprete revisar que se importe la biblioteca de openCV

Si no marca error, se instaló correctamente

Command Prompt - conda install -c conda-forge opencv - python  
>>> cv2.\_\_version\_\_  
'4.1.1'  
>>>

Anotar la versión

Command Prompt  
>>> exit()

Salirse del interprete de Python

# Instalar Matplotlib

# Matplotlib

- Es una biblioteca para la generación de gráficos a partir de datos contenidos en listas o arrays en el lenguaje de programación Python y su extensión matemática NumPy.

<https://matplotlib.org/>

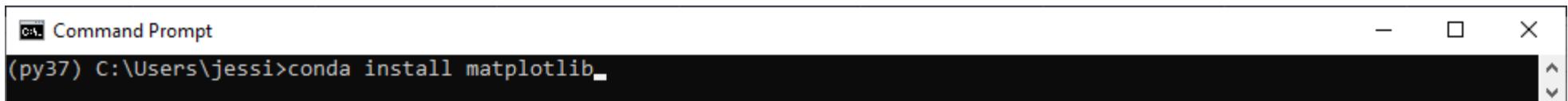


# Instalar Matplotlib

## Paso 1

Instalar Matplotlib

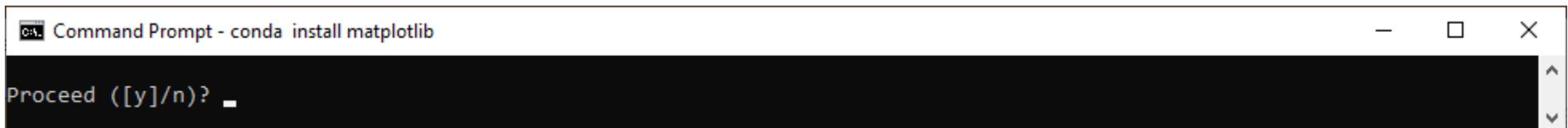
conda install matplotlib



Command Prompt  
(py37) C:\Users\jessi>conda install matplotlib

## Paso 2

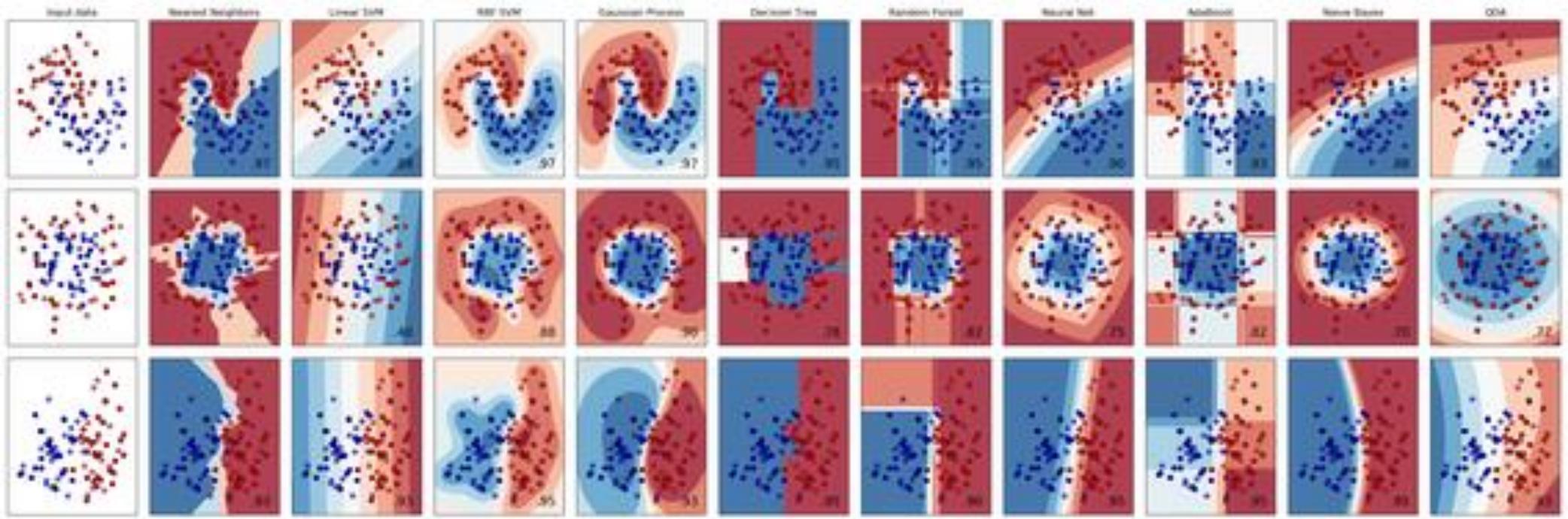
Proceder con la instalación con y



Command Prompt - conda install matplotlib  
Proceed ([y]/n)? -

# Instalar Scikit

- Es una biblioteca libre en Python para machine learning.



# Instalar Matplotlib

## Paso 1

### Instalar Scikit

```
conda install scikit-learn
```



A screenshot of a Windows Command Prompt window titled "Command Prompt". The window shows the command "conda install scikit-learn" being typed. The prompt "(py37) C:\Users\jessi>" is visible, followed by the command text.

## Paso 2

### Proceder con la instalación con y



A screenshot of a Windows Command Prompt window titled "Command Prompt - conda install scikit-learn". The window shows the confirmation prompt "Proceed ([y]/n)? y" after the installation command was run.

# Instalar Jupyter

# Jupyter

- Es una aplicación web de código libre que permite crear y compartir documentos con código vivo, ecuaciones, visualizaciones y texto explicatorio.



# Instalar jupyter

## Paso 1

Instalar jupyter

conda install jupyter

```
C:\ Command Prompt - conda install -c conda-forge opencv
(py37) C:\Users\jessi>conda install jupyter
```

## Paso 2

Proceder con la instalación con y

```
C:\ Command Prompt - conda install -c conda-forge opencv - conda install jupyter
Proceed ([y]/n)?
```

# Instalar TensorFlow

# Tensor flow

- Es una biblioteca de Código libre para el aprendizaje automático desarrollado por Google.



<https://www.tensorflow.org/>

# Instalar Tensorflow

## Paso 1

### Instalar Tensorflow

```
pip install tensorflow==2.0.0-rc1
```

## Paso 2

### Entrar al interprete

## Paso 3

### Importar la biblioteca y revisar que no hay errores

The screenshot shows two Command Prompt windows side-by-side.

**Top Window:** Title: "Command Prompt - pip install tensorflow==2.0.0-rc1". It displays the command being run and the output of the pip install process for TensorFlow version 2.0.0-rc1, including dependencies like tensorflow and grpcio.

**Bottom Window:** Title: "Command Prompt - conda install -c conda-forge opencv - conda install scikit-learn - pip install tensorflow". It shows the command to install TensorFlow via pip. Below it, the Python interpreter is running, showing the Anaconda distribution details and a command prompt where "import tensorflow as tf" is typed.

**Annotations:**

- A yellow box highlights the command "pip install tensorflow==2.0.0-rc1" in the top window.
- A yellow box highlights the text "Instalar Tensor Flow (pip debe ser versión > 19)" in the top window.
- A yellow box highlights the command "import tensorflow as tf" in the bottom window.
- A yellow box highlights the text "Dentro del interprete revisar que se importe la biblioteca de tensorflow" and "Si no marca error, se instaló correctamente" in the bottom window.

# Entrar a jupyter

# Entrar a jupyter

## Paso 1

Entrar a jupyter

jupyter notebook

```
C:\ Command Prompt - conda install -c conda-forge opencv - conda install jupyter  
(py37) C:\Users\jessi>jupyter notebook
```

## Paso 2

En el navegador, se abren los directorios en la ruta de donde se mandó llamar jupyter

The screenshot shows a file browser interface for a Jupyter Notebook. The title bar says "localhost:8888/tree". The main area displays a list of files and directories under the root path "/". The files listed include: 3D Objects, Anaconda3, ansel, Contacts, Creative Cloud Files, Desktop, Documents, Downloads, Favorites, go, HP, Links, Music, OneDrive, Pictures, Roaming, Saved Games, Searches, source, Videos, and Untitled.ipynb. Each item has a checkbox to its left and columns for Name, Last Modified, and File size.

Name	Last Modified	File size
3D Objects	12 days ago	
Anaconda3	12 minutes ago	
ansel	4 months ago	
Contacts	12 days ago	
Creative Cloud Files	17 hours ago	
Desktop	6 days ago	
Documents	6 days ago	
Downloads	4 days ago	
Favorites	12 days ago	
go	3 months ago	
HP	6 months ago	
Links	12 days ago	
Music	12 days ago	
OneDrive	4 minutes ago	
Pictures	12 days ago	
Roaming	a year ago	
Saved Games	12 days ago	
Searches	12 days ago	
source	5 months ago	
Videos	12 days ago	
Untitled.ipynb	7 months ago	1.42 kB

## Paso 3

Dirigirse a la carpeta Actividad 0 en el archivo Revisar bibliotecas.ipynb

NOTA: Se te enviará por correo, deberás copiarla en algún lugar de tu computadora y anota la ruta.

The screenshot shows a file browser interface for a Jupyter Notebook, similar to the previous one but with a different path. The title bar says "localhost:8888/tree". The main area displays a list of files and directories under the path "/OneDrive/TRABAJO/MATERIAS/CURSO ML - PYTHON/Curso AplicacionesML/Actividad 0". The files listed are .. and Revisar bibliotecas.ipynb. The "Revisar bibliotecas.ipynb" file is highlighted with a yellow background. The interface includes tabs for Files, Running, and Clusters, and buttons for Upload, New, and Logout.

Name	Last Modified	File size
..	seconds ago	
Revisar bibliotecas.ipynb	seconds ago	1.48 kB

# Posibles errores en jupyter

Puede ocurrir un error en el kernel de Jupyter.

## Paso 1

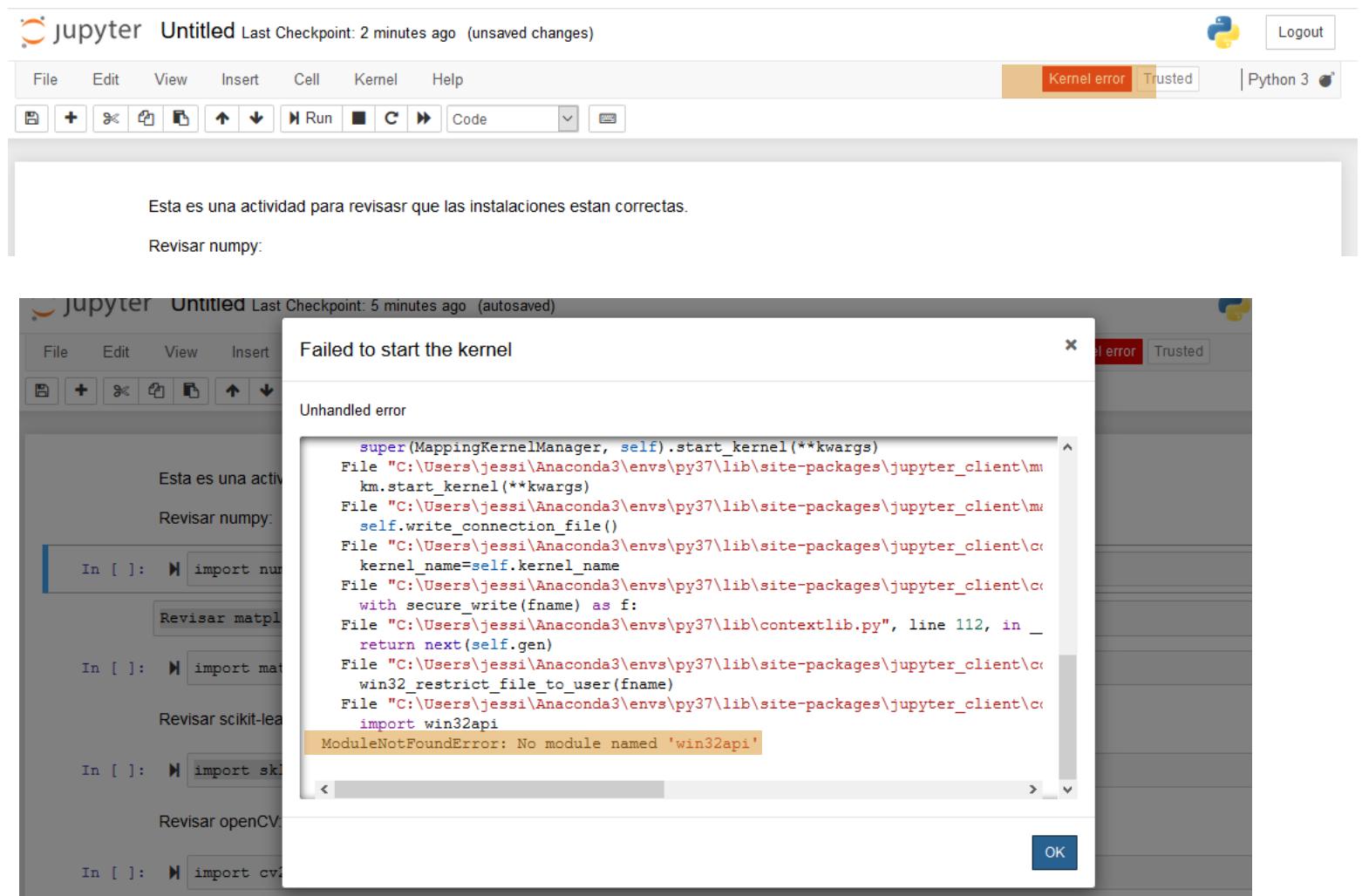
Revisa el error dando click al botón kernel error.

En este caso el error es que no encuentra 'win32api'

## Paso 2

Regresa a la ventana de comandos en el ambiente py37 e instálala.

```
conda install pywin32
```



```
(py37) C:\Users\jessi>conda install pywin32
```

# Revisar notebook en jupyter

## Paso 1

Dirigirse a la carpeta  
Actividad 0 en el archivo  
Revisar bibliotecas.ipynb

NOTA: Se te enviará por correo, deberás copiarla en algún lugar de tu computadora y anota la ruta.

## Paso 2

Posíóngate sobre cada Celda que es una entrada `In [ ]:` y presiona el botón Run.  
Si no marca error, entonces todo fue instalado correctamente.

The screenshot shows a Jupyter Notebook interface with the title "Revisar bibliotecas". The notebook has a "Run" button highlighted in orange at the top toolbar. Below the toolbar, there are several code cells labeled "In [ ]:" and their corresponding output cells labeled "Celda In [ ]:". The first cell contains the code "import numpy as np", the second "import matplotlib", and the third "import sklearn". The fourth cell starts with "import cv2" and the fifth with "import tensorflow". The "sklearn" cell is currently active, indicated by a green border around its input field. The "cv2" cell has a red border around its input field, suggesting it has run successfully. The "tensorflow" cell has a grey border around its input field, suggesting it has not run yet or is still running.

# Instrucciones de Instalación en Ubuntu 18.04 para el curso de **Aplicaciones del aprendizaje automático con Python**

Dra. Jessica Beltrán Márquez

[www.jessicabeltran.net](http://www.jessicabeltran.net)

Se sugiere seguir las instrucciones en:

<https://www.digitalocean.com/community/tutorials/how-to-install-anaconda-on-ubuntu-18-04-quickstart>

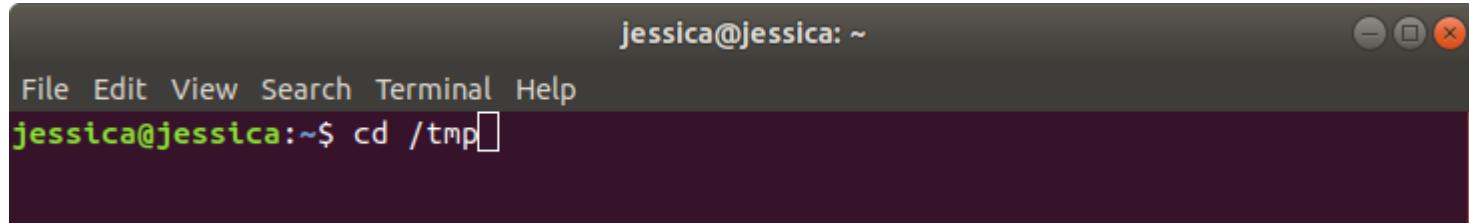
Para detalles sobre las bibliotecas y el uso de jupyter se sugiere revisar la sección de Windows.

# Instalar Anaconda

## Paso 1

Entrar a carpeta tmp

```
Cd /tmp
```



jessica@jessica: ~

File Edit View Search Terminal Help

```
jessica@jessica:~$ cd /tmp
```

## Paso 2

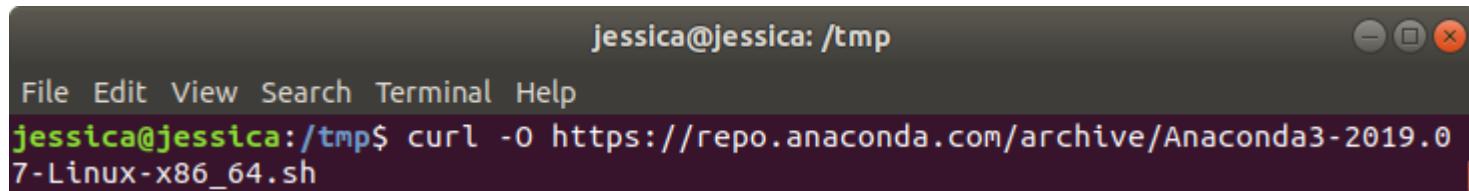
Descargar Script

Anaconda.

Esta versión es

2019.07. Revisar cual

está disponible.



jessica@jessica: ~

File Edit View Search Terminal Help

```
jessica@jessica:~$ curl -o https://repo.anaconda.com/archive/Anaconda3-2019.07-Linux-x86_64.sh
```

## Paso por si hay

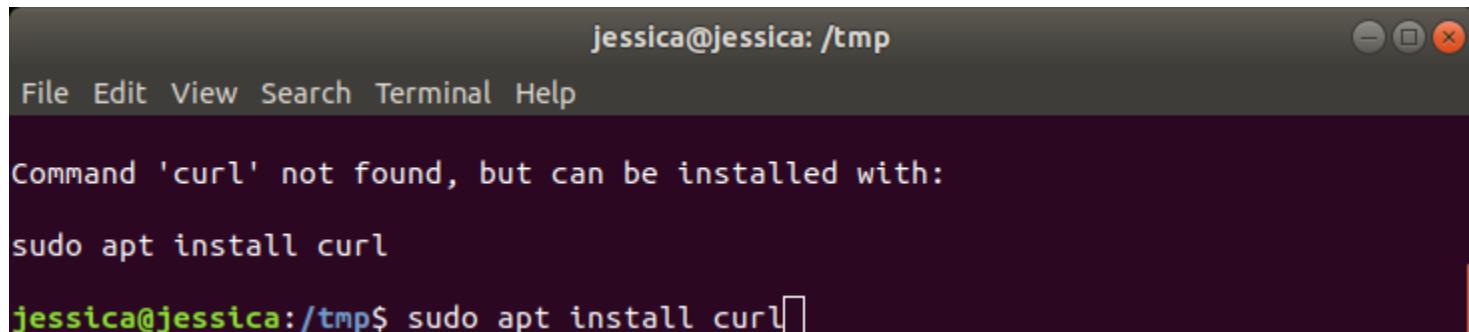
error de curl

Instalar curl.

Omitir si no hay

error.

```
Sudo apt install curl
```



jessica@jessica: ~

File Edit View Search Terminal Help

Command 'curl' not found, but can be installed with:

```
sudo apt install curl
```

```
jessica@jessica:~$ sudo apt install curl
```

### Paso 3

Verificar la integridad del archivo.

```
jessica@jessica: /tmp
File Edit View Search Terminal Help
jessica@jessica:/tmp$ sha256sum Anaconda3-2019.07-Linux-x86_64.sh
69581cf739365ec7fb95608eef694ba959d7d33b36eb961953f2b82cb25bdf5a  Anaconda3-2019.07-Linux-x86_64.sh
jessica@jessica:/tmp$
```

### Paso 4

Correr el script Anaconda

```
jessica@jessica: /tmp
File Edit View Search Terminal Help
jessica@jessica:/tmp$ bash Anaconda3-2019.07-Linux-x86_64.sh

Welcome to Anaconda3 2019.07

In order to continue the installation process, please review the license
agreement.
Please, press ENTER to continue
>>>
```

### Paso 5

Dar *Enter* mientras se muestra la licencia

```
jessica@jessica: /tmp
File Edit View Search Terminal Help
=====
Copyright 2015, Anaconda, Inc.

All rights reserved under the 3-clause BSD License:

Redistribution and use in source and binary forms, with or without modification, are permitted provided
that the following conditions are met:

    * Redistributions of source code must retain the above copyright notice, this list of conditions and
      the following disclaimer.
    * Redistributions in binary form must reproduce the above copyright notice, this list of conditions
      and the following disclaimer in the documentation and/or other materials provided with the distribution.

--More--
```

## Paso 6

Presionar enter

```
jessica@jessica: /tmp
File Edit View Search Terminal Help
>>>
Please answer 'yes' or 'no':'
>>>
Please answer 'yes' or 'no':'
>>> yes

Anaconda3 will now be installed into this location:
/home/jessica/anaconda3

- Press ENTER to confirm the location
- Press CTRL-C to abort the installation
- Or specify a different location below

[/home/jessica/anaconda3] >>> 
```

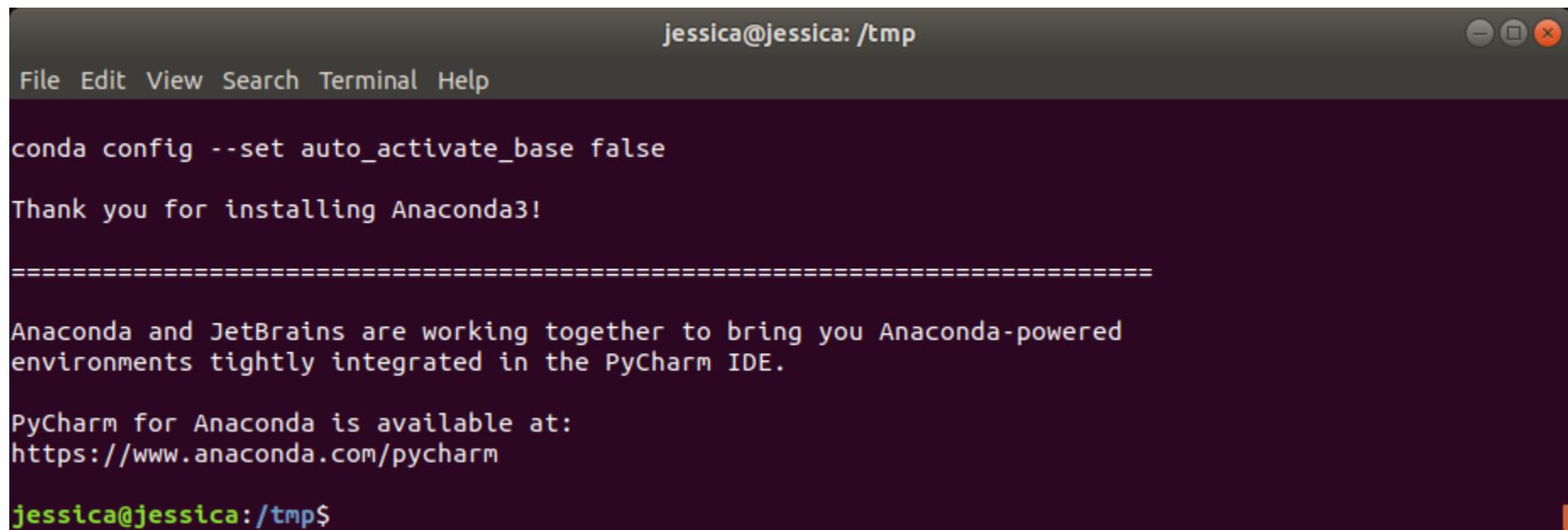
## Paso 7

Indicar yes

```
jessica@jessica: /tmp
File Edit View Search Terminal Help
zlib          pkgs/main/linux-64::zlib-1.2.11-h7b6447c_3
zstd          pkgs/main/linux-64::zstd-1.3.7-h0b5b093_0

Preparing transaction: done
Executing transaction: | WARNING conda.core.envs_manager:register_env(46): Unable to register environment. Path not writable or missing.
environment location: /home/jessica/anaconda3
registry file: /home/jessica/.conda/environments.txt
done
installation finished.
Do you wish the installer to initialize Anaconda3
by running conda init? [yes|no]
[no] >>> yes
```

# Instalación completada



jessica@jessica: /tmp

File Edit View Search Terminal Help

```
conda config --set auto_activate_base false

Thank you for installing Anaconda3!

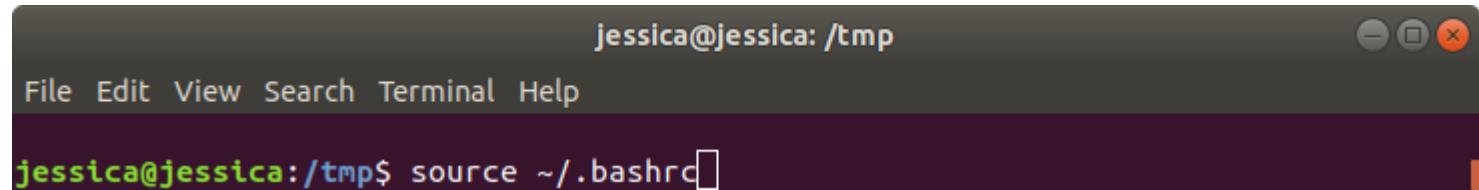
=====
Anaconda and JetBrains are working together to bring you Anaconda-powered
environments tightly integrated in the PyCharm IDE.

PyCharm for Anaconda is available at:
https://www.anaconda.com/pycharm

jessica@jessica:/tmp$
```

## Paso 8

### Activar la instalación



A screenshot of a terminal window titled "jessica@jessica: /tmp". The window has a dark grey header bar with white text. Below it is a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The main area of the terminal is dark purple and contains the command "jessica@jessica:/tmp\$ source ~/.bashrc" in white text. The cursor is positioned at the end of the command line.

## Crear ambiente

### Paso 1

Crear ambiente con  
Python 3.7  
jupyter notebook

```
jessica@jessica: /tmp
File Edit View Search Terminal Help
(base) jessica@jessica:/tmp$ conda create --name py37 python=3.7
```

### Paso 2

Entrar a ambiente

Conda activate py37

```
jessica@jessica: /tmp
File Edit View Search Terminal Help
(base) jessica@jessica:/tmp$ conda activate py37
```

### Paso 3

Revisar paquetes  
que requieren  
update

```
jessica@jessica: ~
File Edit View Search Terminal Help
(py37) jessica@jessica:~$ sudo apt update
```

```
jessica@jessica: ~
File Edit View Search Terminal Help
(py37) jessica@jessica:~$ python --version
Python 3.7.4
(py37) jessica@jessica:~$ pip --version
pip 19.2.3 from /home/jessica/anaconda3/envs/py37/lib/python3.7/site-packages/pip (python 3.7)
(py37) jessica@jessica:~$
```

### Paso 4

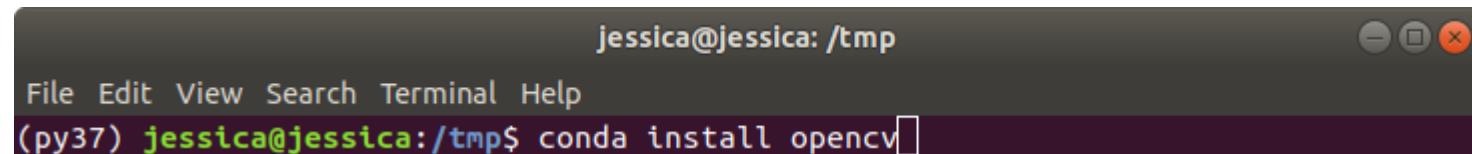
Revisar la versión de  
Python y pip

## Instalar openCV

### Paso 1

Instalar openCV

```
conda install opencv
```



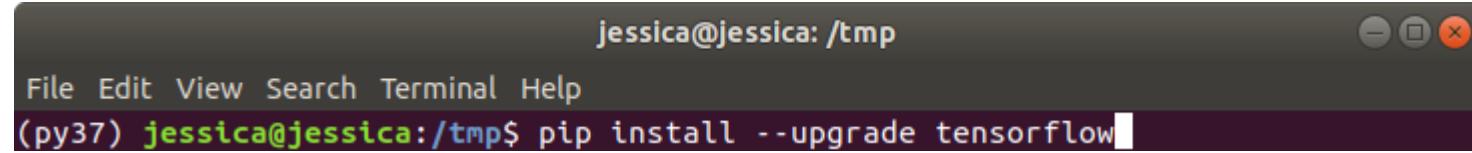
jessica@jessica: /tmp  
File Edit View Search Terminal Help  
(py37) jessica@jessica:/tmp\$ conda install opencv

## Instalar tensorflow

### Paso 1

Instalar tensorflow

```
pip install --upgrade tensorflow
```



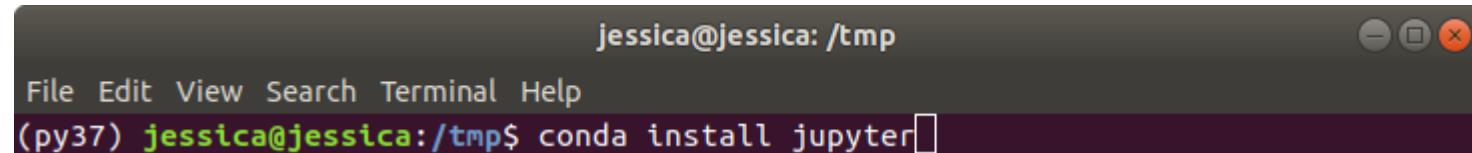
jessica@jessica: /tmp  
File Edit View Search Terminal Help  
(py37) jessica@jessica:/tmp\$ pip install --upgrade tensorflow

## Instalar jupyter

### Paso 1

Instalar jupyter

```
conda install jupyter
```



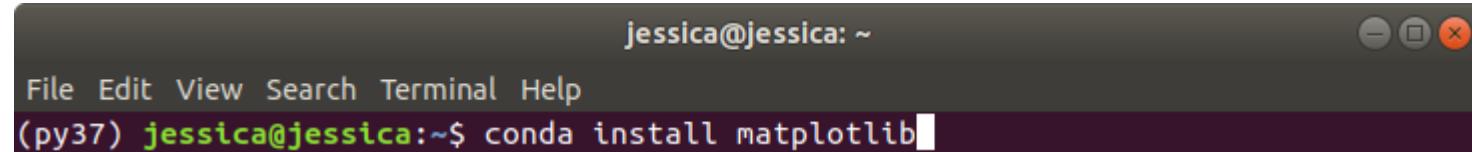
jessica@jessica: /tmp  
File Edit View Search Terminal Help  
(py37) jessica@jessica:/tmp\$ conda install jupyter

## Instalar matplotlib

### Paso 1

Instalar matplotlib

```
conda install matplotlib
```



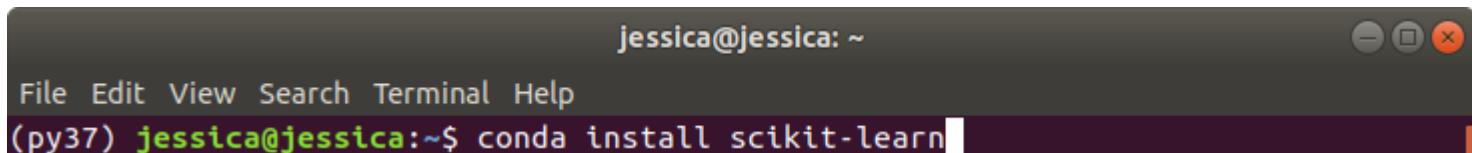
jessica@jessica: ~  
File Edit View Search Terminal Help  
(py37) jessica@jessica:~\$ conda install matplotlib

## Instalar scikit

### Paso 1

Instalar scikit

```
conda install scikit-learn
```



jessica@jessica: ~  
File Edit View Search Terminal Help  
(py37) jessica@jessica:~\$ conda install scikit-learn

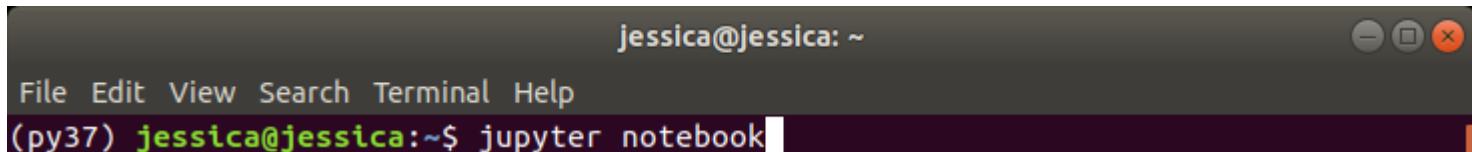
## Lanzar jupyter

### Paso 1

Lanzar jupyter

09-Oct-19

```
jupyter notebook
```



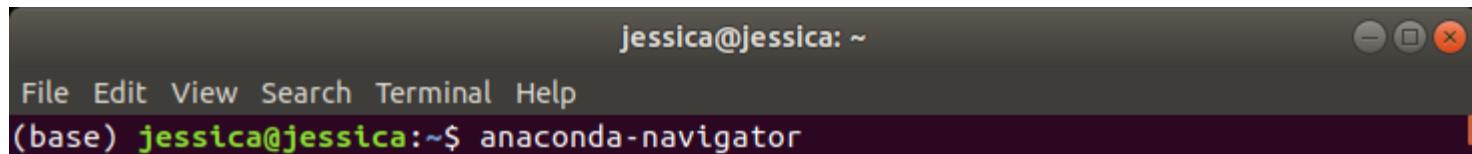
jessica@jessica: ~  
File Edit View Search Terminal Help  
(py37) jessica@jessica:~\$ jupyter notebook

## Abrir anaconda

### Paso 1

Abrir Anaconda

Anaconda-navigator



# Instalar Spyder y Orange

# ANACONDA® NAVIGATOR

[Sign in to Anaconda Cloud](#)[Home](#)[Environments](#)[Learning](#)[Community](#)[Documentation](#)[Developer Blog](#)

Applications on

py37

Channels

## 1. Cambiar al ambiente py37

[Refresh](#)

1.1.4

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

[Install](#)

3.23.0

Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

[Install](#)

1.1.456

A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

[Install](#)

Spyder

3.3.6

Scientific PYTHON Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

[Install](#)

## 2. Presionar el botón de instalar. Tarda de 2 a 3 minutos.

# ANACONDA® NAVIGATOR

[Sign in to Anaconda Cloud](#)[Home](#)[Environments](#)[Learning](#)[Community](#)[Documentation](#)[Developer Blog](#)

Applications on

py37

Channels

## 1. Cambiar al ambiente py37

[Refresh](#)

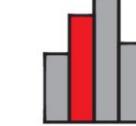
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

[Launch](#)

Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

[Launch](#)

Streamlined code editor with support for development operations like debugging, task running and version control.

[Launch](#)

0.15.2

Multidimensional data visualization across files. Explore relationships within and among related datasets.

[Install](#)

1.1.4

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

[Install](#)

3.23.0

Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

[Install](#)

## 2. Presionar el botón de instalar. Tarda de 2 a 3 minutos.

# Actividad 1

<https://github.com/jebema/Taller-aprendizaje-automatico-con-Python>

# Iniciando

The screenshot shows a Jupyter Notebook interface. At the top, there's a header bar with the Jupyter logo, the title "Iniciando", the last checkpoint date "09/24/2019 (unsaved changes)", and user icons for Python 3 and Logout. Below the header is a toolbar with various icons for file operations like Open, Save, and Run, along with Kernel and Help buttons. The main content area has a title "Iniciando" and a subtitle "Curso: Aplicaciones del aprendizaje automático con Python". It also includes the text "Actividad 1: Iniciando con Python.", a section for "Objetivos:" with a numbered list of five items, and author information "Dra. Jessica Beltrán Márquez" with a link "www.jessicabeltran.net".

A continuación asignaremos valores a dos variables y después las sumaremos. Python es sensible a mayúsculas y minúsculas, por lo que la variable "a" es distinta que "A".

```
In [ ]: ┌ a = 90
          b = 7
          A = 25

          c = a + b
          d = A + b

          print("a + b = ", c)
          print("A + b = ", d)
```

**Pruébalo:** Cambia los valores de las siguientes variables y prueba los distintos resultados.

```
In [ ]: ┌ f = 5## <-- Escribe un número para asignarle un valor a f
```

# Iniciando

**Pruébalo:** Cambia los valores de las siguientes variables y prueba los distintos resultados.

```
In [2]: f = 5 ## <-- Escribe un número para asignarle un valor a f
g = 7 ## <-- Escribe un número para asignarle un valor a g
x = f+g*g
print("f + g * g = ",x)
```

f + g \* g = 54

**Pruébalo:** Indica para que se utilizan las siguientes funciones:

- floor(x)
- fmod(x)
- pow(x)
- atan(x)

```
In [5]: x = 7.2
#Usa las funciones de la biblioteca math floor, fmod, pow y atan y encuentra para que se utilizan.
floor_val = math.floor(x)
fmod_val = math.fmod(x,3)
pow_val = math.pow(x,4)
atan_val = math.atan(x)

print("floor(x)= ", floor_val)
print("fmod(x)= ",fmod_val)
print("pow(x)= ",pow_val)
print("atan(x)= ",atan_val)
```

floor(x)= 7  
fmod(x)= 1.2000000000000002  
pow(x)= 2687.3856  
atan(x)= 1.4327903031373772

# Iniciando

Aunque las listas son útiles en varias aplicaciones. No proveen la ventaja de poder realizar operaciones matemáticas. Por ejemplo, si se quisiera obtener la raíz cuadrada de todos los elementos de la lista Python regresa un **error**. Por este motivo se utiliza la biblioteca *numpy*.

```
In [13]: ┌─▶ a = [1,2,3]
      c = math.sqrt(a)
      print(a)
```

```
-----  
TypeError                                     Traceback (most recent call last)  
<ipython-input-13-a5c13a1611ec> in <module>  
      1 a = [1,2,3]  
----> 2 c = math.sqrt(a)  
      3 print(a)
```

```
TypeError: must be real number, not list
```

# Iniciando

**Pruébalo:** Crea una matriz aleatoria con shape (4,5) con Python e imprime la raíz cuadrada de todos los elementos.

Puedes investigar la función random en <https://het.as.utexas.edu/HET/Software/Numpy/reference/generated/numpy.random.rand.html>

Quizas te convenga usar np.sqrt en vez de math.sqrt

```
In [20]: ➜ matriz = np.random.rand(4,5)
          res = np.sqrt(matriz)
          print(res)
```

```
[[0.57340336 0.47695059 0.98975579 0.71330205 0.9106562 ]
 [0.98740253 0.28247437 0.69305996 0.74070611 0.80887246]
 [0.8935509  0.79688789 0.97868132 0.22428168 0.81220507]
 [0.88186444 0.74965589 0.57743539 0.69965089 0.84133776]]
```

# Actividad 2

# OpenCV

The screenshot shows a Jupyter Notebook interface. At the top, there's a toolbar with various icons for file operations like 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. To the right of the toolbar are buttons for 'Trusted' and 'Python 3'. On the far right, there are 'Logout' and a Python logo. Below the toolbar, there's a menu bar with icons for file operations like 'New', 'Open', 'Save', etc., followed by 'Run', 'Cell', 'C', 'Markdown', and a dropdown menu.

**OpenCV**

Curso: Aplicaciones del aprendizaje automático con Python  
Actividad 2: Iniciando con OpenCV

En esta actividad nos familiarizaremos con la biblioteca openCV.

Objetivos:

1. Leer una imagen
2. Desplegar una imagen con matplotlib
3. Aplicar procesamiento a una imagen.
4. Guardar una imagen nueva.

Dra. Jessica Beltrán Márquez  
[www.jessicabeltran.net](http://www.jessicabeltran.net)

Lo primero que tenemos que hacer es importar las bibliotecas que vamos a necesitar.

- cv2: Es la biblioteca openCV
- numpy: Es una biblioteca que se utiliza para realizar operaciones numéricas.
- matplotlib: Es una librería que se utiliza para graficar en jupyter.

In [2]:

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
cv2.__version__
```

Out[2]: '3.4.2'

# OpenCV

## Leer una imagen

A continuación leeremos una imagen usando la función *imread* dentro de openCV.

La imagen *dog1* está almacenada en la carpeta *images*.

También vamos a mostrar cuál es la forma de la imagen (*shape*) usando la función *shape*.

```
In [3]: img = cv2.imread('./images/dog1.jpg')
imgShape = img.shape
print(imgShape)
```

The screenshot shows a Jupyter Notebook interface. At the top, there is a code cell with the following Python code:

```
In [3]: img = cv2.imread('./images/dog1.jpg')
imgShape = img.shape
print(imgShape)
```

The output of this cell is not visible in the screenshot. Below the code cell is a file browser interface. The title bar says "jupyter". The top navigation bar has tabs for "Files", "Running", and "Clusters". On the right side of the top bar are "Quit" and "Logout" buttons. Below the top bar is a toolbar with "Upload", "New", and a refresh icon. The main area is a file list:

	Name	Last Modified	Type
<input type="checkbox"/>	0	seconds ago	
<input type="checkbox"/>	..		
<input type="checkbox"/>	images	10 days ago	
<input type="checkbox"/>	OpenCV.ipynb	Running 10 days ago	351 kB

At the bottom of the interface, there is a detailed view of the "images" folder:

Name	Status	Date modified	Type	Size
.ipynb_checkpoints		11-Jun-19 4:58 PM	File folder	
images		26-Sep-19 1:21 PM	File folder	
OpenCV.ipynb	⟳	06-Oct-19 3:05 PM	IPYNB File	344 KB

Next to the "images" folder, there are two thumbnail preview images labeled "dog1" and "dog2".

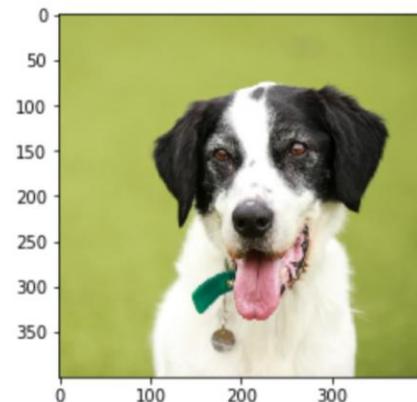
# OpenCV

## Desplegar una imagen

Ahora vamos a desplegar la imagen utilizando la función `pyplot --> plt` contenida en la biblioteca matplotlib.

Nota: Matplotlib utiliza un esquema de colores distinto a RGB (usa BGR), por tal motivo utilizamos la linea `img = img[:, :, ::-1]`. Puedes comentar esta linea usando "#" para ver lo que sucede.

```
In [4]: img = img[:, :, ::-1] #Esto porque matplotlib utiliza BGR en vez de RGB  
plt.imshow(img)  
plt.show()
```



RGB



# OpenCV

**Pruébalo:** Utiliza el filtro CornerHarris [https://docs.opencv.org/2.4/modules/imgproc/doc/feature\\_detection.html?highlight=cornerharris](https://docs.opencv.org/2.4/modules/imgproc/doc/feature_detection.html?highlight=cornerharris) en vez de Canny.

Se requiere convertir la imagen a grises. Puedes probar con `cv2.cvtColor` antes de usar CornerHarris.

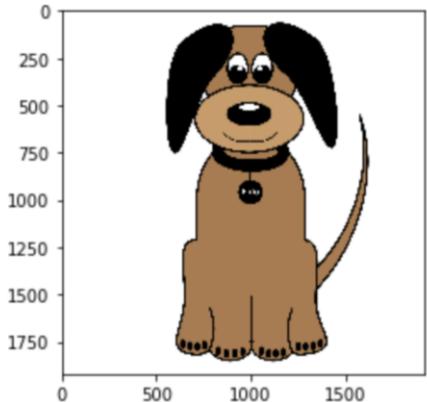
Si usas los parámetros 3,3,0.01 tendrás una imagen como ésta:



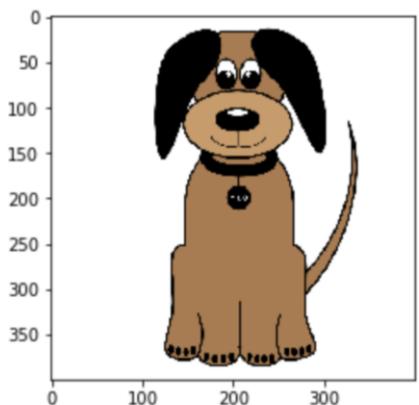
```
In [7]: ► gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
gray = np.float32(gray)
edgesImgHarris = cv2.cornerHarris(gray,3,3,0.01) #Se aplica el filtro CornerHarris
plt.subplot(121),plt.imshow(img,cmap = 'gray') #Se hace una subgráfica de la imagen original
plt.title('Imagen original'), plt.xticks([]), plt.yticks([]) #Se establece el título de la imagen y se especifica que
plt.subplot(122),plt.imshow(edgesImgHarris,cmap = 'gray') #Se hace una subgráfica de la imagen filtrada con los bordes
plt.title('Imagen filtrada con los bordes'), plt.xticks([]), plt.yticks([]) #Se establece el título de la imagen y se
plt.show()
```

# OpenCV

Forma de la imagen original = (1920, 1920, 3)



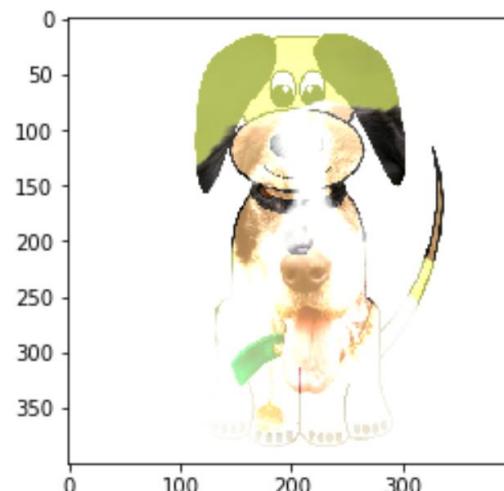
Forma de la imagen despues del resize = (400, 400, 3)



# OpenCV

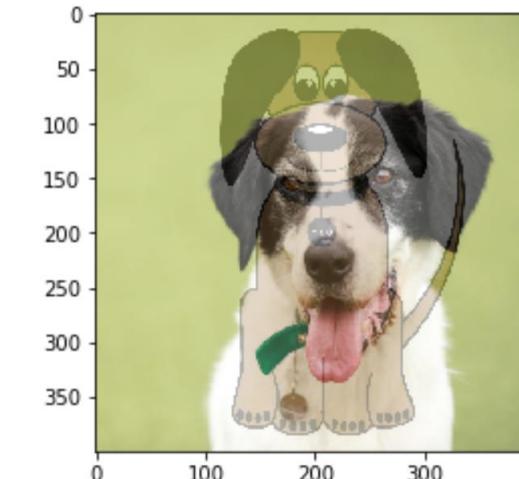
Lo siguiente que hacemos es sumar ambas imágenes y posteriormente desplegarlas.

```
In [11]: ► imagenMezclada = cv2.add(img, img2Resized)
#imagenMezclada = cv2.addWeighted(img, 0.7, img2Resized, 0.3, 0)
plt.imshow(imagenMezclada)
plt.show()
```



Lo siguiente que hacemos es sumar ambas imágenes y posteriormente desplegarlas.

```
In [12]: ► #imagenMezclada = cv2.add(img, img2Resized)
imagenMezclada = cv2.addWeighted(img, 0.7, img2Resized, 0.3, 0)
plt.imshow(imagenMezclada)
plt.show()
```

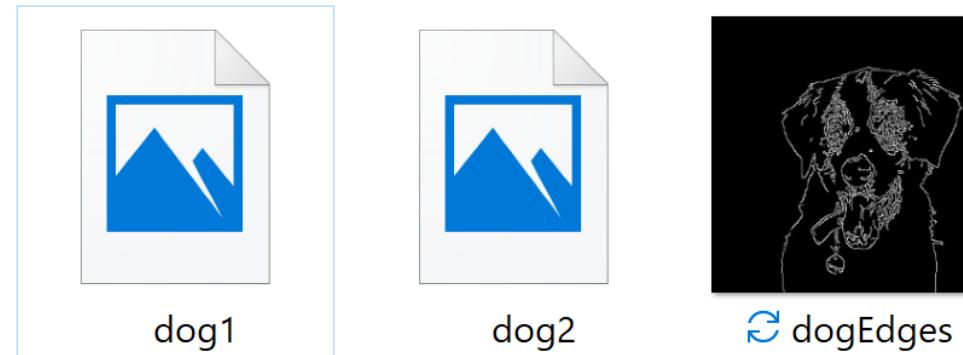


## Guardar una imagen

Lo siguiente que haremos será guardar una nueva imagen en la carpeta `images`. Guardaremos la imagen procesada `edgesImg`.

```
In [6]: ┌─ cv2.imwrite('./images/dogEdges.jpg',edgesImg) #Se especifica la ruta y nombre de la imagen.
```

```
Out[6]: True
```



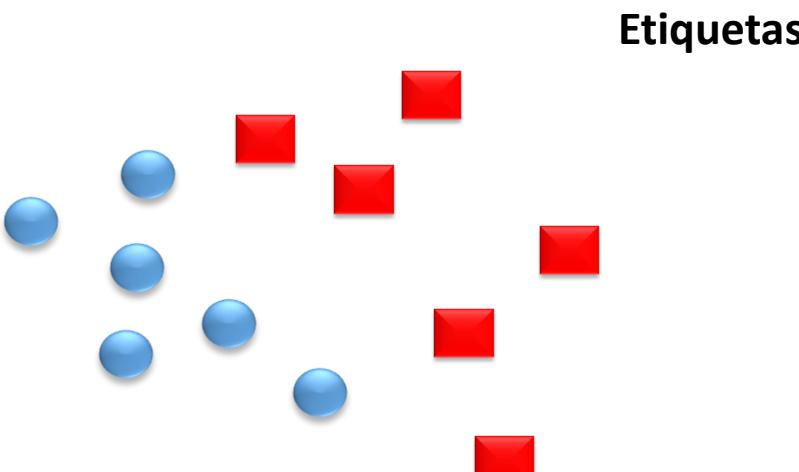
# -Más sobre aprendizaje automático-

# Conceptos importantes en el área de clasificación



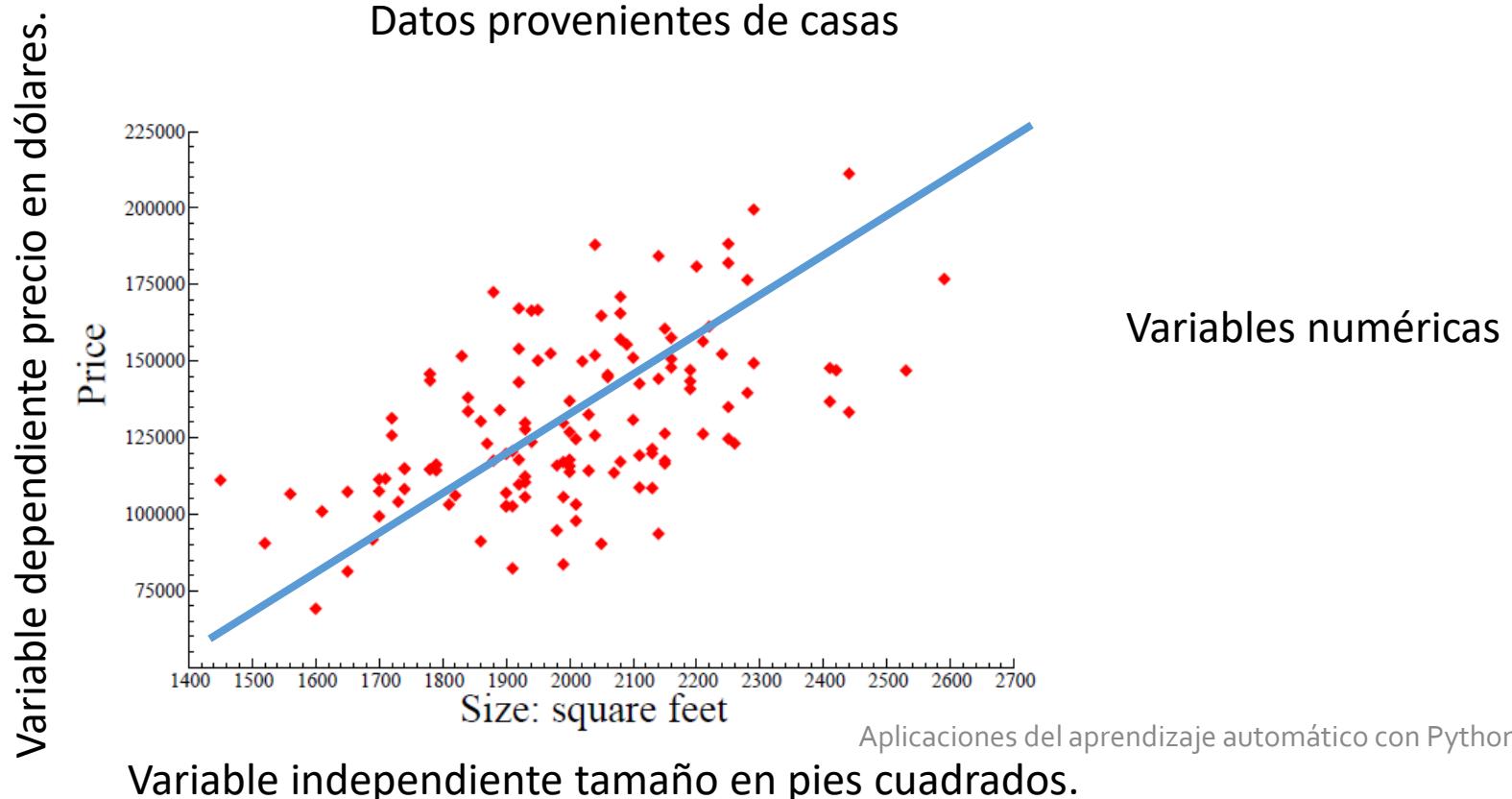
# Clasificación

Consiste en predecir un número discreto de valores que representan a **clases (categorías)**. Por ejemplo: Clasificar perros vs gatos, tipo de planta 1 vs tipo de planta 2 vs tipo de planta 3.



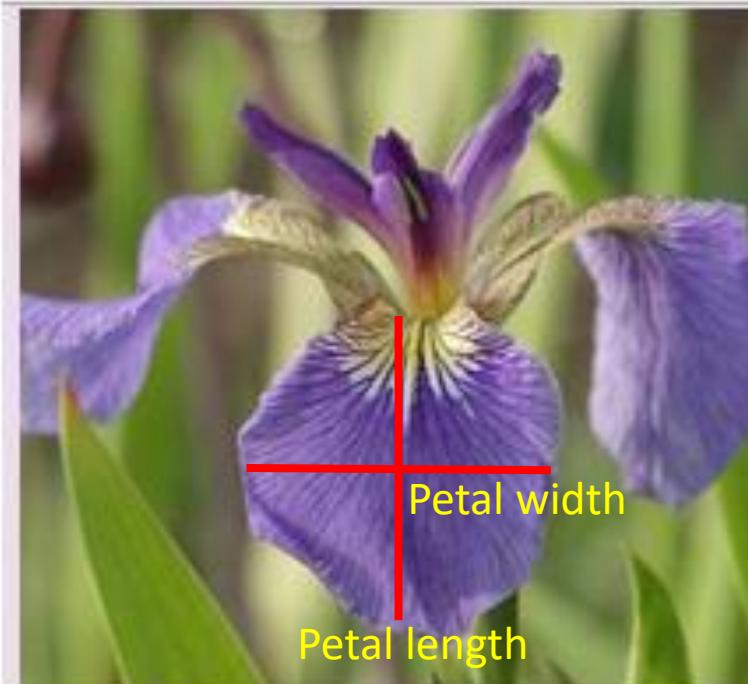
# Regresión

Los modelos de regression mapean el espacio de entrada a un dominio con valores reales . Consiste en encontrar la mejor relación que representa al conjunto de datos. Se usan las relaciones históricas entre las variables independientes y las variables dependientes para predecir valores futuros de la variable dependiente.



# Características (Atributos)

- Es una propiedad (característica) medible de un fenómeno que está siendo observado. Es importante obtener características informativas, discriminantes e independientes para diferenciar entre distintas clases.



Setosa Iris



Versicolor Iris



Virginica Iris



Setosa Iris

	iris	sepal length	sepal width	petal length	petal width
1	Iris-setosa	5.1	3.5	1.4	0.2
2	Iris-setosa	4.9	3.0	1.4	0.2
3	Iris-setosa	4.7	3.2	1.3	0.2
4	Iris-setosa	4.6	3.1	1.5	0.2
5	Iris-setosa	5.0	3.6	1.4	0.2



Versicolor Iris

	iris	sepal length	sepal width	petal length	petal width
51	Iris-versicolor	7.0	3.2	4.7	1.4
52	Iris-versicolor	6.4	3.2	4.5	1.5
53	Iris-versicolor	6.9	3.1	4.9	1.5
54	Iris-versicolor	5.5	2.3	4.0	1.3
55	Iris-versicolor	6.5	2.8	4.6	1.5



Virginica Iris

	iris	sepal length	sepal width	petal length	petal width
101	Iris-virginica	6.3	3.3	6.0	2.5
102	Iris-virginica	5.8	2.7	5.1	1.9
103	Iris-virginica	7.1	3.0	5.9	2.1
104	Iris-virginica	6.3	2.9	5.6	1.8
105	Iris-virginica	6.5	3.0	5.8	2.2

# ANACONDA® NAVIGATOR

[Sign in to Anaconda Cloud](#)[Home](#)[Environments](#)[Learning](#)[Community](#)[Documentation](#)[Developer Blog](#)

Applications on

py37

Channels

Refresh



**Jupyter**  
Notebook  
6.0.1

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

[Launch](#)



**Orange 3**  
3.23.0

Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

[Launch](#)



**Spyder**  
3.3.6

Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

[Launch](#)



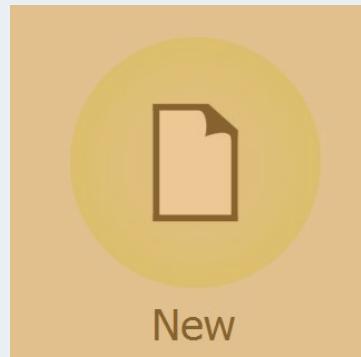
**VS Code**  
1.29.1



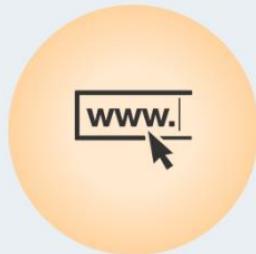
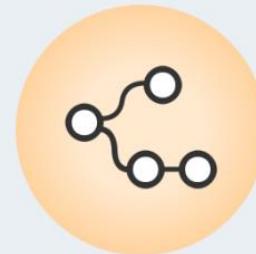
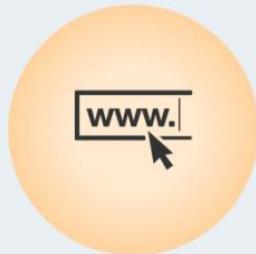
**Glueviz**  
0.15.2

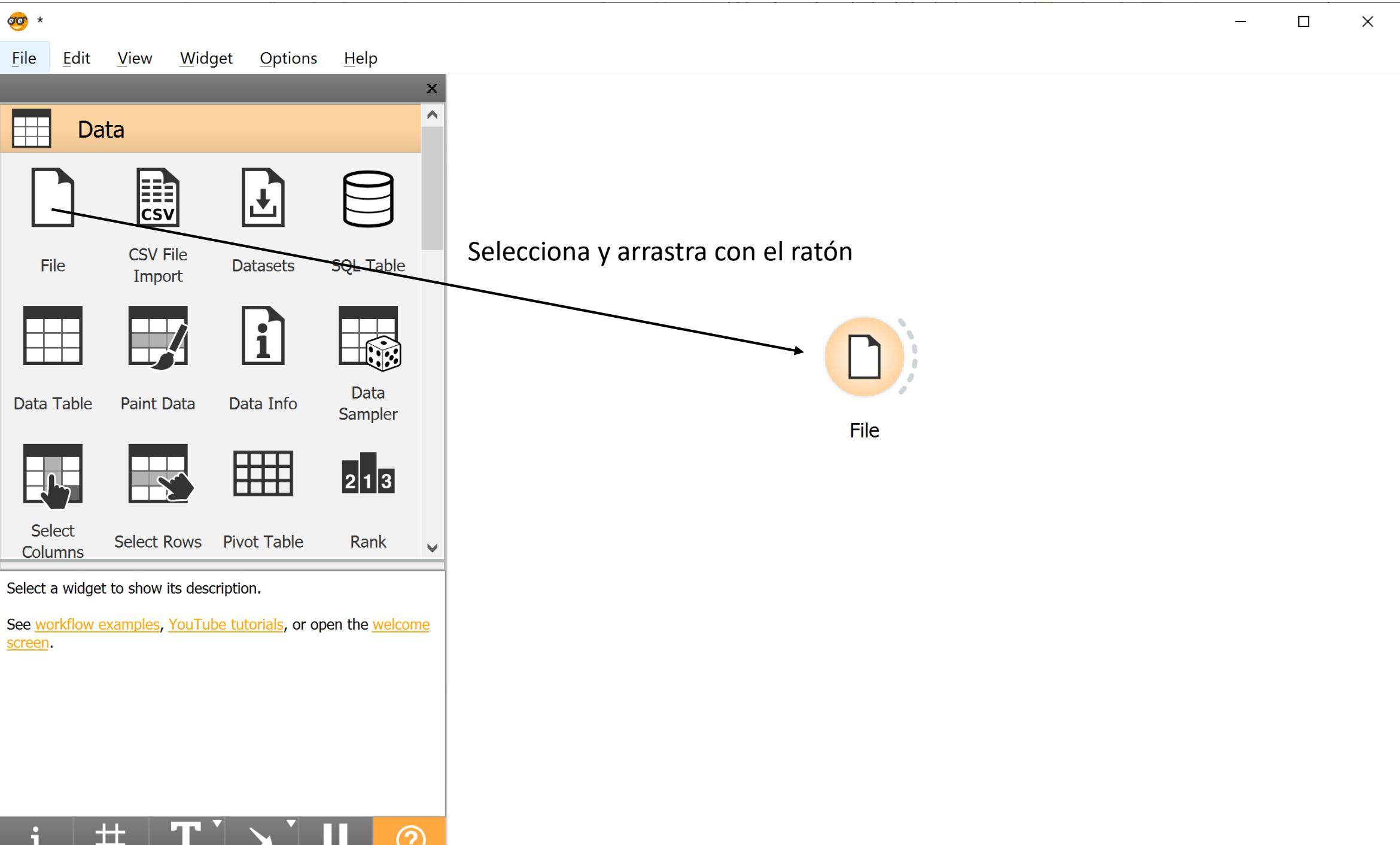


**JupyterLab**  
1.1.4



Video Tutorials







File Edit View Widget Options Help

### Data

File CSV File Import Datasets SQL Table

Data Table Paint Data Data Info Data Sampler

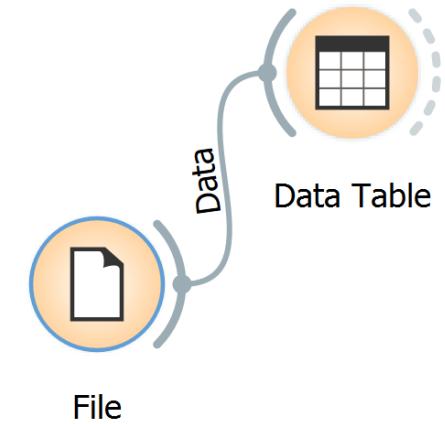
Select Columns Select Rows Pivot Table Rank

### File

Read data from an input file or network and send a data table to the output.

[more...](#)

i # T ↻ || ?



Data Table

Info  
150 instances (no missing values)  
4 features (no missing values)  
Discrete class with 3 values (no missing values)  
No meta attributes

Variables  
 Show variable labels (if present)  
 Visualize numeric values  
 Color by instance classes

Selection  
 Select full rows

Restore Original Order

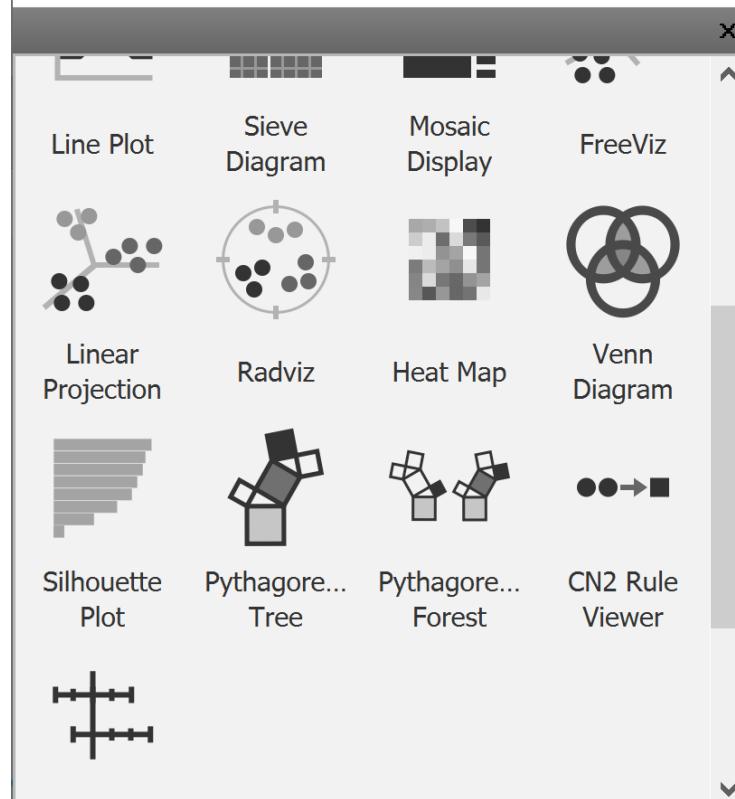
Send Automatically

	iris	sepal length	sepal width	petal length	petal width
1	Iris-setosa	5.1	3.5	1.4	0.2
2	Iris-setosa	4.9	3.0	1.4	0.2
3	Iris-setosa	4.7	3.2	1.3	0.2
4	Iris-setosa	4.6	3.1	1.5	0.2
5	Iris-setosa	5.0	3.6	1.4	0.2
6	Iris-setosa	5.4	3.9	1.7	0.4
7	Iris setosa	4.6	3.4	1.4	0.3
8	Iris-setosa	5.0	3.4	1.5	0.2
9	Iris-setosa	4.4	2.9	1.4	0.2
10	Iris-setosa	4.9	3.1	1.5	0.1
11	Iris-setosa	5.4	3.7	1.5	0.2
12	Iris-setosa	4.8	3.4	1.6	0.2
13	Iris-setosa	4.8	3.0	1.4	0.1
14	Iris setosa	4.3	3.0	1.1	0.1
15	Iris-setosa	5.8	4.0	1.2	0.2
16	Iris setosa	5.7	4.4	1.5	0.4
17	Iris-setosa	5.4	3.9	1.3	0.4
18	Iris setosa	5.1	3.5	1.4	0.3
19	Iris setosa	5.7	3.8	1.7	0.3
20	Iris-setosa	5.1	3.8	1.5	0.3
21	Iris setosa	5.4	3.4	1.7	0.2
22	Iris setosa	5.1	3.7	1.5	0.4
23	Iris-setosa	4.6	3.6	1.0	0.2

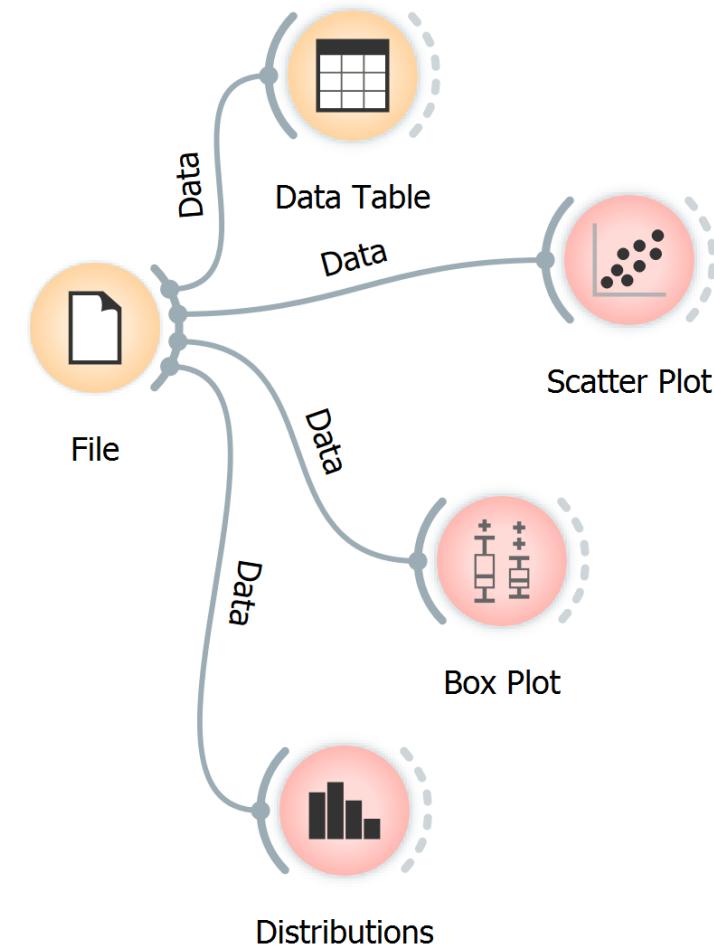


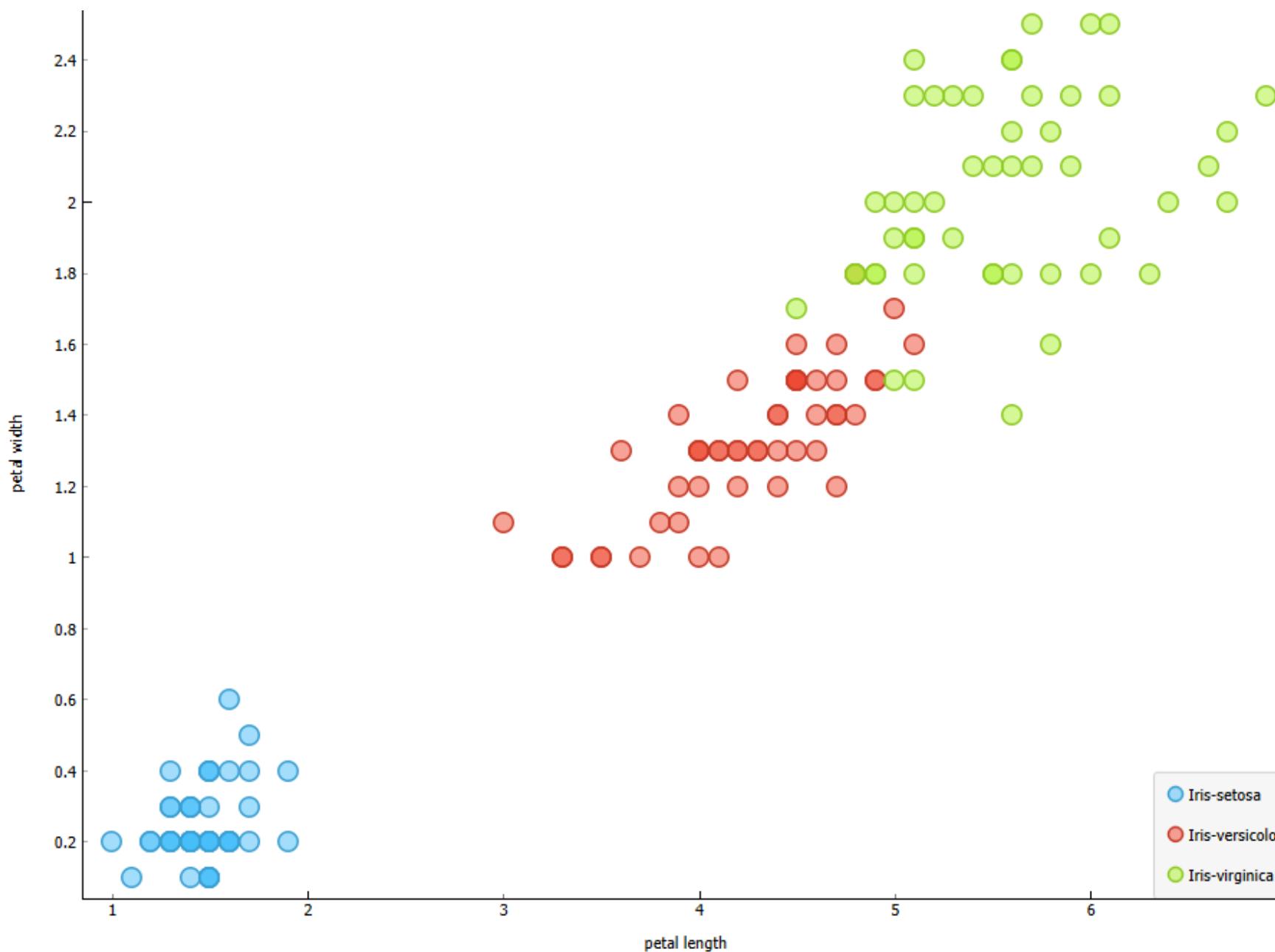
\*

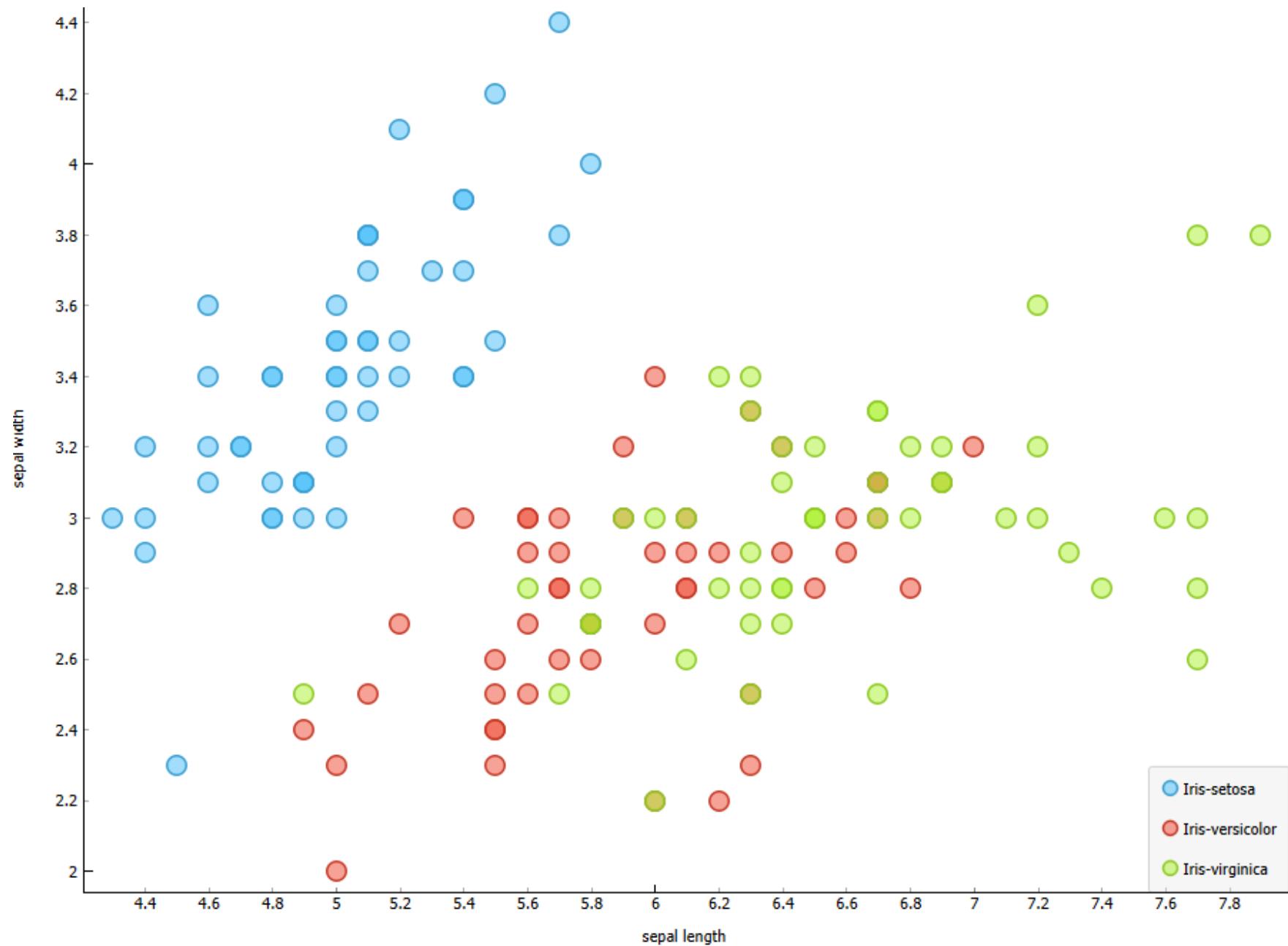
File Edit View Widget Options Help

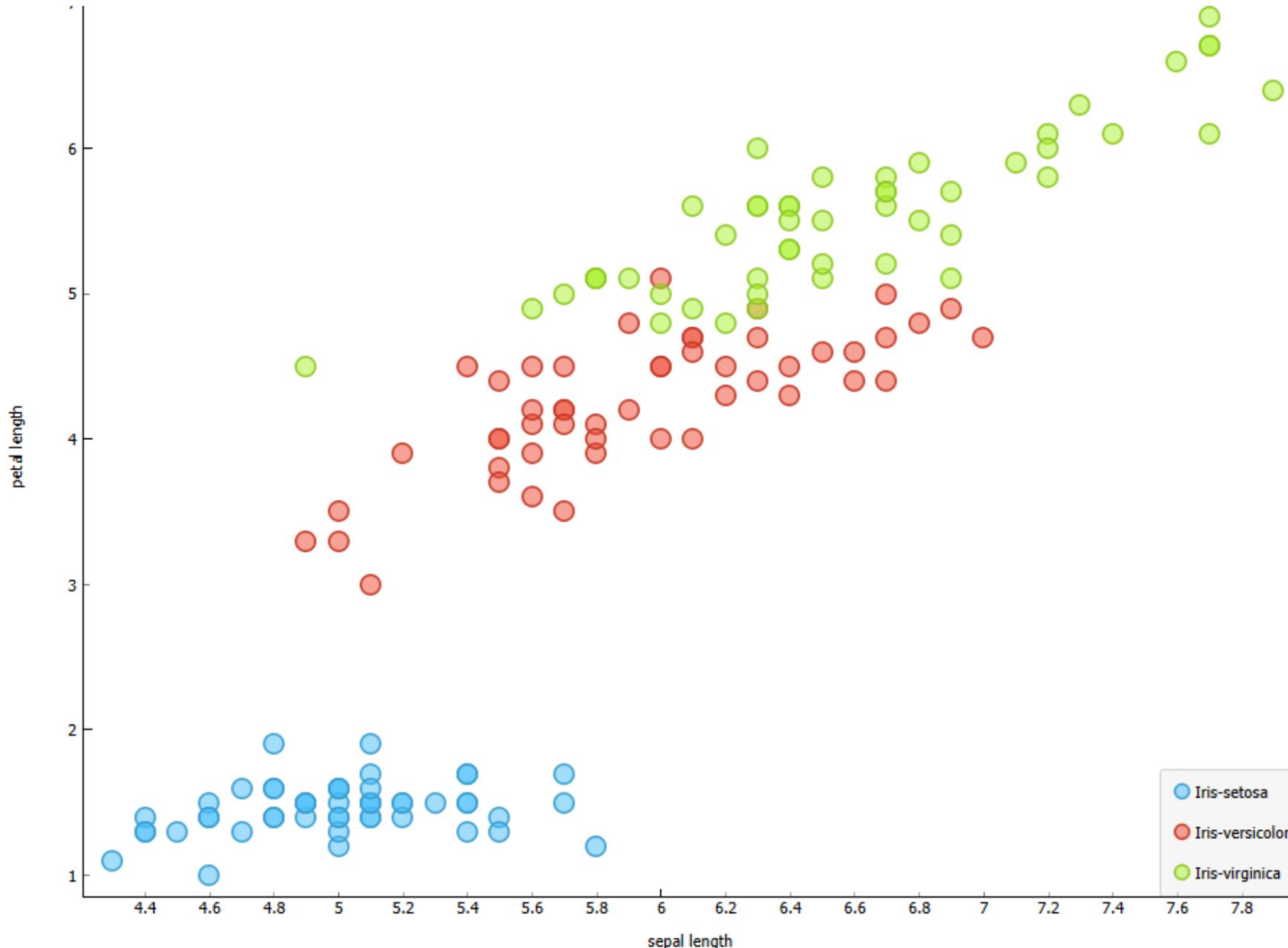


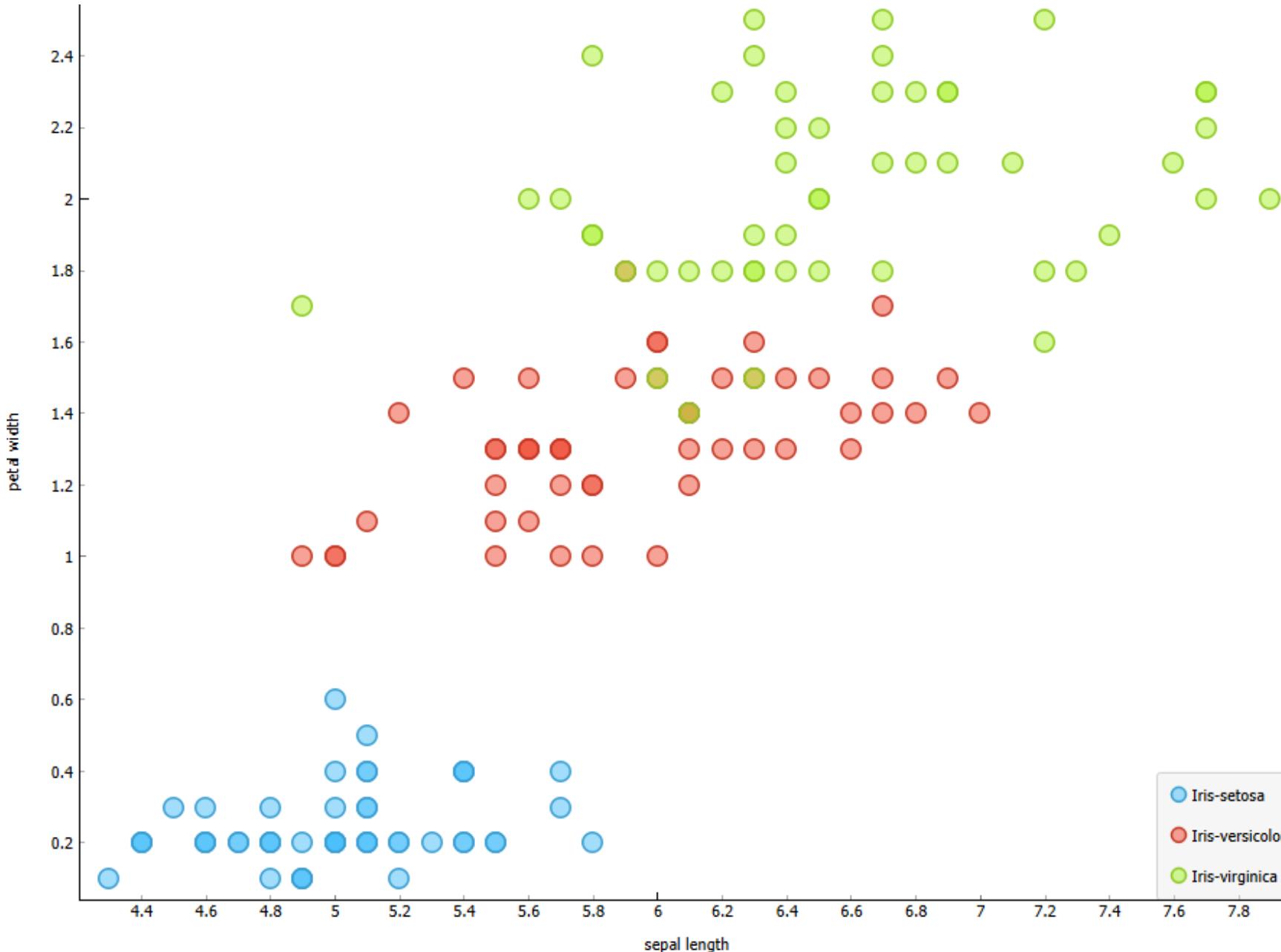
See [workflow examples](#), [YouTube tutorials](#), or open the [welcome screen](#).

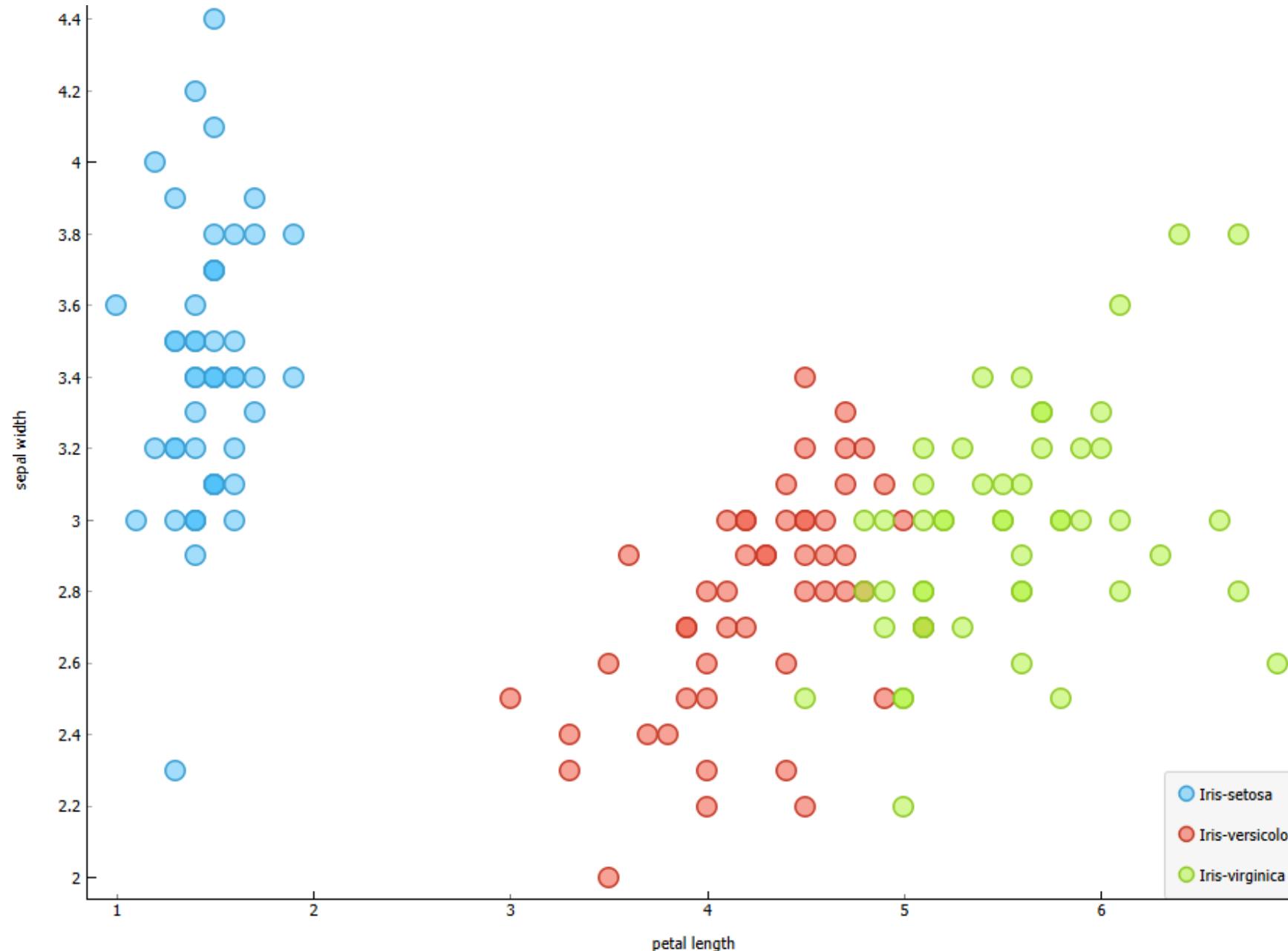


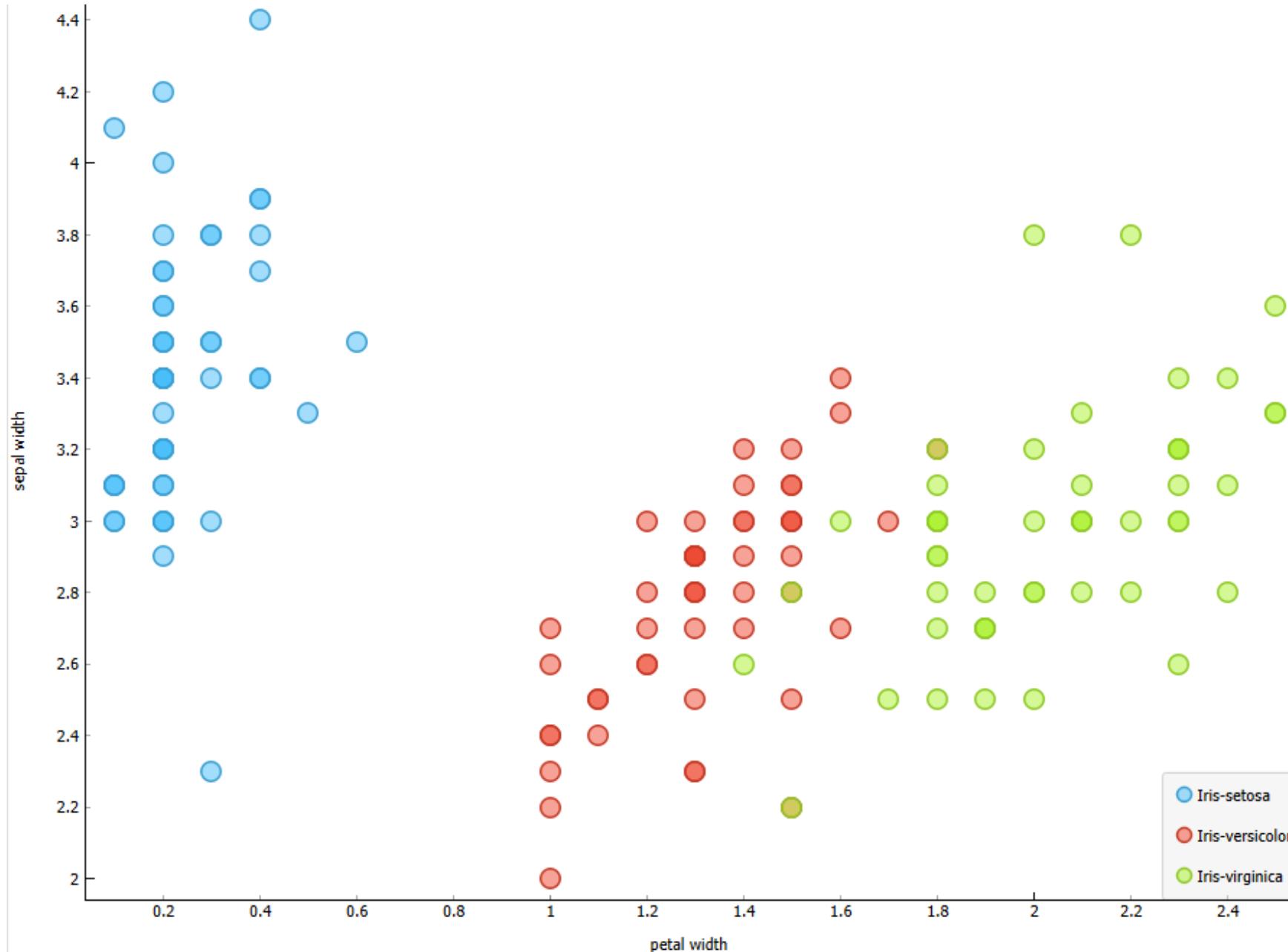


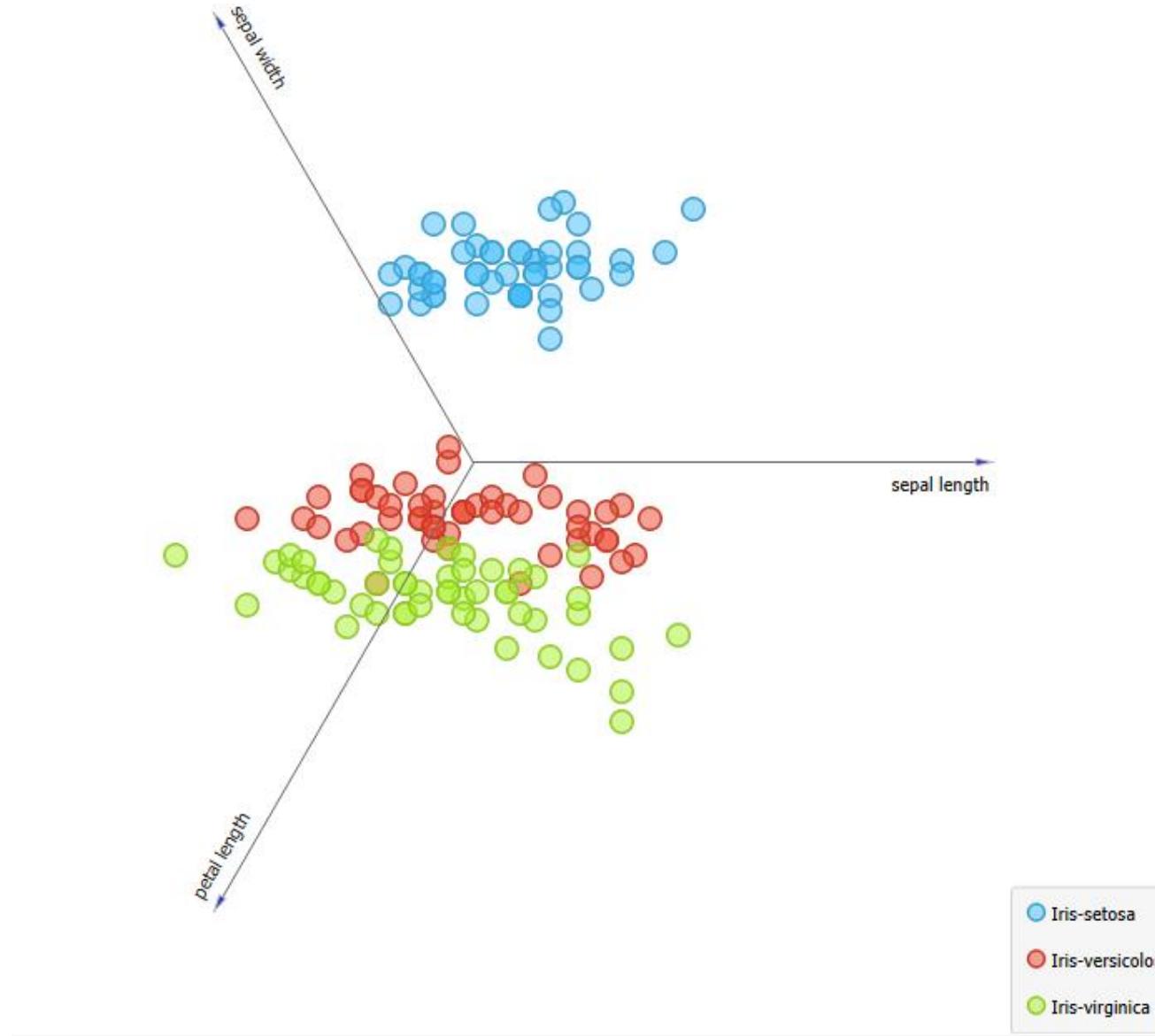


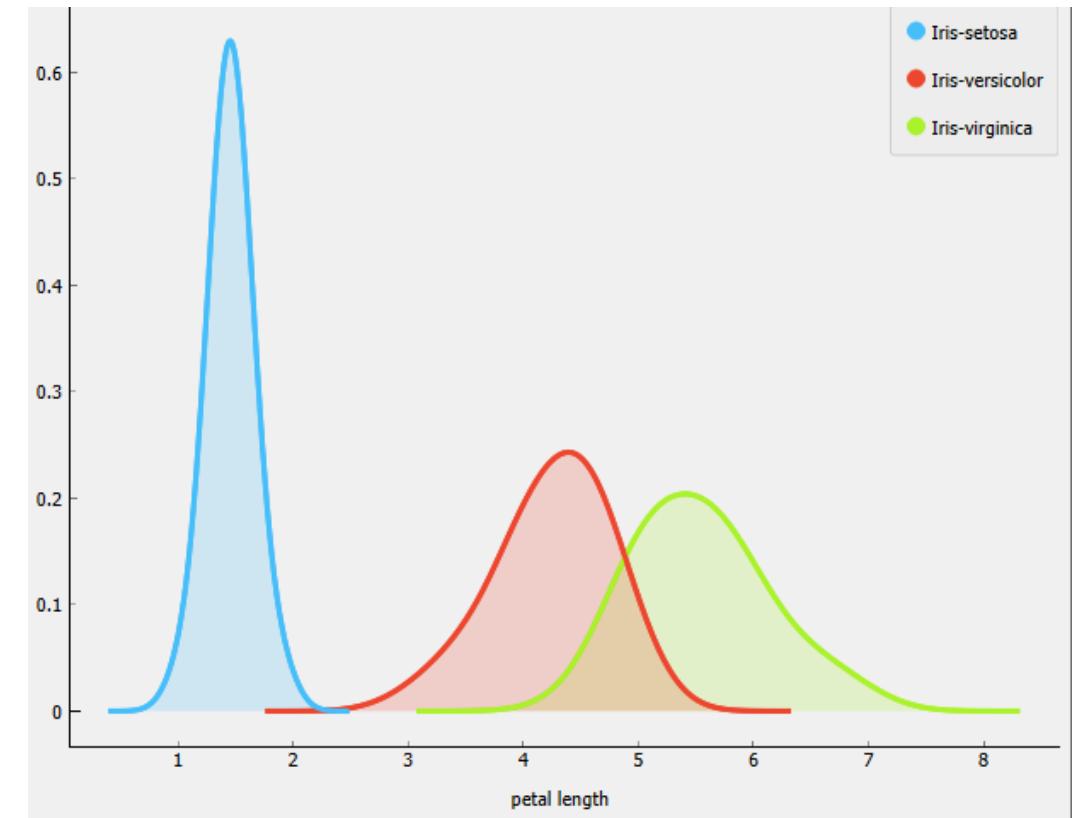
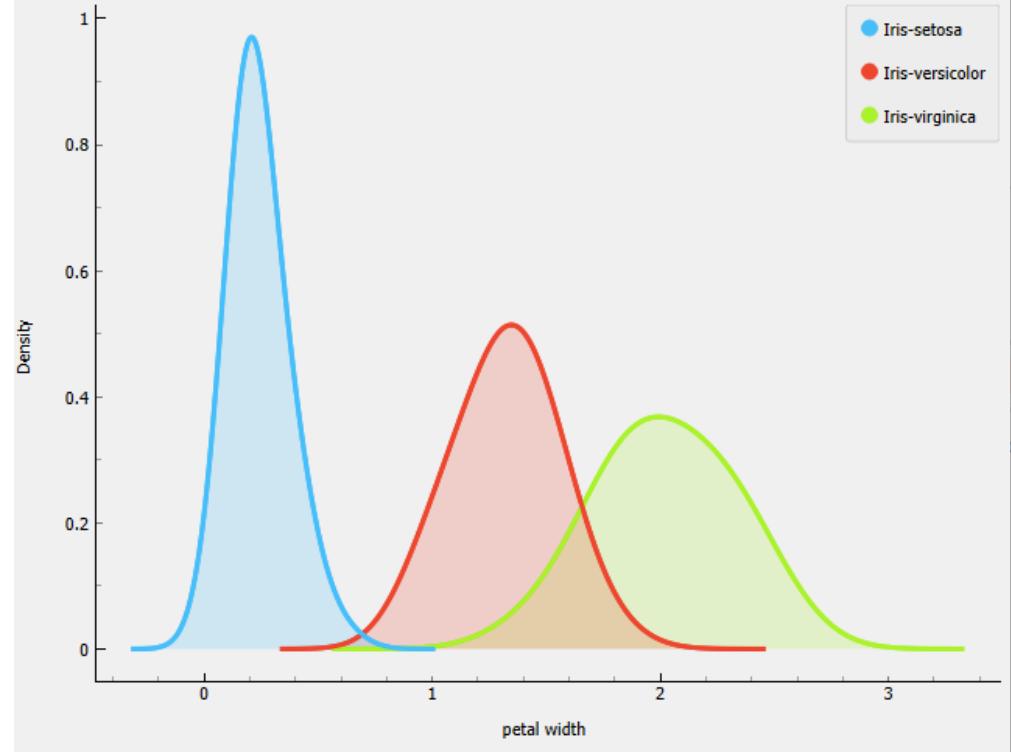


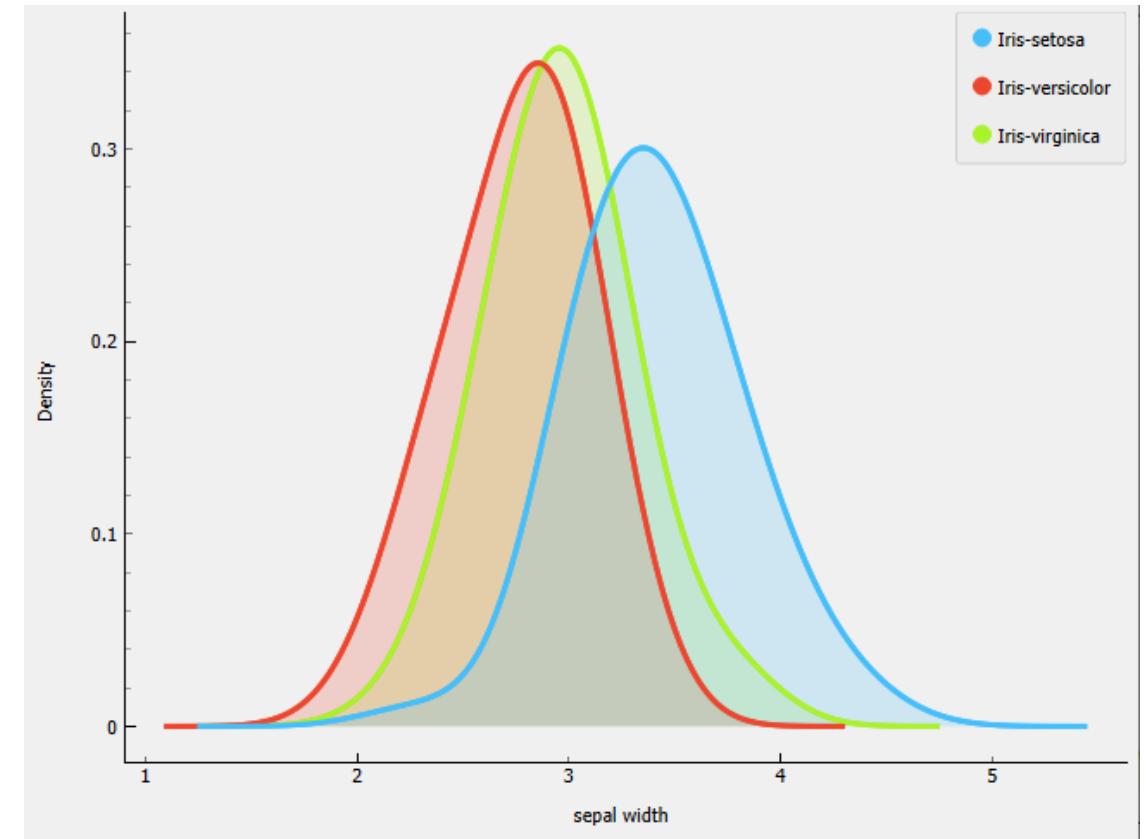
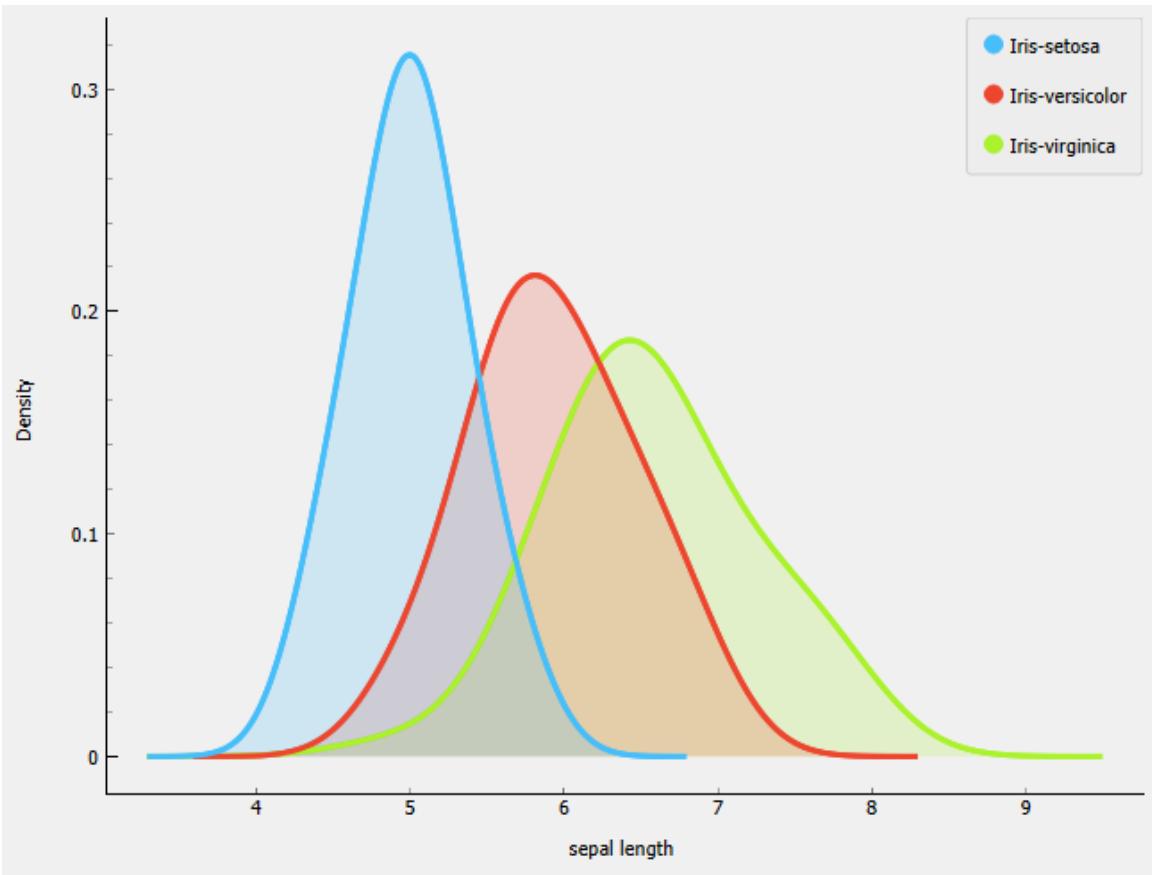




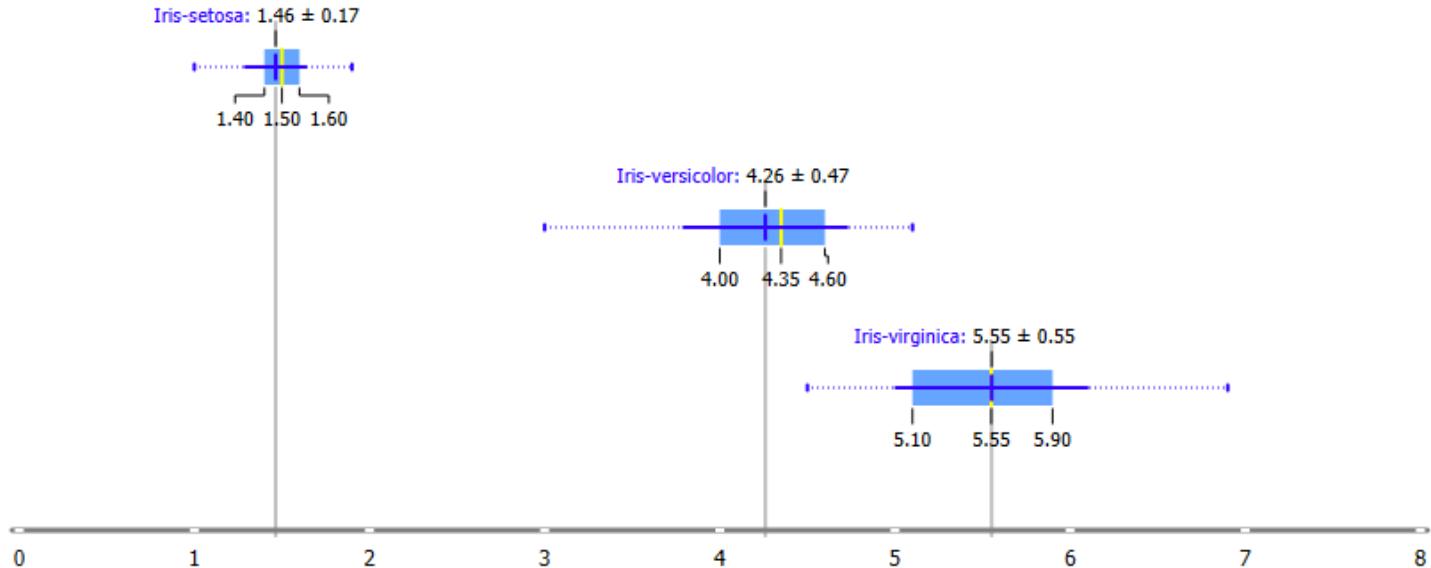




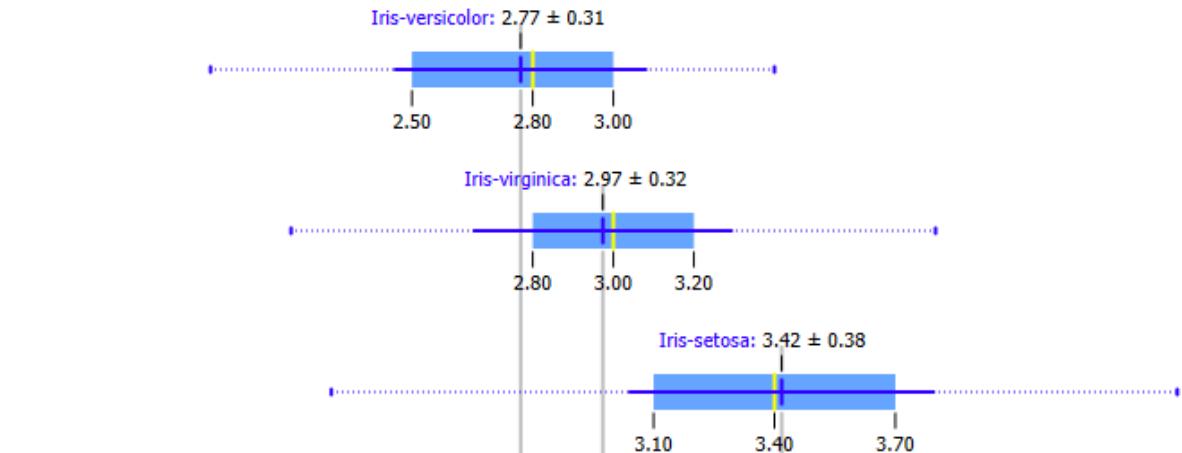




## Petal length

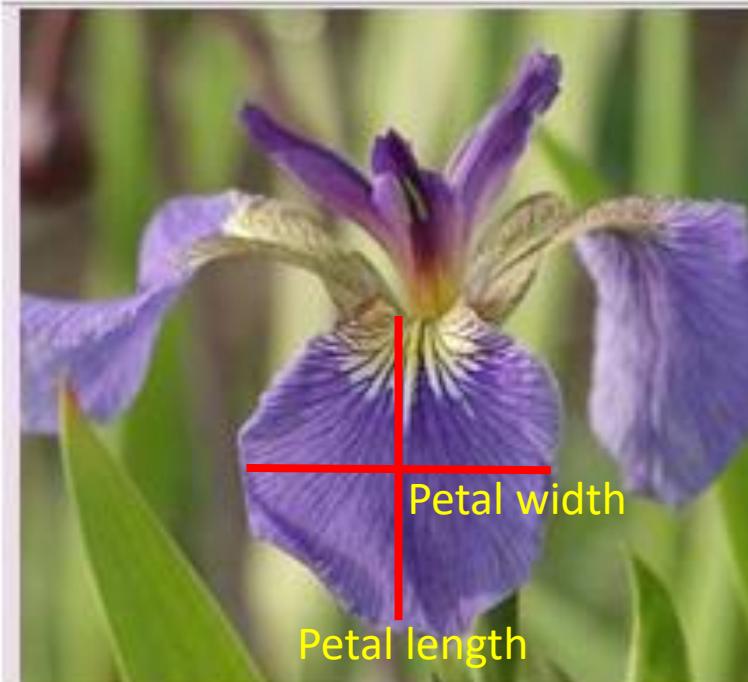


## Sepal width



# Características (Atributos)

- Es una propiedad (característica) medible de un fenómeno que está siendo observado. Es importante obtener características informativas, discriminantes e independientes para diferenciar entre distintas clases.



Setosa Iris



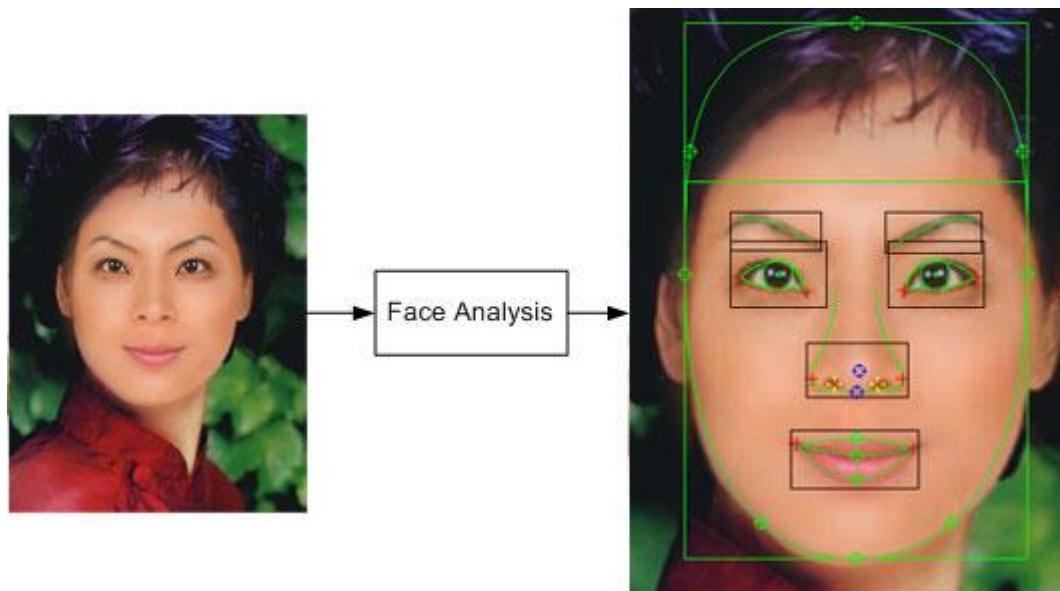
Versicolor Iris



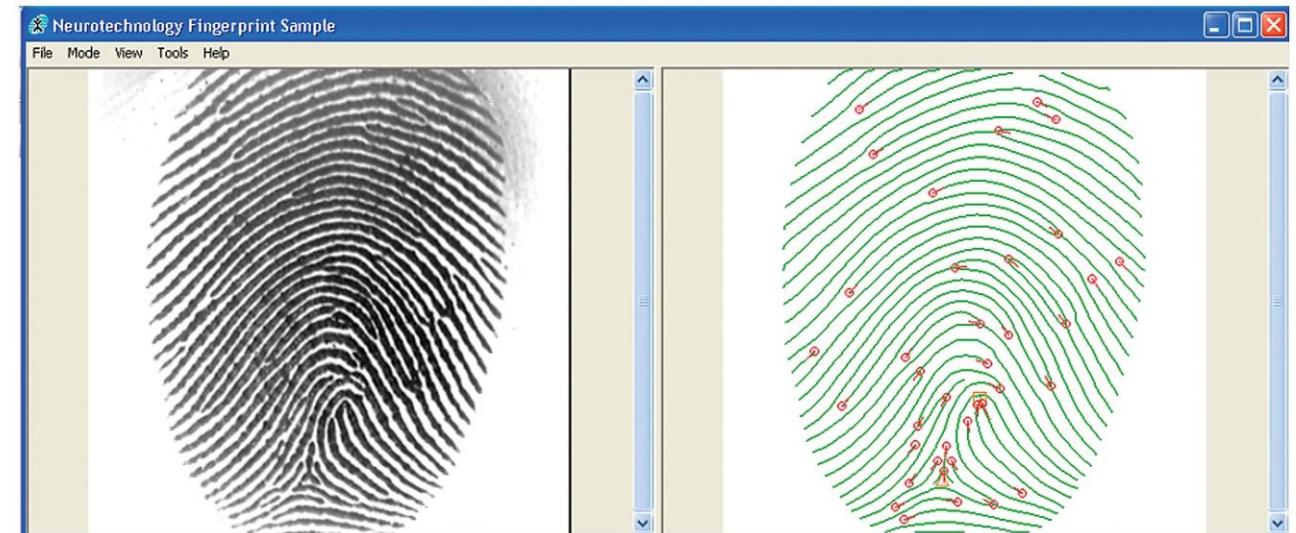
Virginica Iris

# Ejemplos de extracción de características

Análisis facial

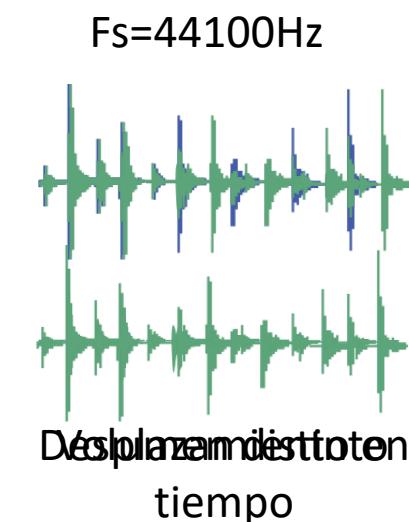
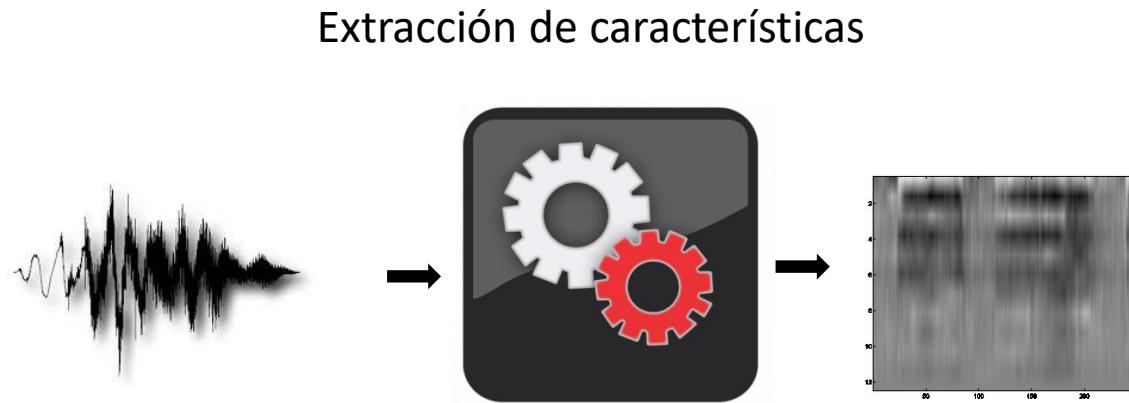


Huella digital de dedos

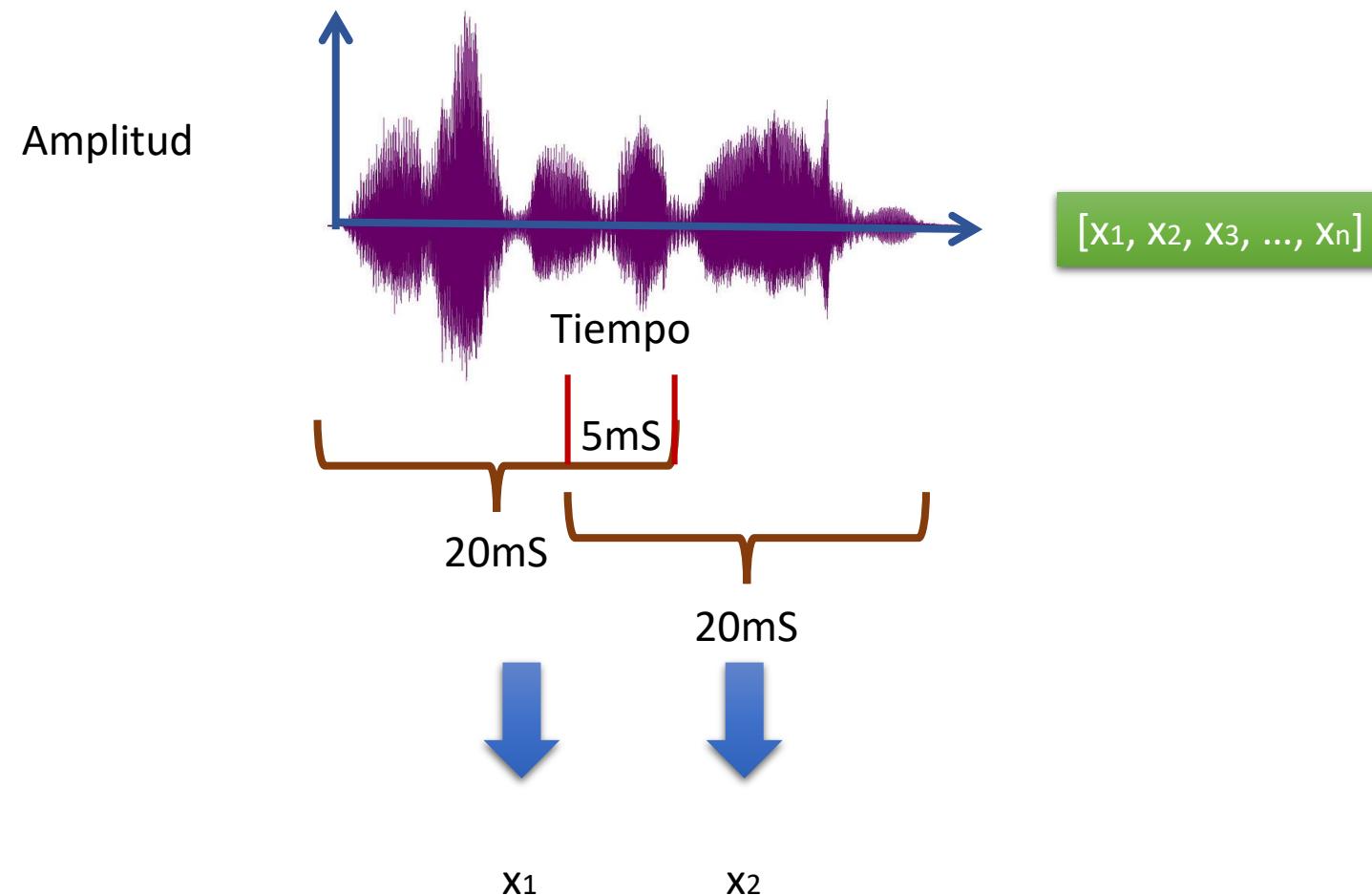


# Ejemplo de extracción de características en audio

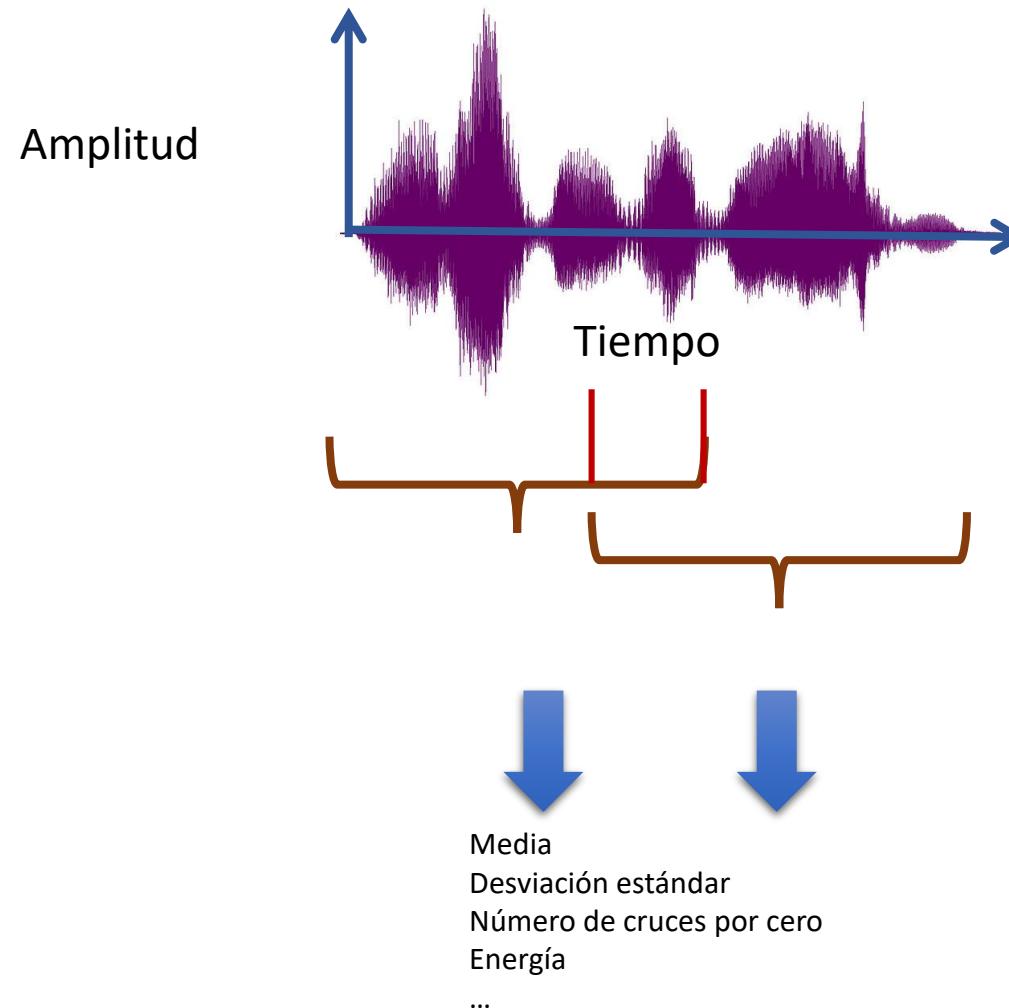
**La extracción de características** de audio se refiere a obtener una representación que puede usarse para medir similitud entre dos señales auditivas. La mayoría de las características reducen la cantidad de datos a almacenar y el tiempo de procesamiento utilizado en la comparación de señales.



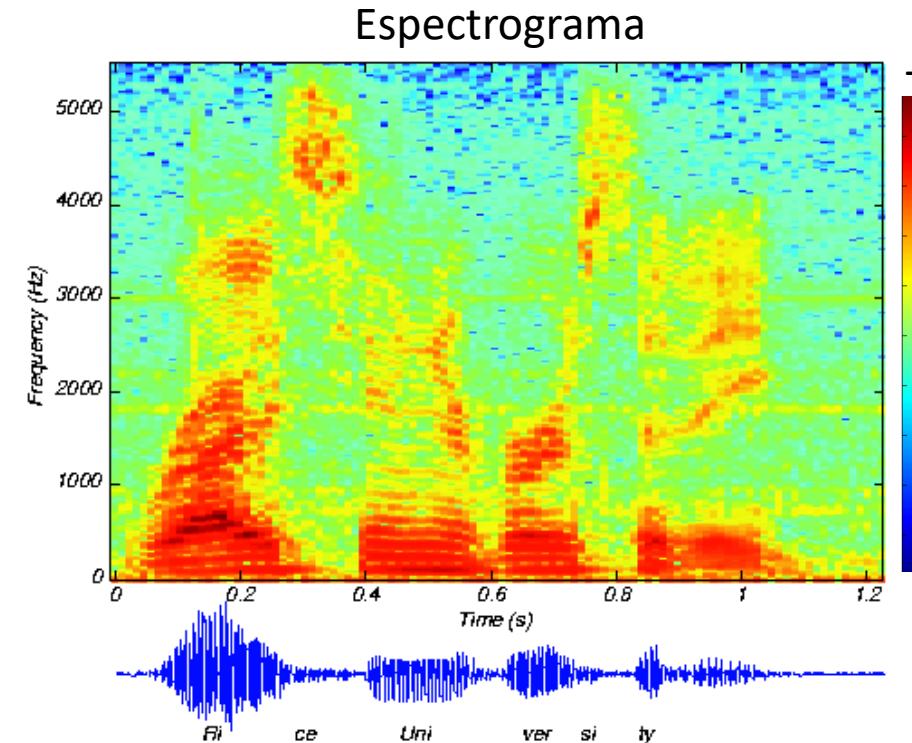
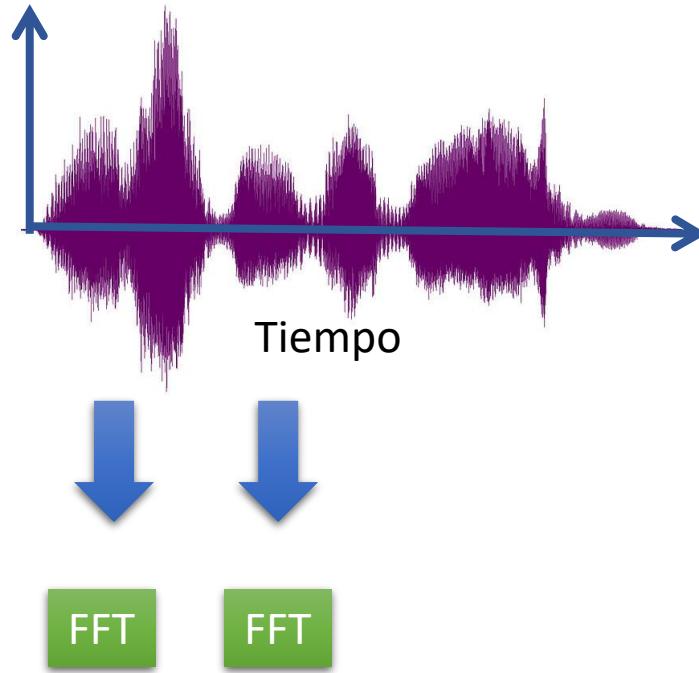
# Ejemplo de extracción de características en audio



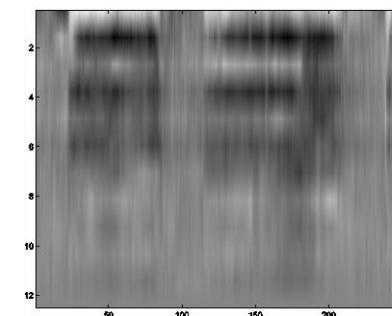
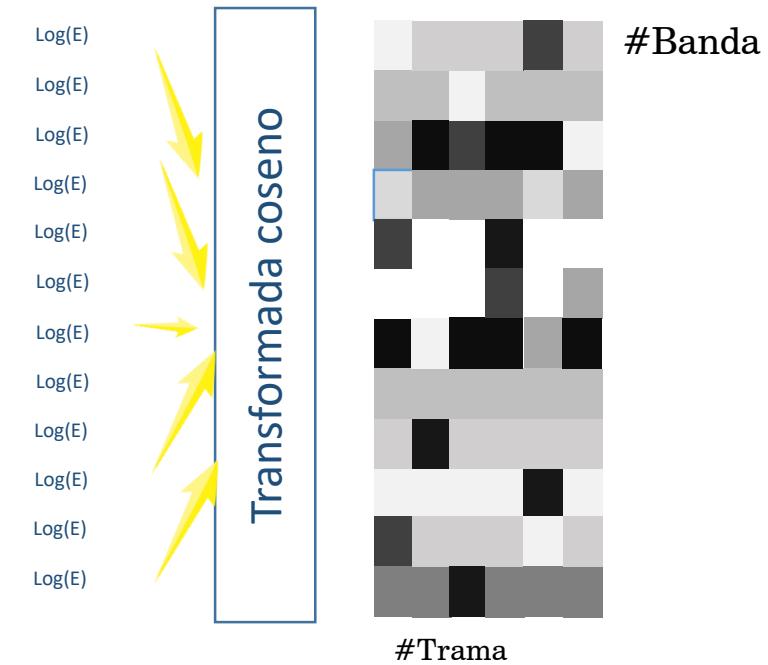
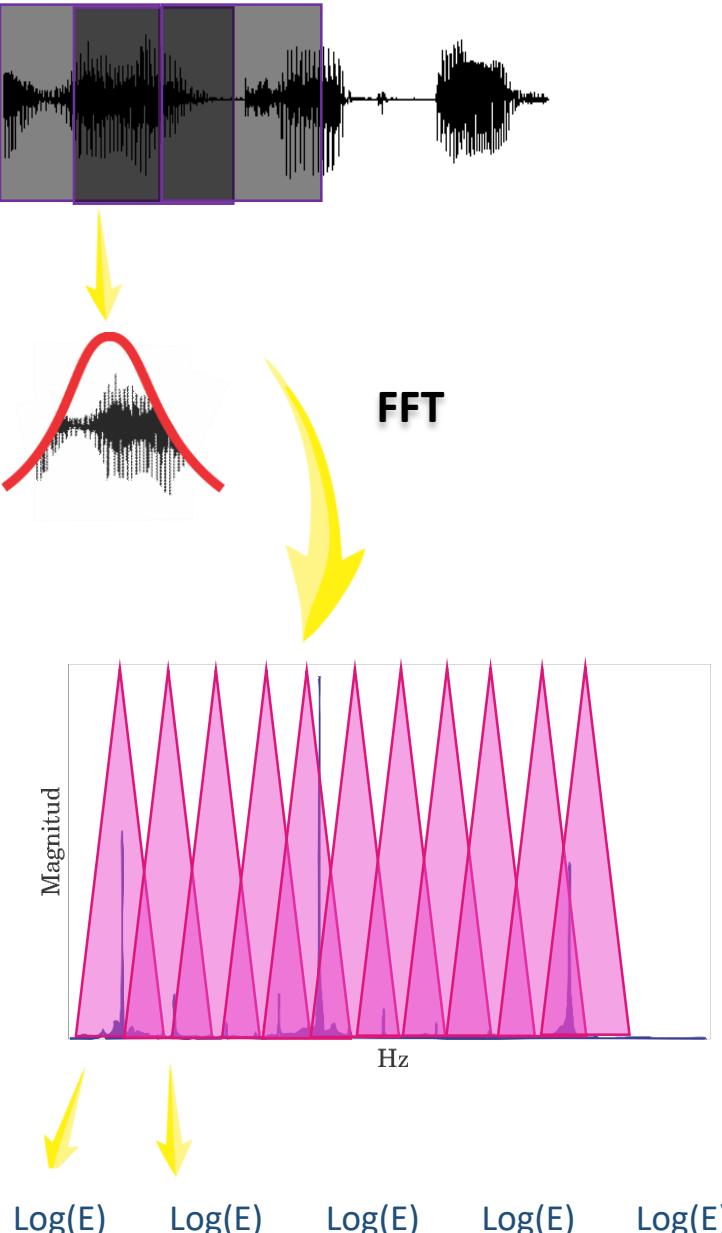
# Ejemplo de extracción de características en audio



# Ejemplo de extracción de características en audio



# Ejemplo de extracción de características en audio



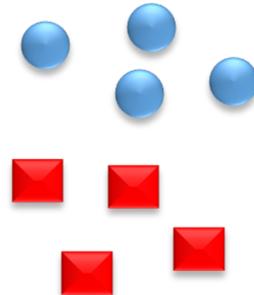
# Aprendizaje supervisado

Los métodos de **aprendizaje supervisado** intentan descubrir la relación entre atributos de entrada (variables independientes) y los atributos de salida (atributos dependientes).

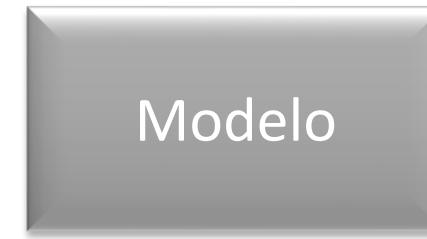
La relación descubierta se representa en una estructura que se refiere a un **modelo**. Usualmente, los modelos describen y explican fenómenos que están escondidos en las base de datos y se pueden utilizar para predecir los valores de los atributos de salida conociendo los valores de los atributos de entrada.

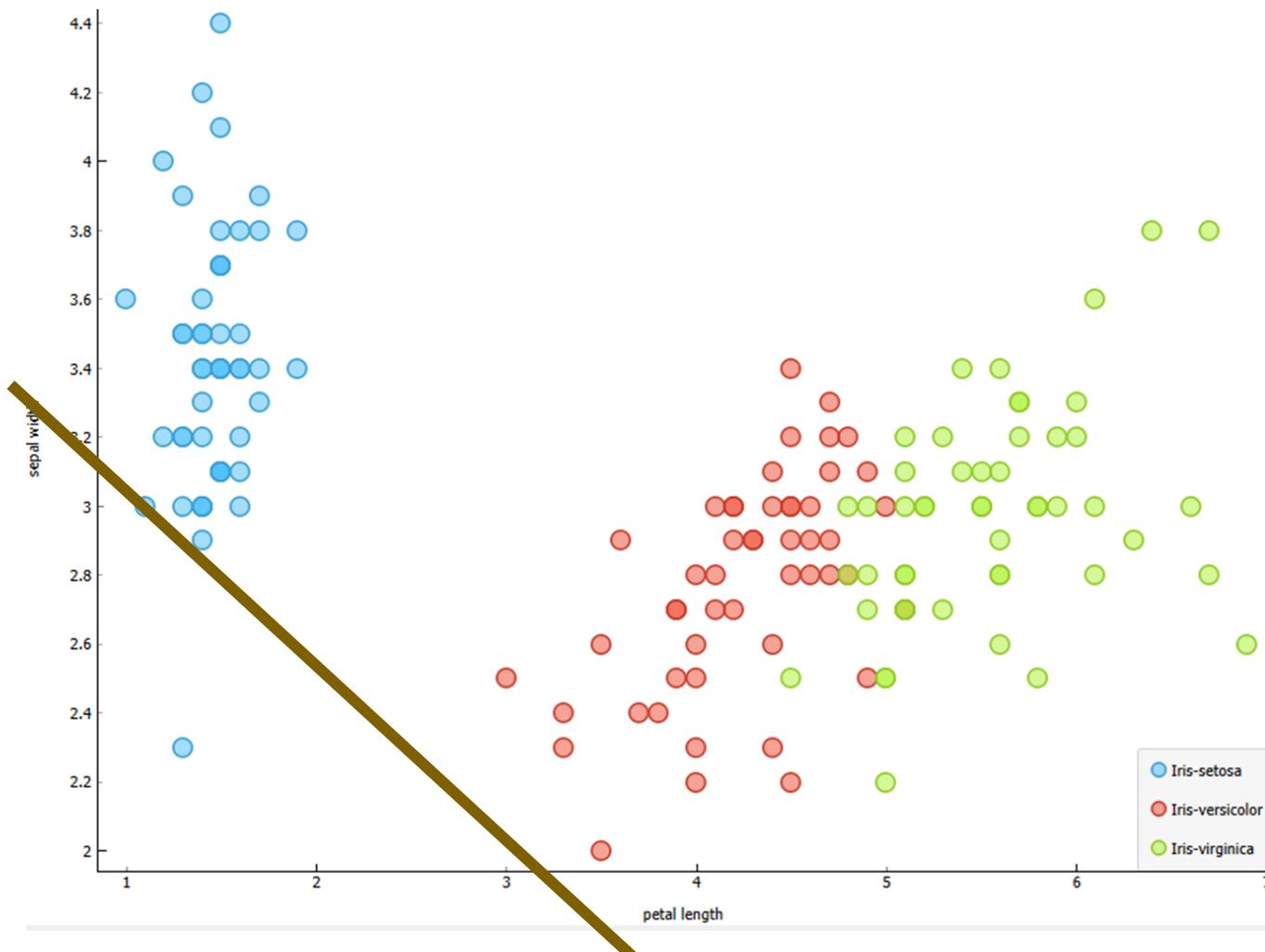
Aprenden de ejemplos etiquetados. En este tipo de aprendizaje se tiene un conocimiento a priori acerca de las clases que se utilizan para el entrenamiento.

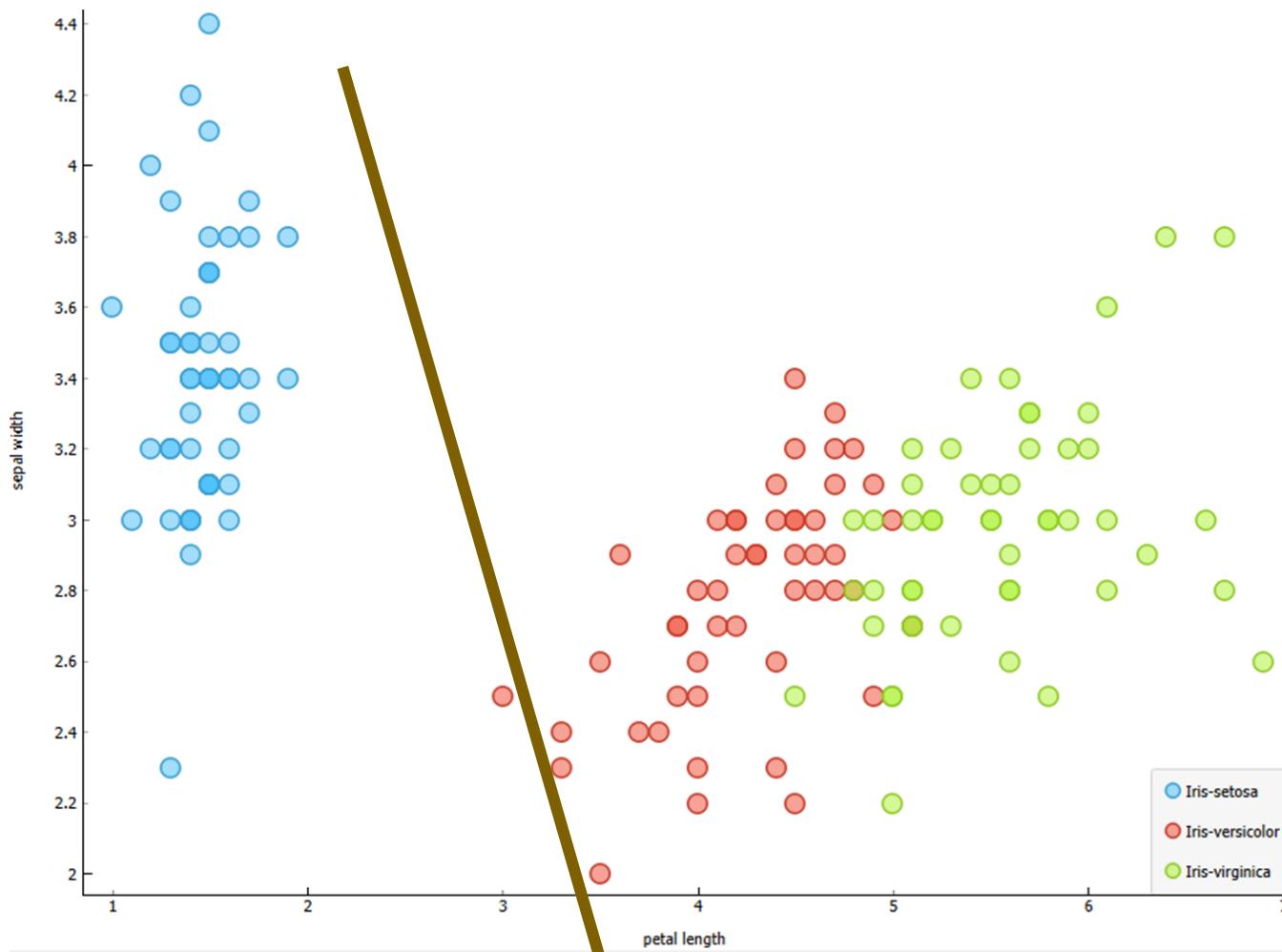
## Entrenamiento



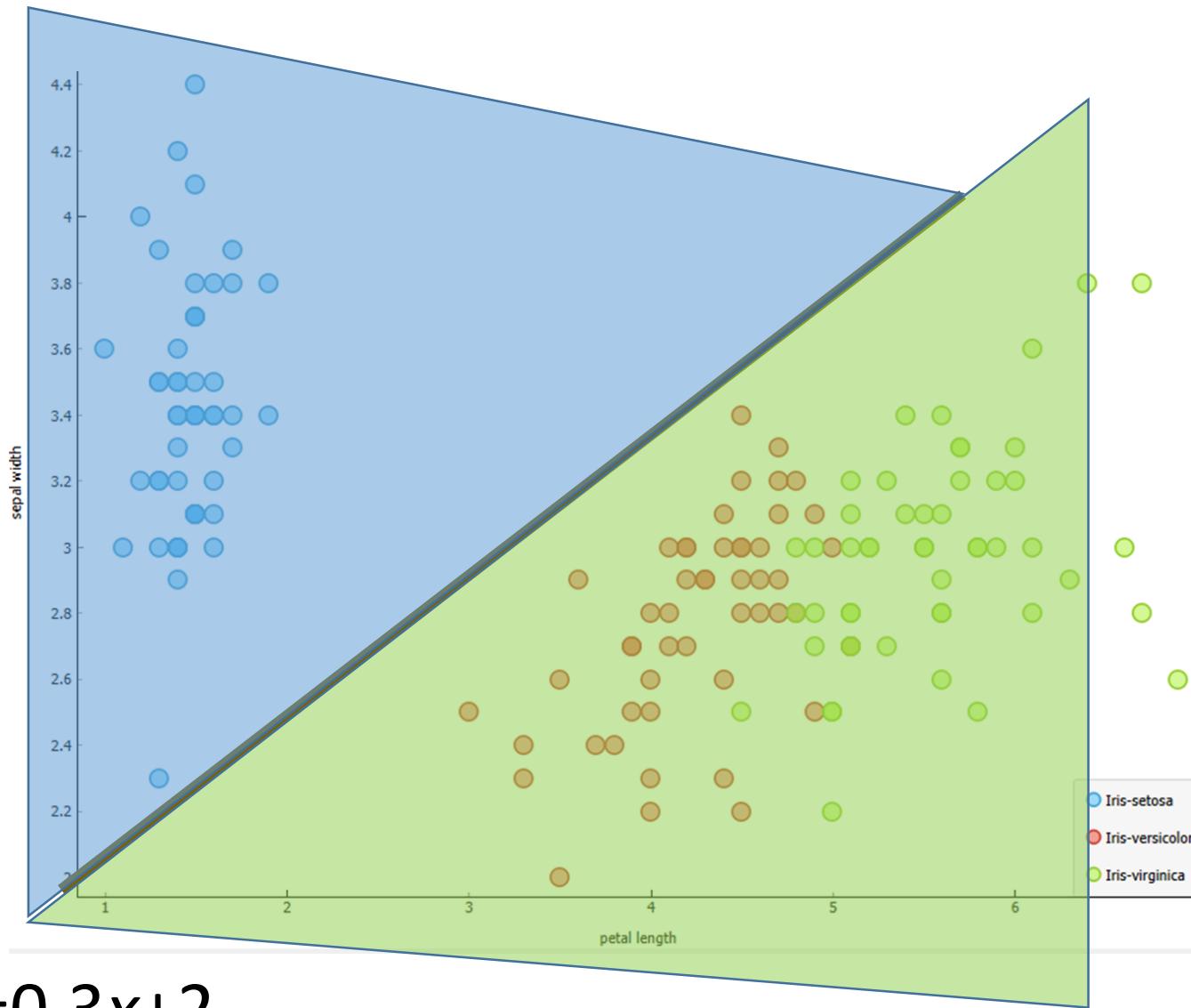
**Características  
del objeto:**  
Medidas  
Colores  
...





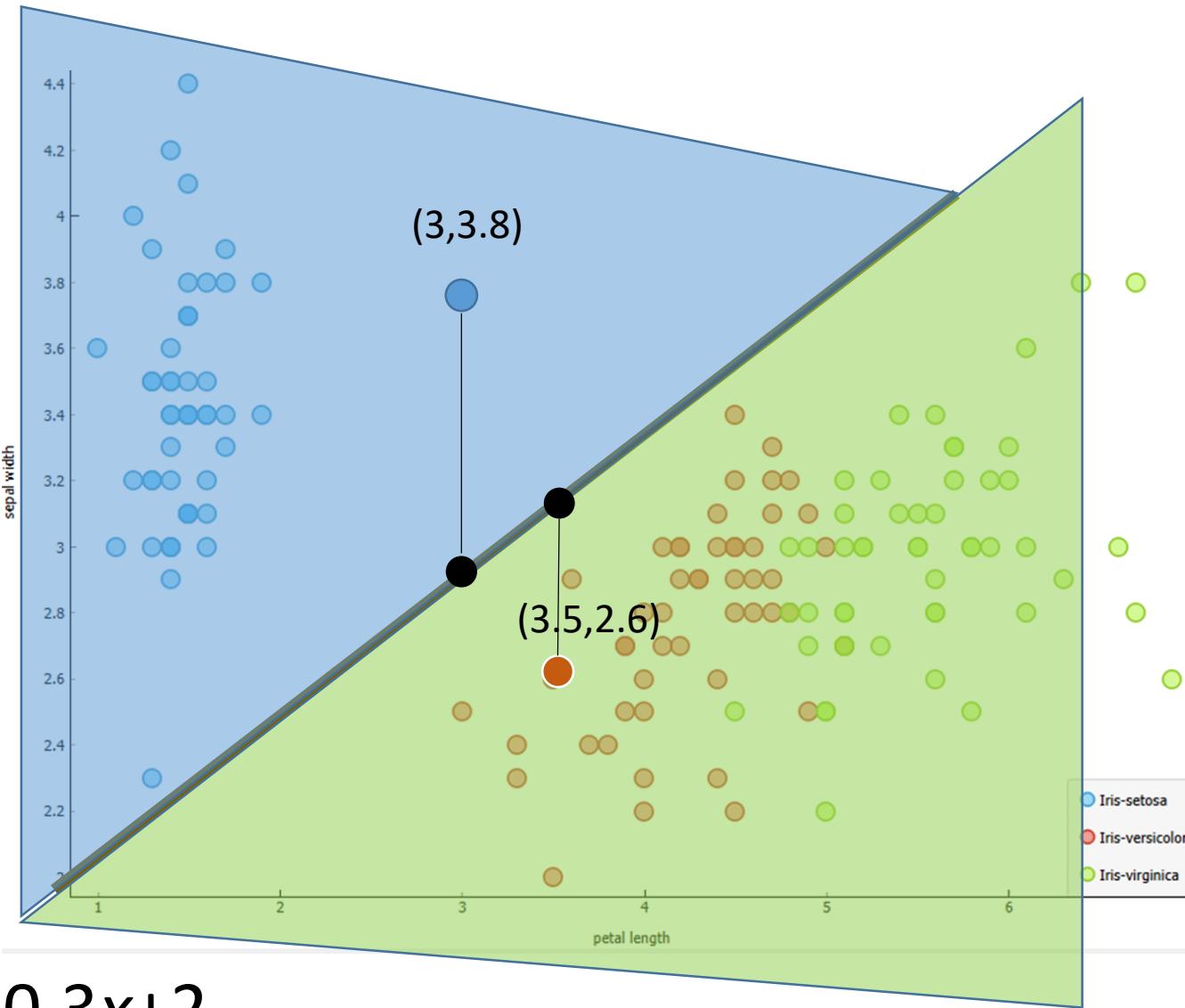


$$y = -2.5x + 8$$



$$y=0.3 (3)+2  
=2.9$$

+



$$y=0.3 (3.5)+2  
=3.05$$

-

# Actividad 3

# Detección de rostros con OpenCV

## Detección de objetos usando openCV

Curso: Aplicaciones del aprendizaje automático con Python

Actividad 3: Detección de objetos con OpenCV

Aquí vamos a probar una función integrada en openCV que sirve para detectar objetos. En este ejercicio detectaremos caras y ojos.

Objetivos:

1. Usar un modelo previamente entrenado para detectar rostros y ojos (función CascadeClassifier de openCV)

Dra. Jessica Beltrán Márquez

[www.jessicabeltran.net](http://www.jessicabeltran.net)

### Obtener la imagen

Importamos las bibliotecas que vamos a utilizar.

Leemos los modelos de clasificación para *rostros frontales* y para *ojos*. Estos modelos contienen valores numéricos que ya están entrenados.

Obtenemos la imagen a la cual le detectaremos los rostros y los ojos.

Convertimos la imagen a grises ya que así la requiere el clasificador.

Desplegamos la imagen original.

In [3]: ►

```
import numpy as np
import cv2
from matplotlib import pyplot as plt

#Se cargan los modelos para detectar rostros y ojos. NOTA: Estos modelos ya fueron entrenados previamente, solo los
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml') #Revisa que este archivo está en tu carpeta
eye_cascade = cv2.CascadeClassifier('haarcascade_eye.xml') #Revisa que este archivo está en tu carpeta
```

# Detección de rostros con OpenCV

Name	Status	Date modified	Type
.ipynb_checkpoints		11-Jun-19 4:58 PM	File folder
images		26-Sep-19 1:24 PM	File folder
Detección de objetos con OpenCV.ipynb	⟳	06-Oct-19 3:44 PM	IPYNB File
haarcascade_eye		07-Sep-17 11:53 A...	XML Document
haarcascade_frontalface_default		07-Sep-17 11:52 A...	XML Document

# Detección de rostros con OpenCV

```
In [1]: # import numpy as np
import cv2
from matplotlib import pyplot as plt

#Se cargan los modelos para detectar rostros y ojos. NOTA: Estos modelos ya fueron entrenados previamente, solo los
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml') #Revisa que este archivo esté en tu carpeta
eye_cascade = cv2.CascadeClassifier('haarcascade_eye.xml') #Revisa que este archivo esté en tu carpeta

#Se lee la imagen
img = cv2.imread('./images/twofaces1.jpg')

#Se convierte a escala de grises
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
imgToShow = img[:, :, ::-1]

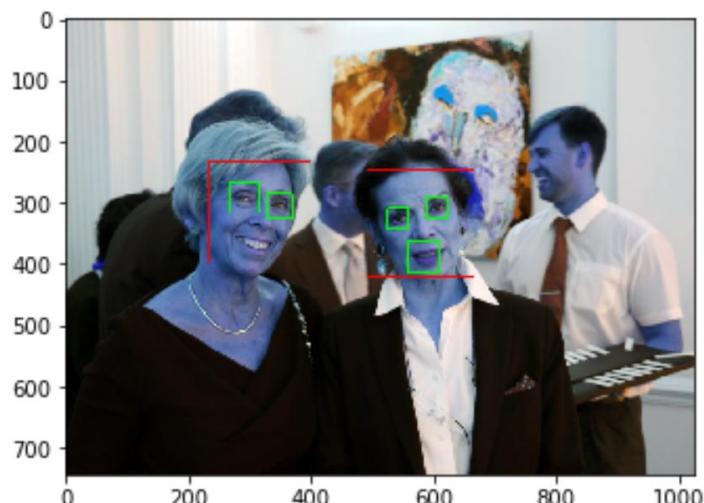
#Se muestra la imagen
plt.imshow(imgToShow)
plt.show()
```



<Figure size 640x480 with 1 Axes>

# Detección de rostros con OpenCV

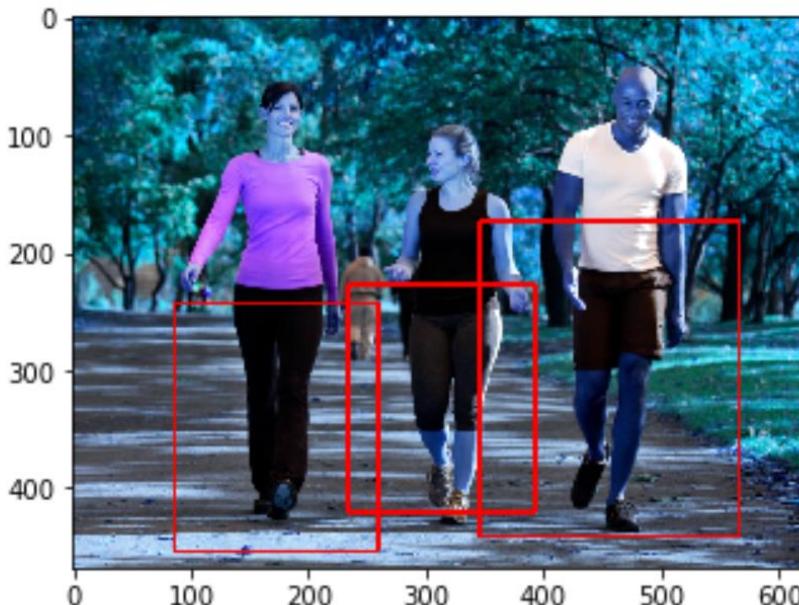
```
In [2]: #Se utiliza el modelo detector de rostros que ya cargamos anteriormente  
faces = face_cascade.detectMultiScale(gray, 1.3, 5) #Indicamos los parámetros  
#faces = face_cascade.detectMultiScale(gray, 1.01, 1)  
for (x,y,w,h) in faces: #iterando en las coordenadas en donde se detectaron rostros con el modelo  
    cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2) #Se dibujan los rectángulos en donde se encontraron rostros  
    roi_gray = gray[y:y+h, x:x+w] #region de interés de la imagen en gris  
    roi_color = img[y:y+h, x:x+w] #region de interés de la imagen a color  
    #Se utiliza el detector de ojos que ya cargamos anteriormente  
    eyes = eye_cascade.detectMultiScale(roi_gray)  
    for (ex,ey,ew,eh) in eyes: #Iterando en las coordenadas de los ojos encontrados  
        cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2) #Se dibujan rectángulos en donde se encontraron ojos  
  
plt.imshow(img)  
plt.show()  
#cv2.imshow('img',img)  
#cv2.waitKey(0)  
#cv2.destroyAllWindows()
```



# Detección de rostros con OpenCV

```
In [5]: lowerBody_cascade = cv2.CascadeClassifier('haarcascade_lowerbody.xml') #Revisa que este archivo está en tu car  
imgCaminando = cv2.imread('./images/caminando.png')  
arrlowerBody = lowerBody_cascade.detectMultiScale(imgCaminando)  
if arrlowerBody is not None:  
    for (x,y,w,h) in arrlowerBody:  
        cv2.rectangle(imgCaminando, (x,y), (x+w,y+h), (255,0,0), 2)  
    print ('body found')  
  
plt.imshow(imgCaminando)  
plt.show()
```

body found



# Más conceptos

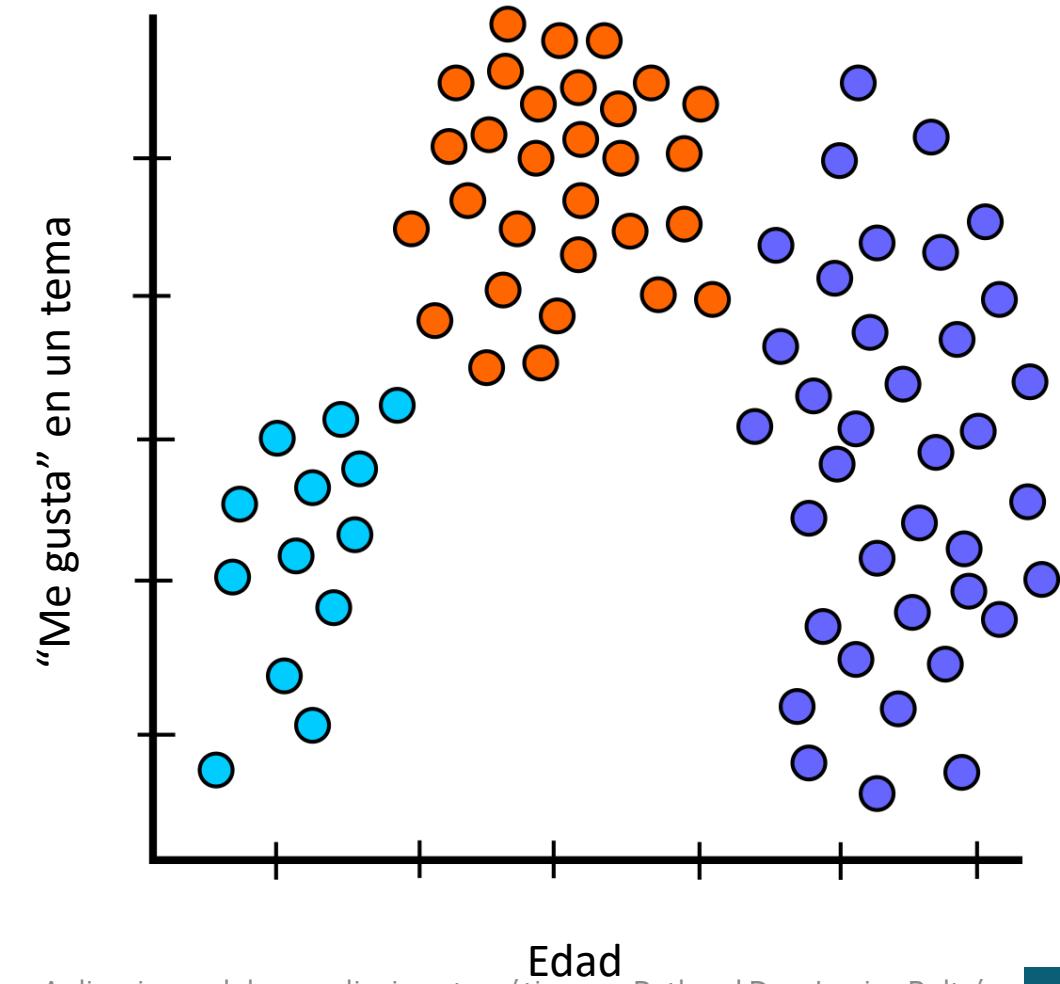
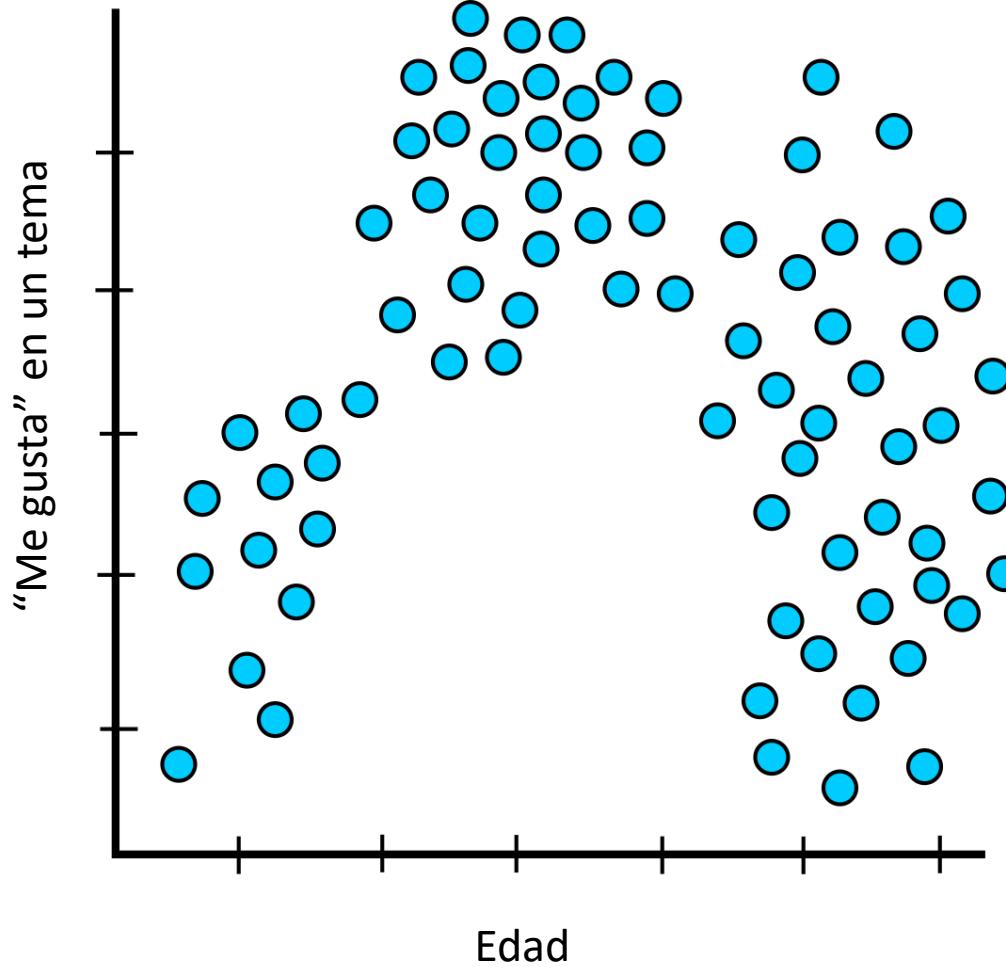
# Aprendizaje no supervisado

Los métodos de **aprendizaje no supervisado** no necesitan de más información que la escena que se va a aprender y algunos parámetros que limiten el número de clases. Estos mecanismos basan su efecto en la búsqueda de clases con suficiente separabilidad como para conseguir diferenciar unos elementos de otros.

No se necesita indicar la categoría de los datos de entrada.

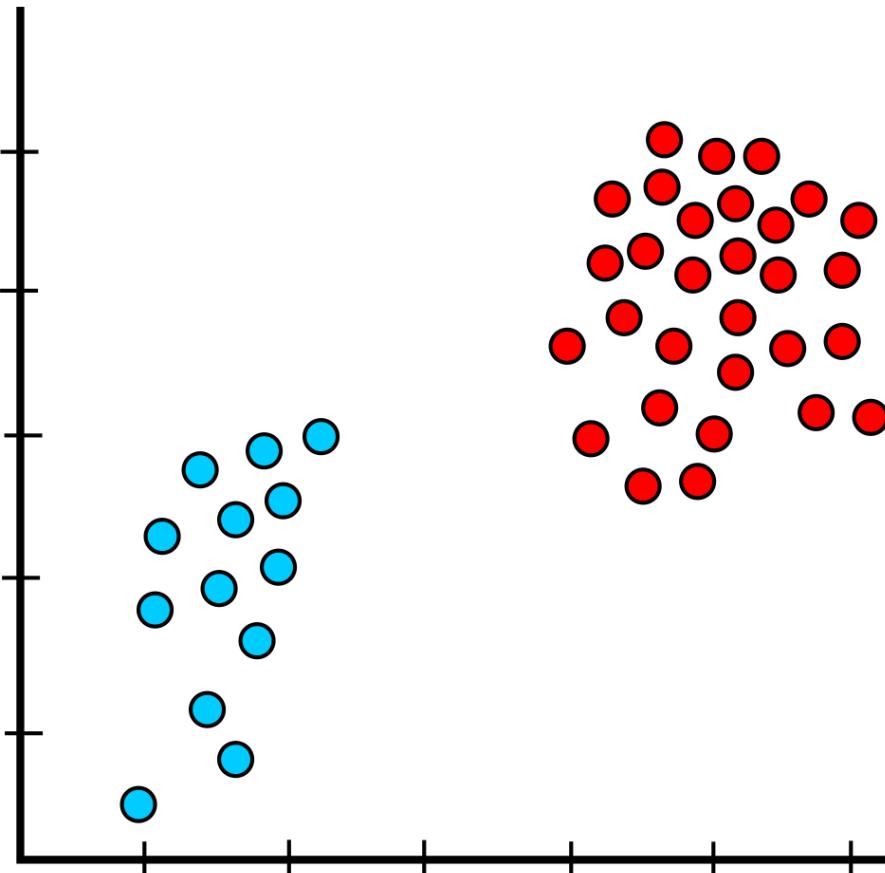
En este tipo de aprendizaje no se tiene un conocimiento a priori acerca de las clase que se utilizan para el entrenamiento.

# Aprendizaje no supervisado

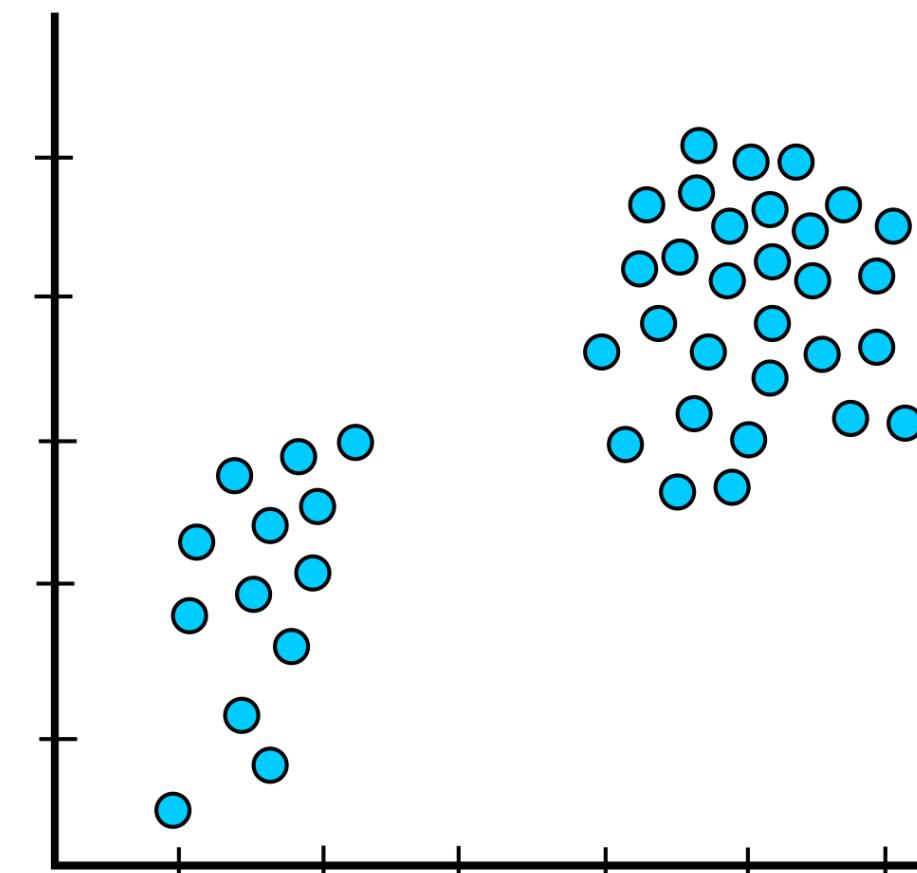


# Aprendizaje supervisado vs no supervisado

Supervisado

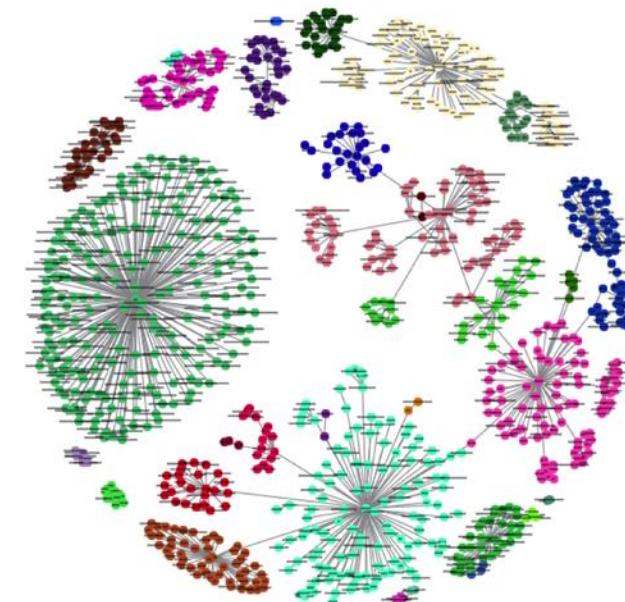
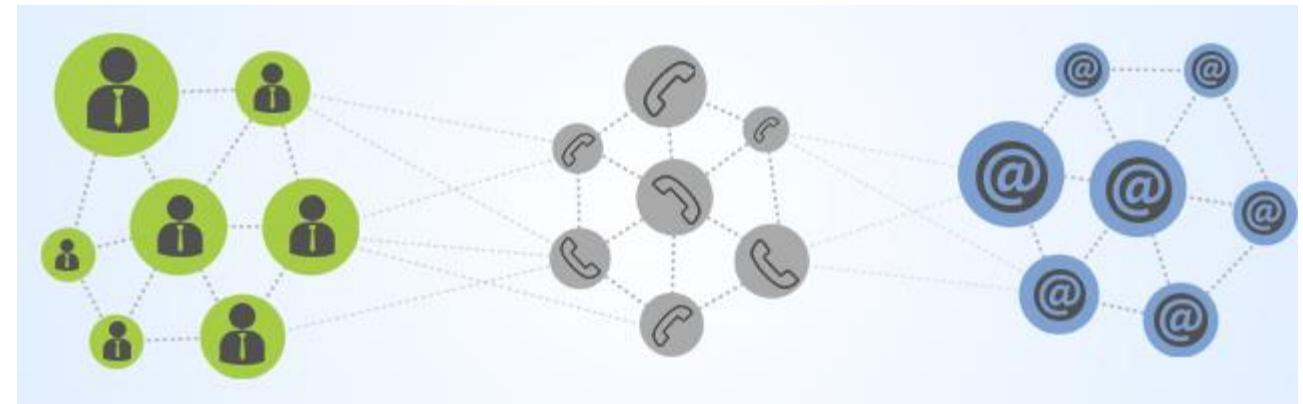


No Supervisado



# Ejemplos de aplicaciones de agrupamientos en datos de negocios

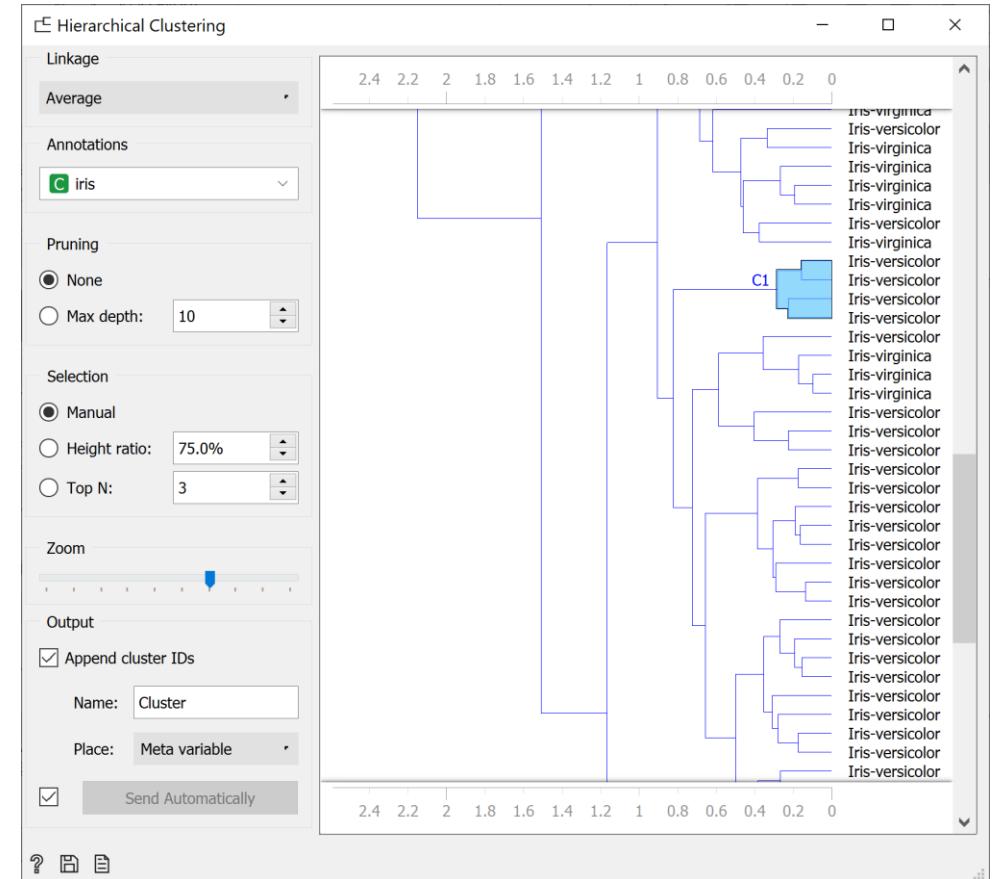
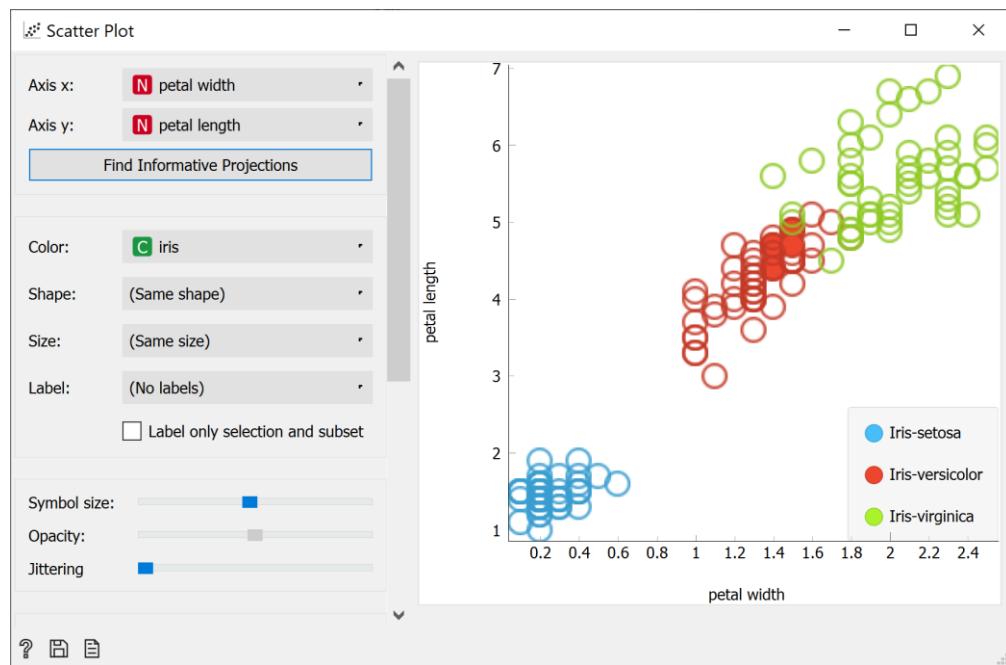
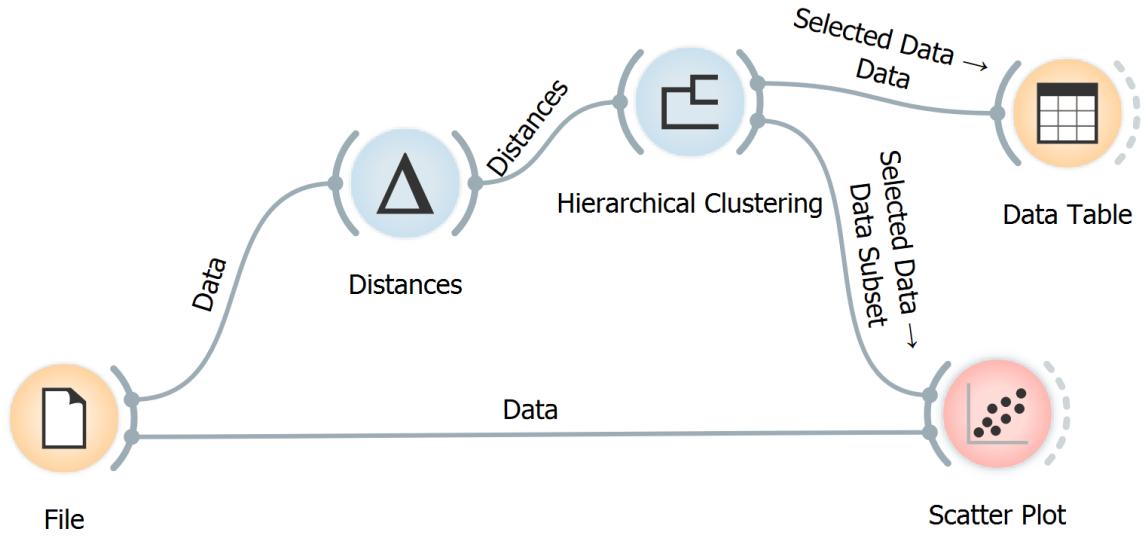
- Agrupamiento de página web
- Segmentación del Mercado.
- Comportamiento de los clientes
- Campañas de venta
- Retención de clientes.



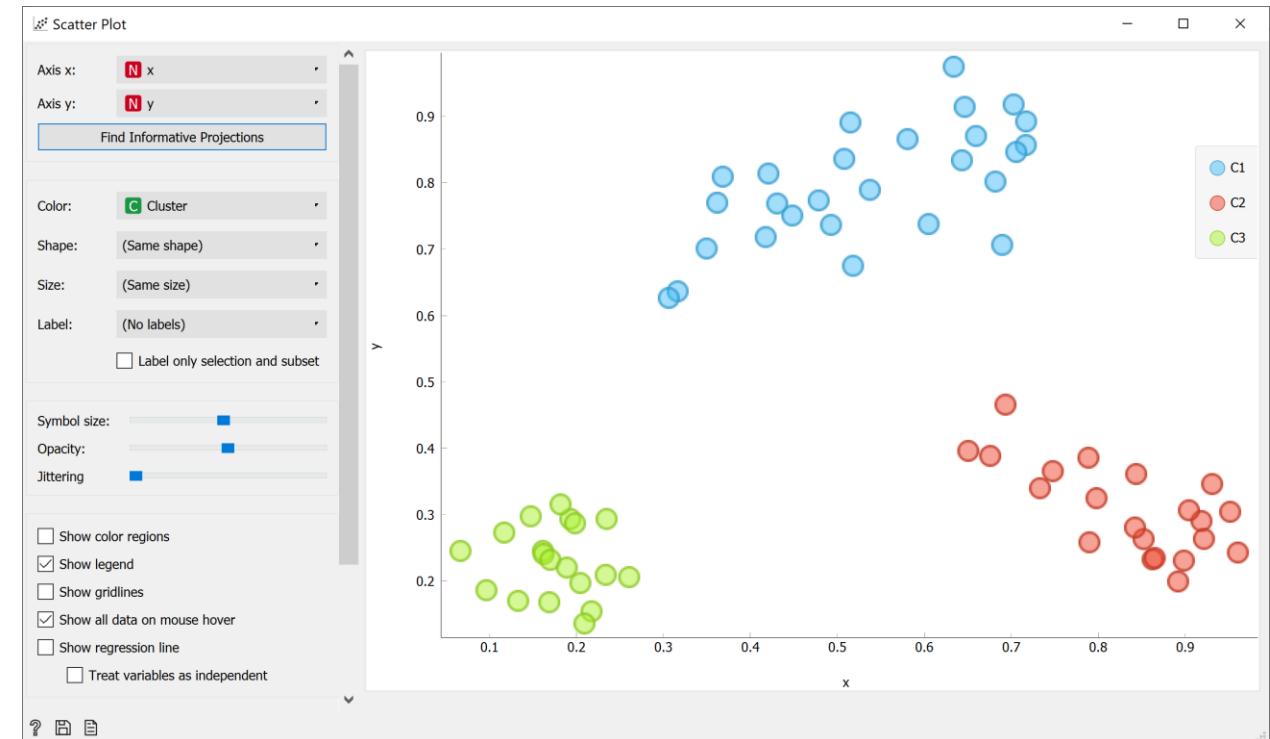
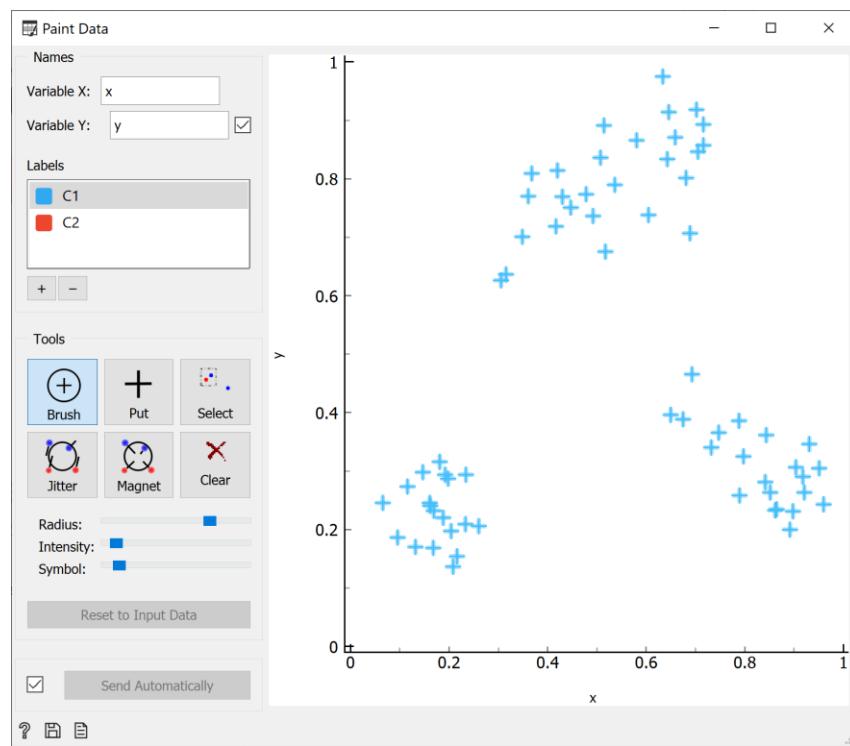
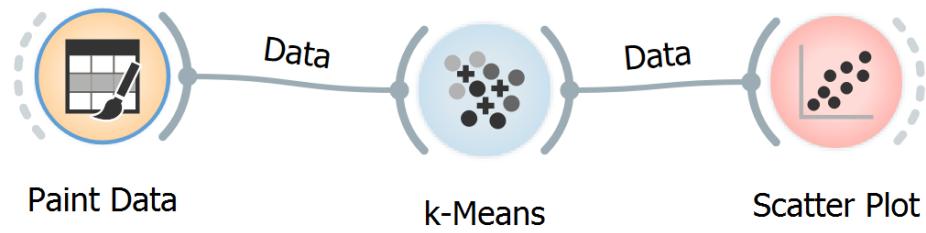
# Ejemplo de aplicación de agrupamiento

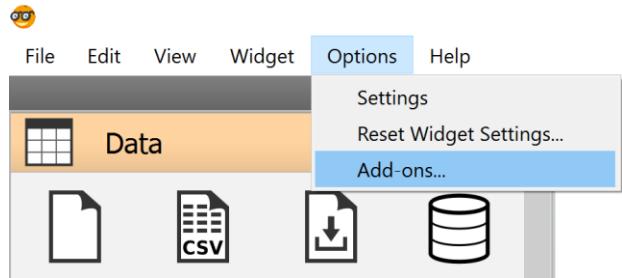
En una tienda departamental de Illinois se vendían trajes de hombres. Todas las tiendas excepto una presentaron un incremento de al menos 100% de las ganancias. Al hacer un *agrupamiento (clustering)*, se notó que la tienda excepción era la única tienda que promocionaba su producto por la radio en vez de por televisión.

# Agrupamiento jerárquico con Orange



# Agrupamiento K-medias con Orange





**Add-ons**

Name	Version	Action
<input type="checkbox"/> Orange3-Bioinformatics	4.0.0	
<input checked="" type="checkbox"/> Orange3-Educational	0.2.1	
<input type="checkbox"/> Orange3-Geo	0.2.4.post1	
<input checked="" type="checkbox"/> Orange3-ImageAnalytics	0.4.0	
<input type="checkbox"/> Orange3-Network	1.5.0	
<input type="checkbox"/> Orange3-Prototypes	0.12.0	
<input type="checkbox"/> Orange3-SingleCell	1.3.0	
<input type="checkbox"/> Orange3-Text	0.7.3	
<input type="checkbox"/> Orange3-Textable	3.1.4	
<input type="checkbox"/> Orange3-Timeseries	0.3.3	

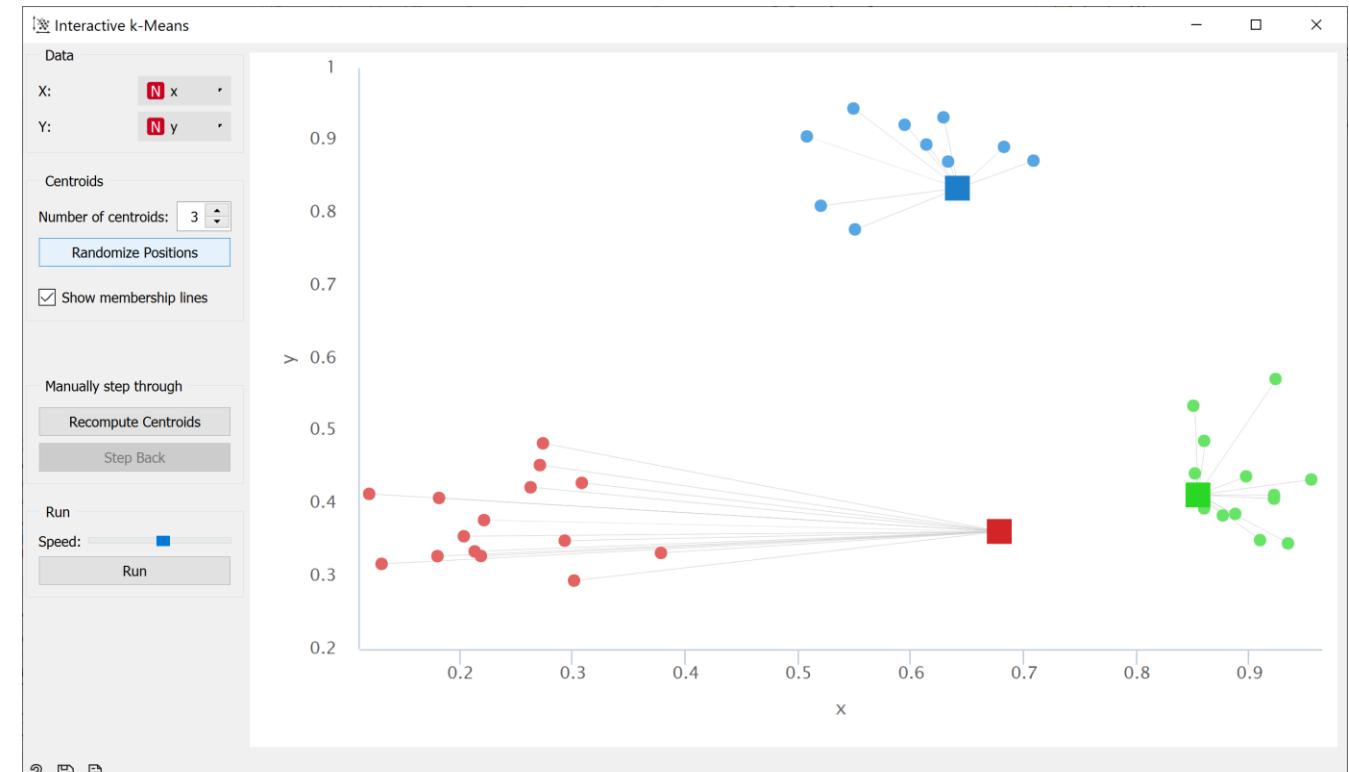
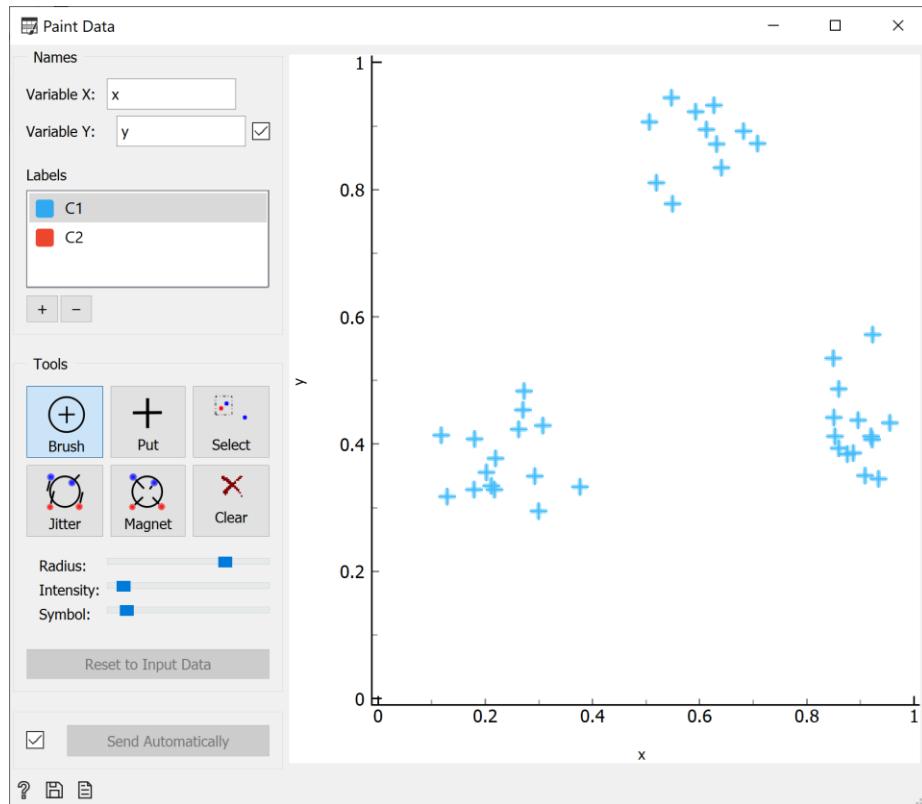
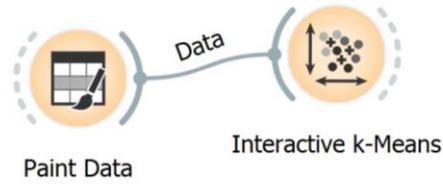
**Spectral Orange**

Orange add-on for spectral data analysis. Provides spectra and hyperspectra plotting, interpolation, preprocessing (cutting, smoothing, normalization), and integration. Supports common spectral file formats.

OK Cancel

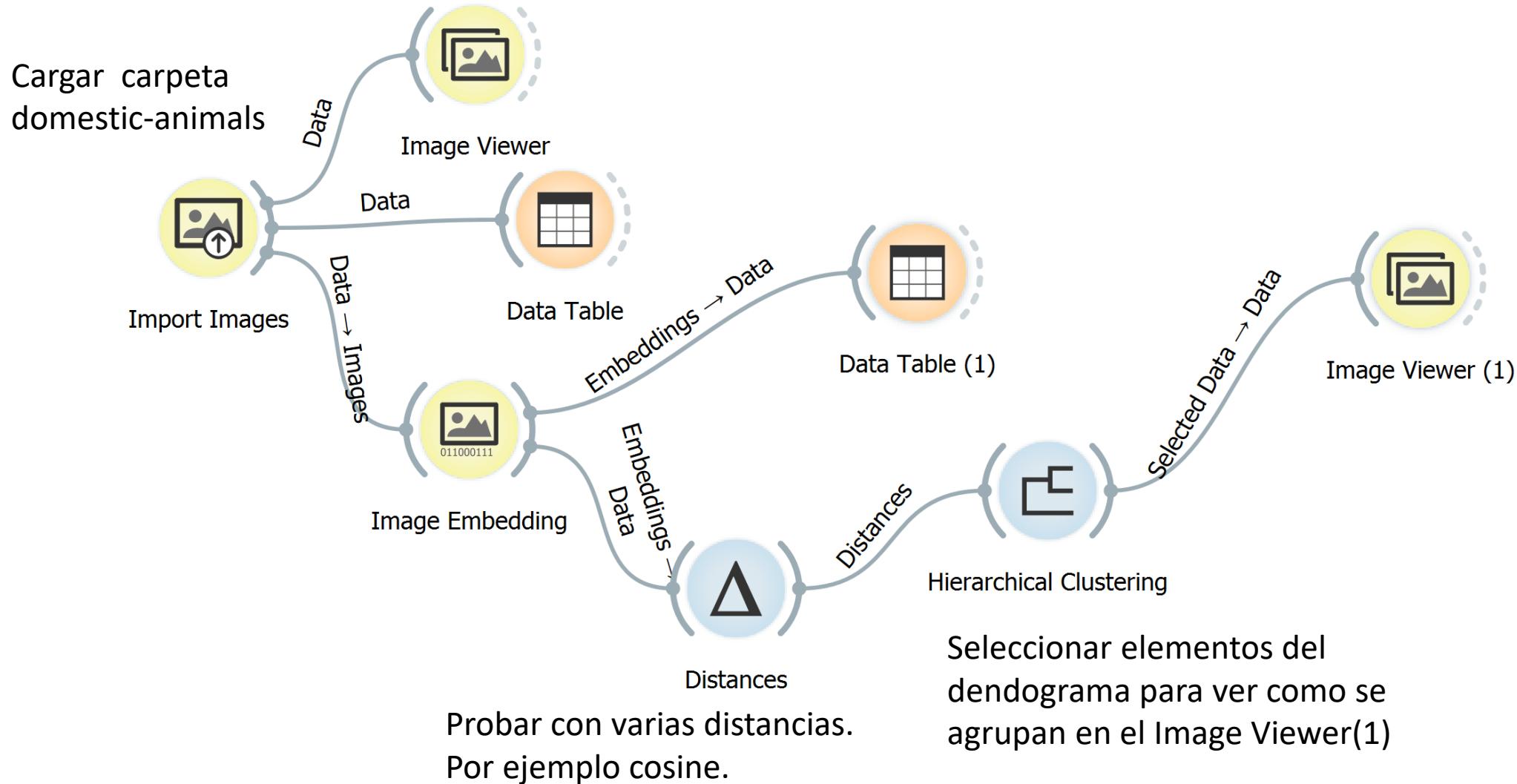
Seleccionar Educational

Seleccionar ImageAnalytics



Se requiere el add on Education

# Ejemplo de aplicación de agrupamiento con Orange



Se requiere el add on Image analytics

# Más conceptos

# Conjuntos de datos

## Entrenamiento

- Es un conjunto de datos que se utiliza para descubrir posibles relaciones entre los datos que pudieran usarse para hacer predicción.

## Prueba

- Es un conjunto de datos que se utiliza para conocer la capacidad y utilidad de la predicción.

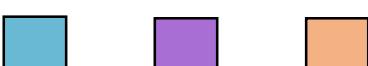
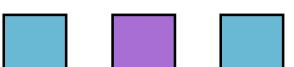
## Validación cruzada

- Técnica que consiste en hacer particiones la base de datos en varios conjuntos y evaluar los resultados en cada una de las particiones.

## Ground truth

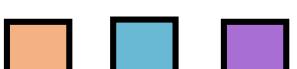
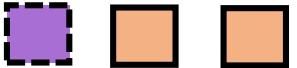
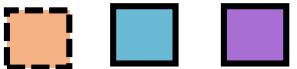
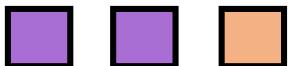
- Conjunto etiquetado con datos confiables para entrenamiento y prueba.

# Datos de entrenamiento y de prueba



Clase 1   Clase 2   Clase 3

# Datos de entrenamiento y de prueba



Clase 1   Clase 2   Clase 3

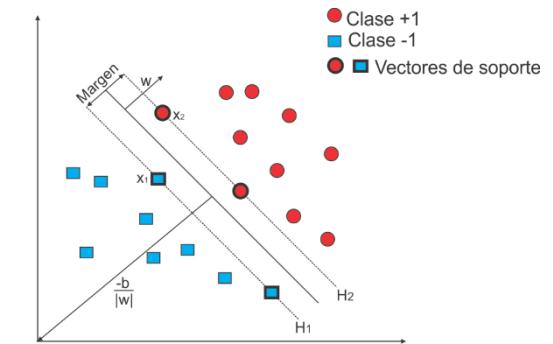
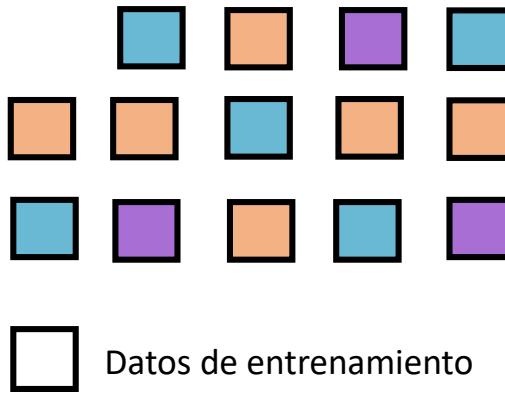


Datos de entrenamiento

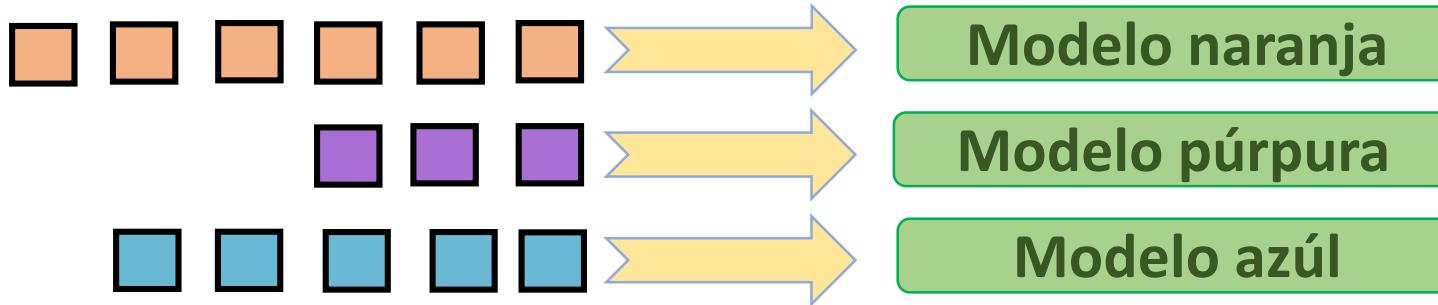


Datos de prueba

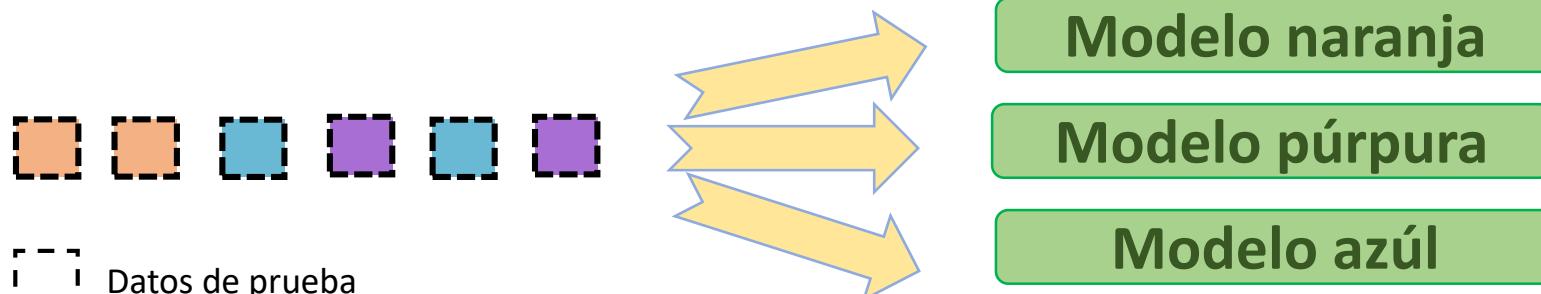
# Datos de entrenamiento y de prueba



# Datos de entrenamiento y de prueba

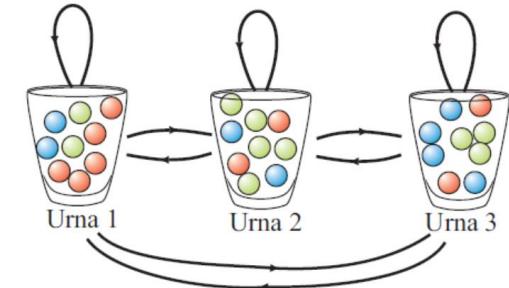


Datos de entrenamiento

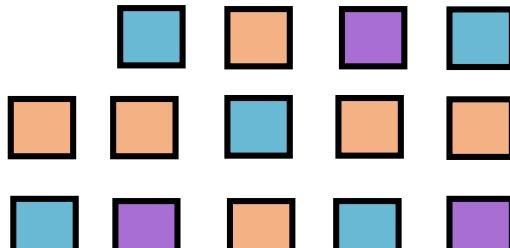


Datos de prueba

Símbolos emitidos por las urnas(estados) = {●, ○, △}



# Modelos

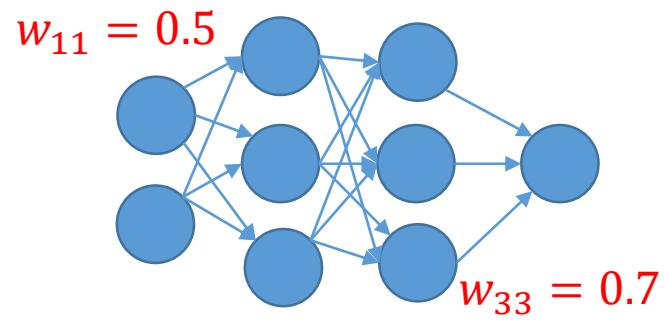


Datos de entrenamiento



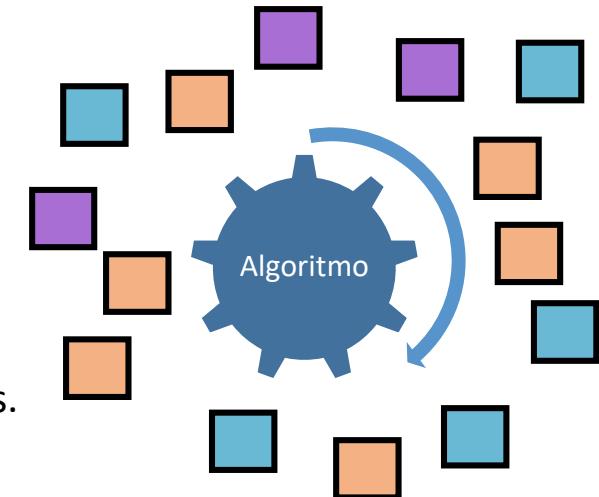
Modelo

Estimación de parámetros



El parámetro de un modelo es una variable que es interna al modelo y se estima a partir de los datos.  
Los parámetros se estiman comúnmente utilizando un algoritmo de optimización.

Algoritmo para estimación de parámetros



Ejemplos:

- Los pesos en las redes neuronales
- Los vectores de soporte en las máquinas de soporte de vectores
- ...

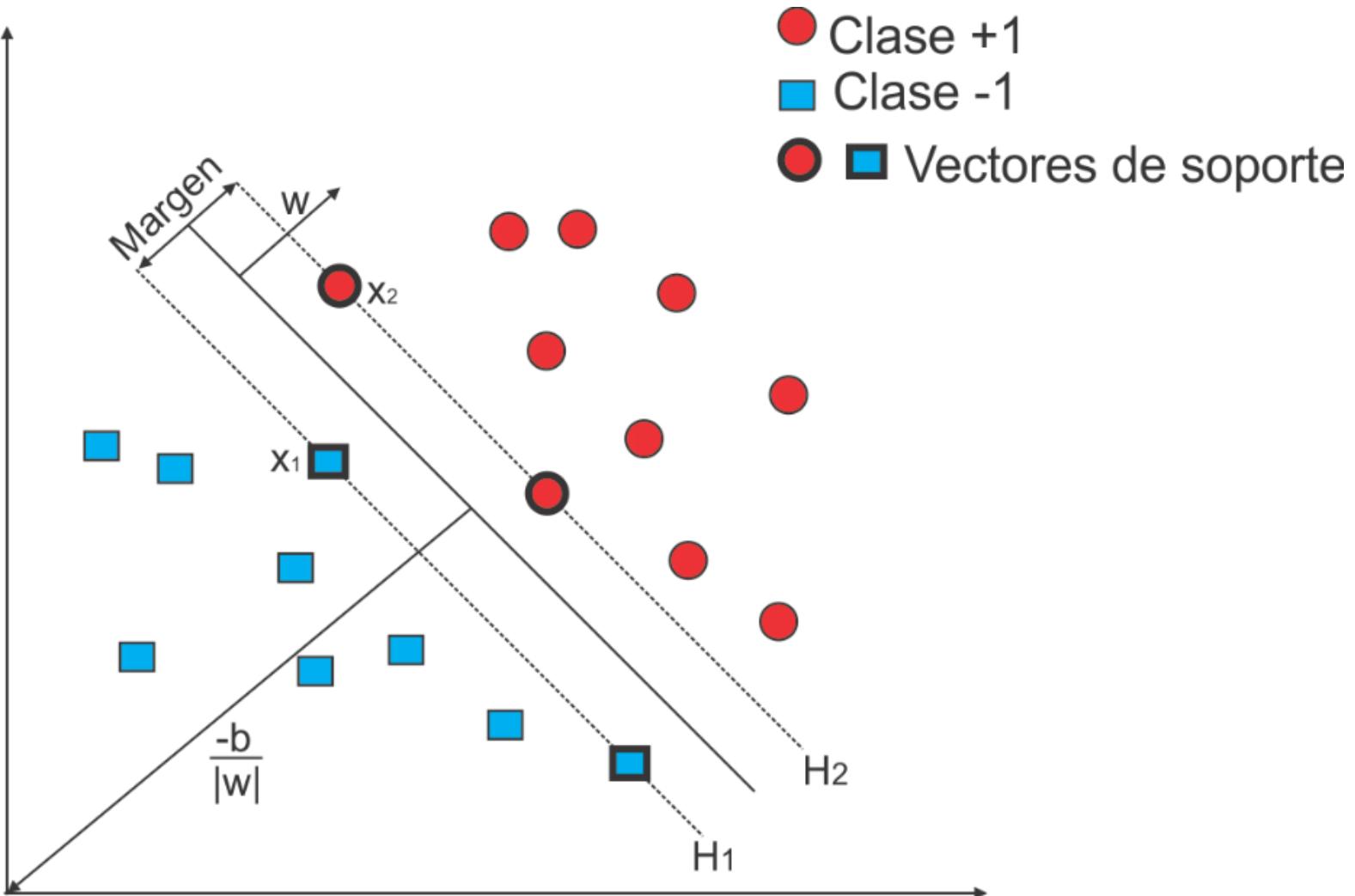
# Algoritmos para aprendizaje automático

Un algoritmo de inducción es una entidad que obtiene un conjunto de datos de entrenamiento y forma un modelo que generaliza la relación entre los atributos de entrada y el atributo de salida.

Existen varios:

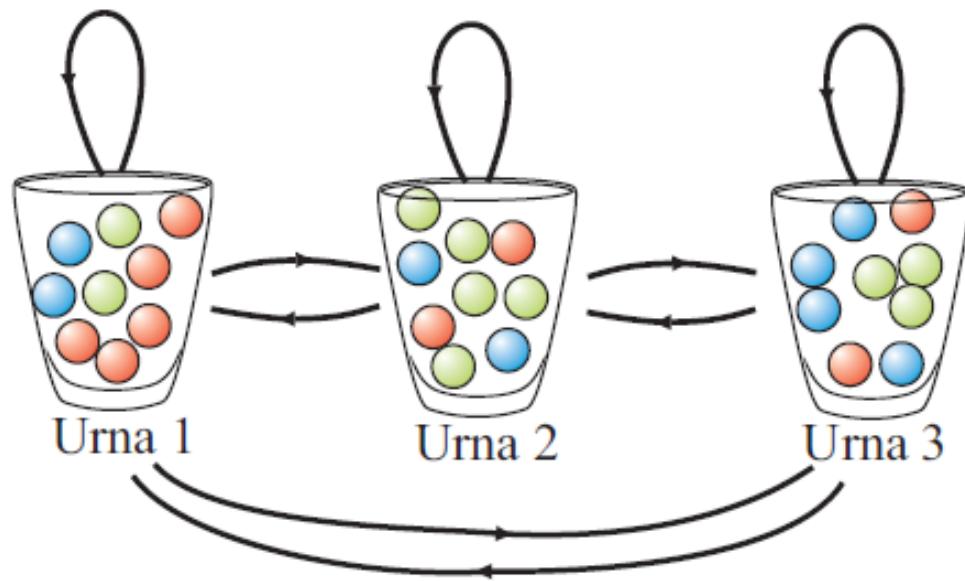
- Bayesiano.
- Redes neuronales.
- Máquina de soporte de vectores.
- Modelos ocultos de markov.
- Vecinos más cercanos.
- ....

# Máquina de soporte de vectores



# Modelos Ocultos de Markov

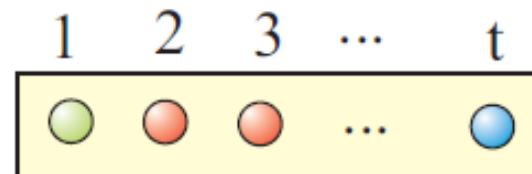
Símbolos emitidos  
por las urnas(estados) = {●, ●, ●}



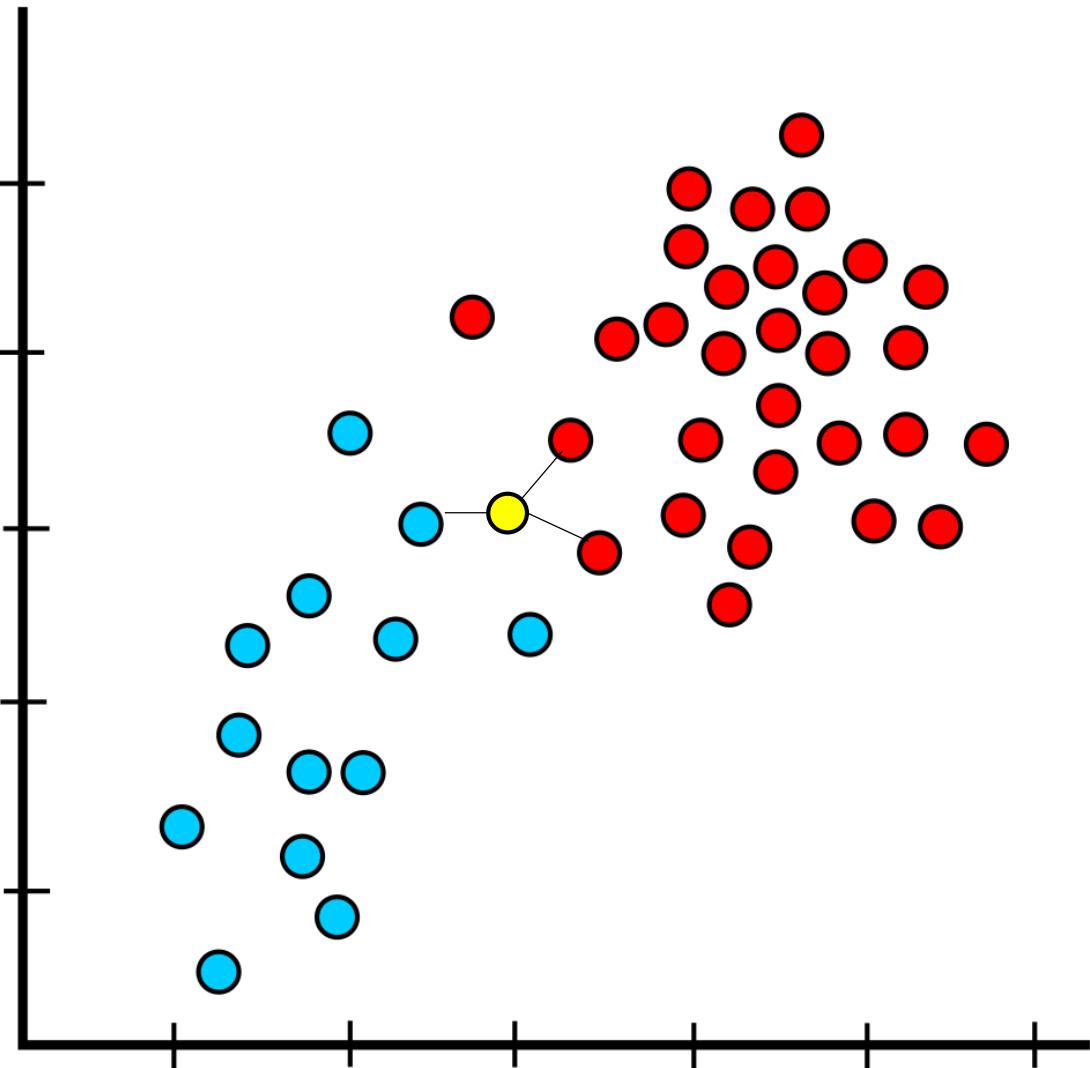
$$A = \begin{bmatrix} 0.5 & 0.3 & 0.2 \\ 0.1 & 0.4 & 0.5 \\ 0.3 & 0.1 & 0.6 \end{bmatrix}$$

$$B = \begin{bmatrix} \bullet_{\text{red}} & \bullet_{\text{green}} & \bullet_{\text{blue}} \\ 5/9 & 2/9 & 2/9 \\ 2/9 & 5/9 & 2/9 \\ 2/9 & 3/9 & 4/9 \end{bmatrix}$$

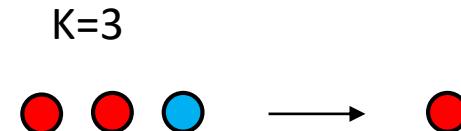
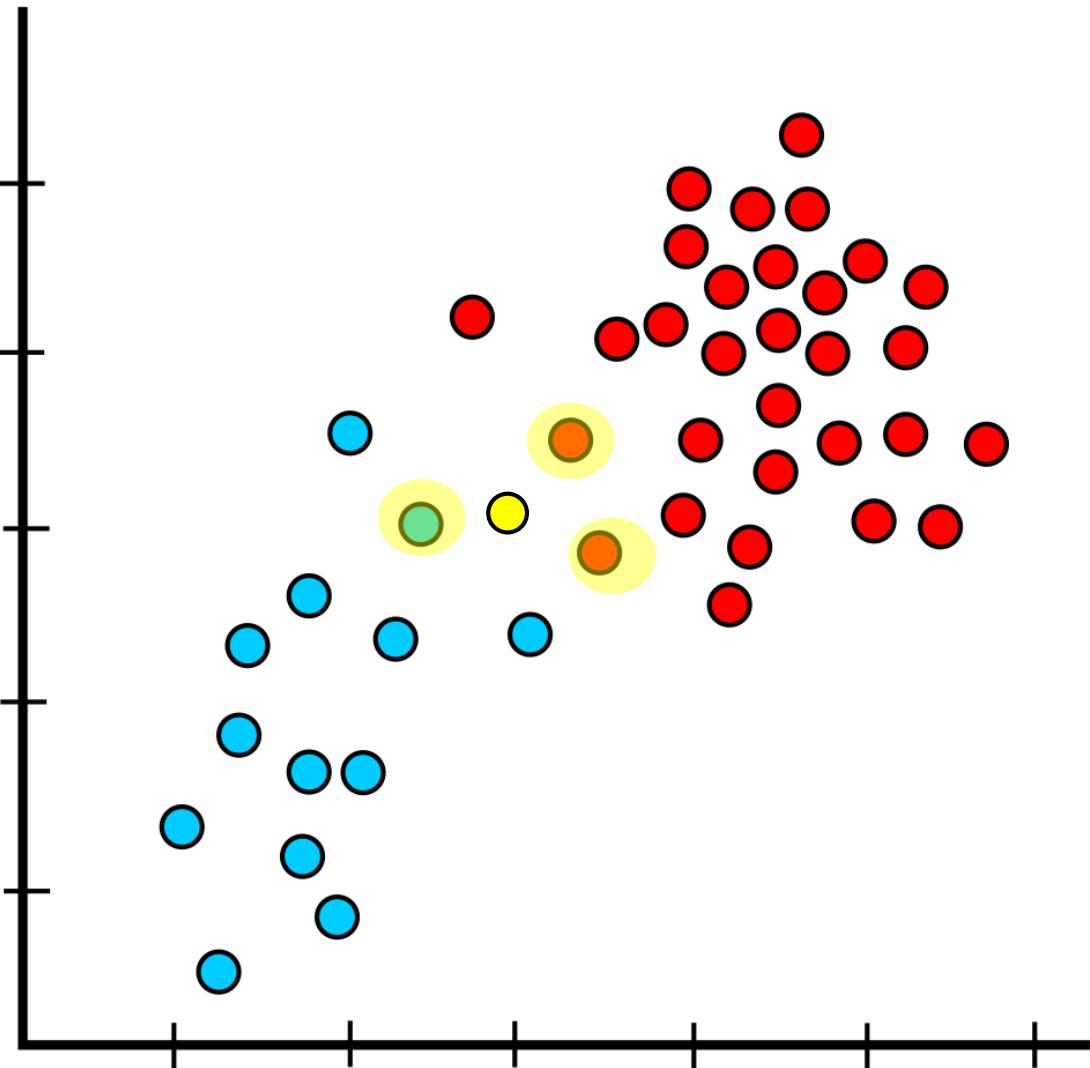
Secuencia observable  
de símbolos



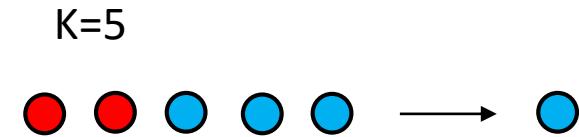
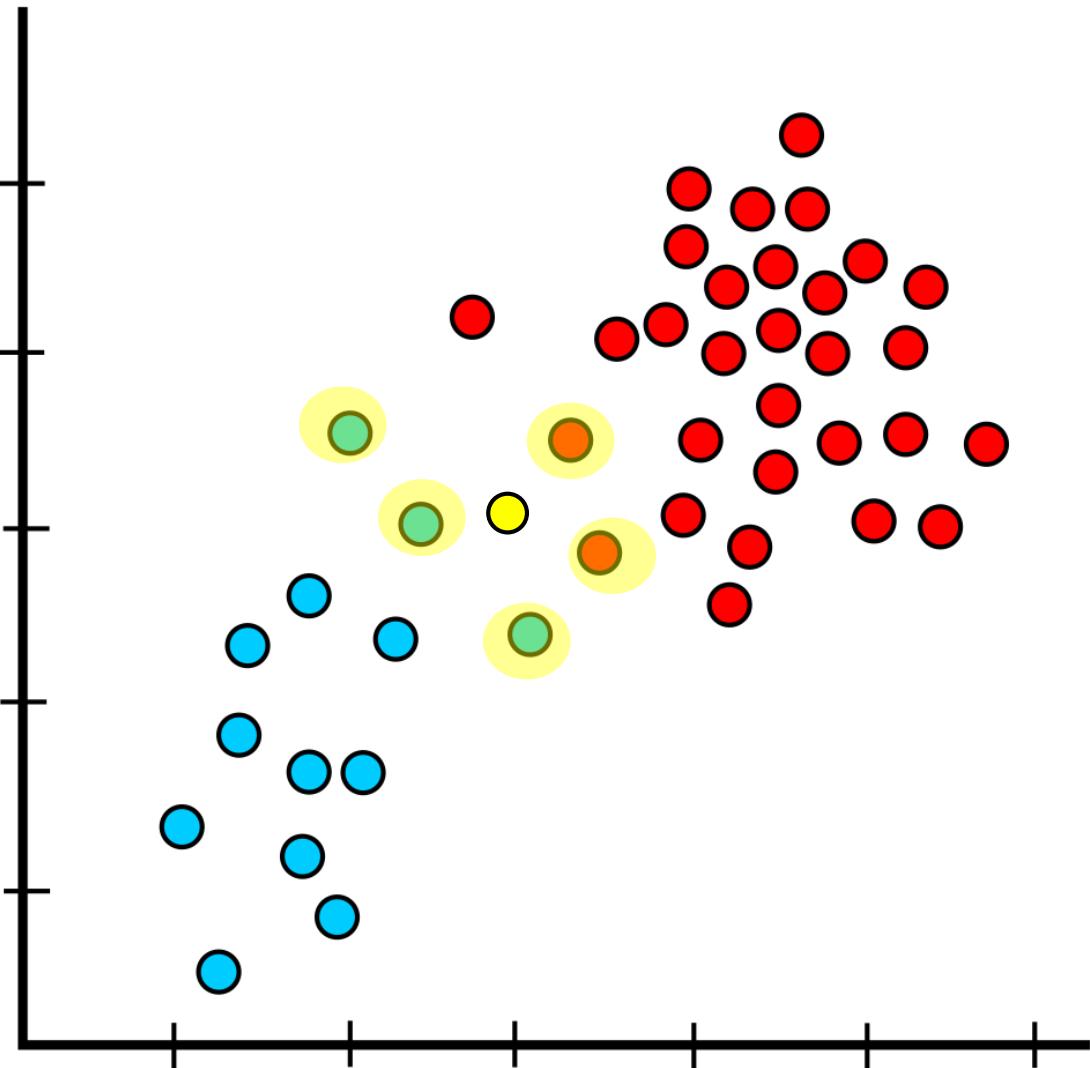
# K Vecinos más cercanos



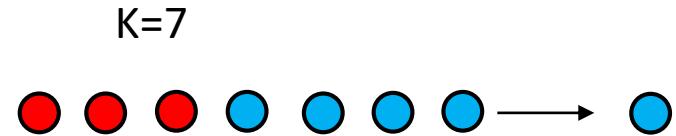
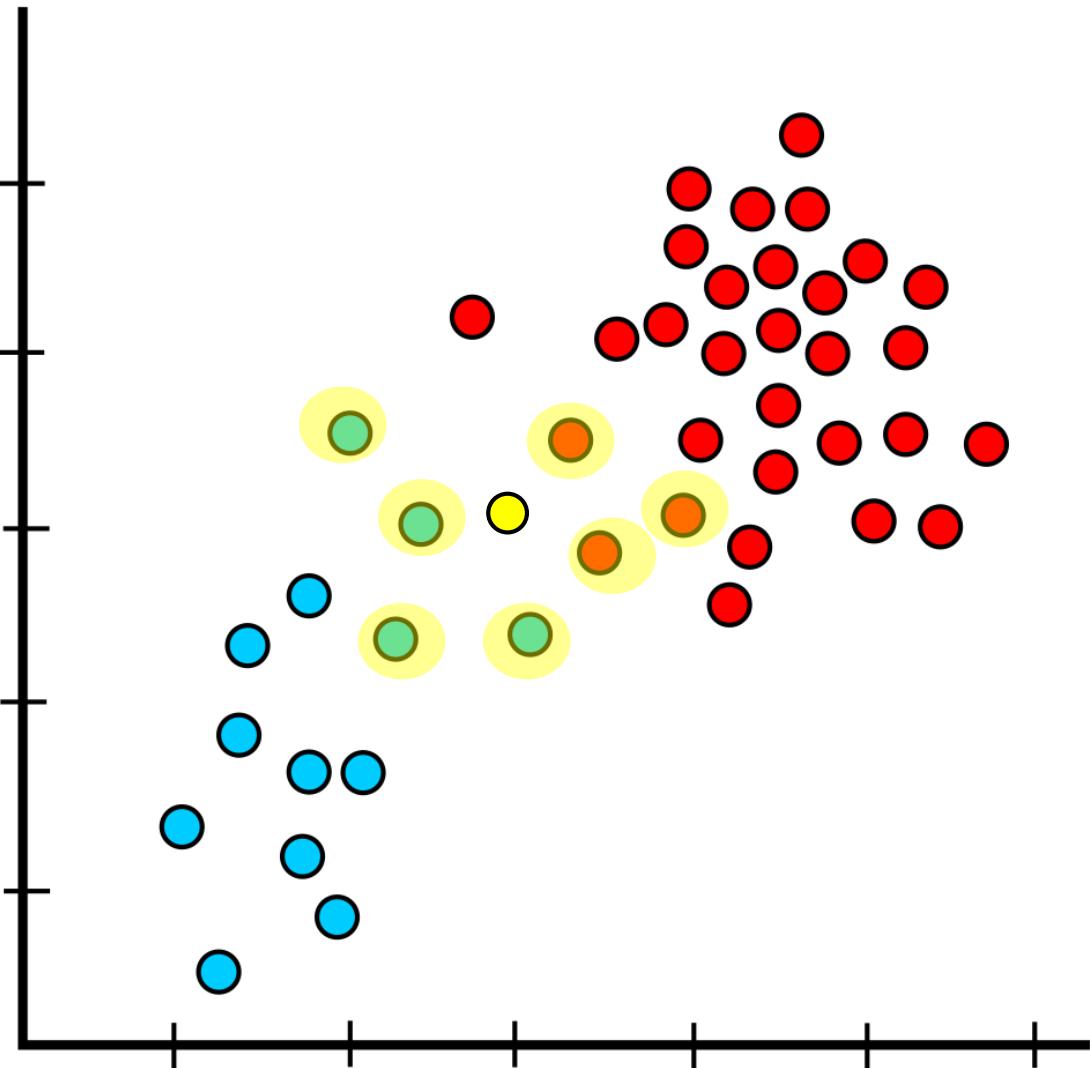
# K Vecinos más cercanos



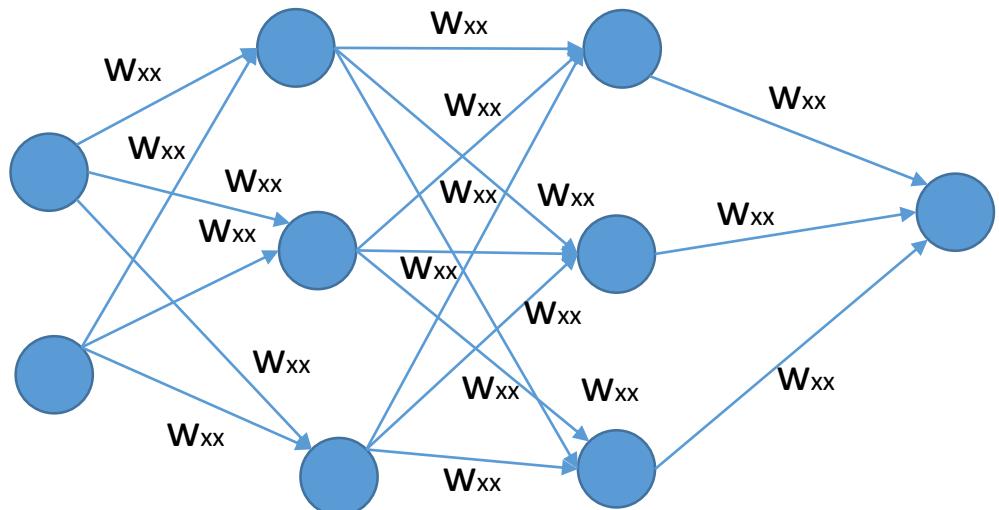
# K Vecinos más cercanos



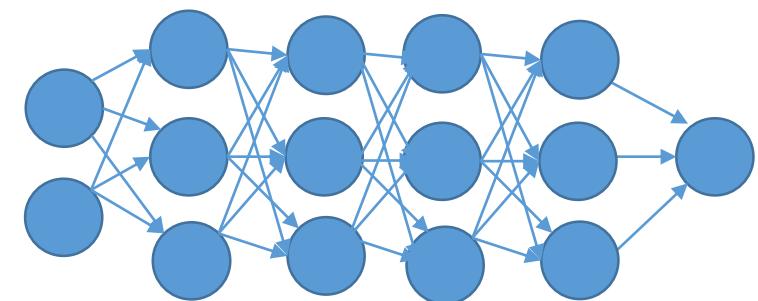
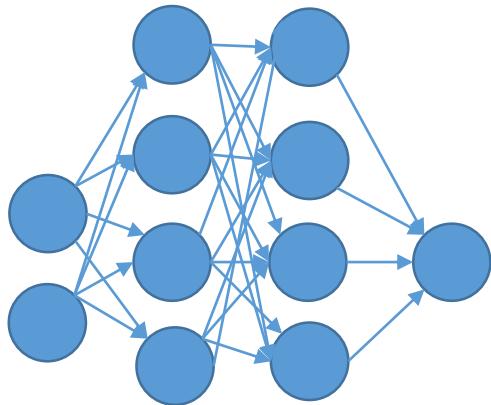
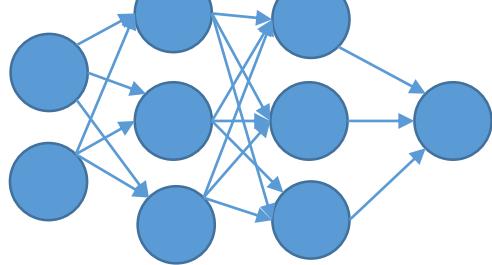
# K Vecinos más cercanos



# Redes neuronales



# Hiperparámetros

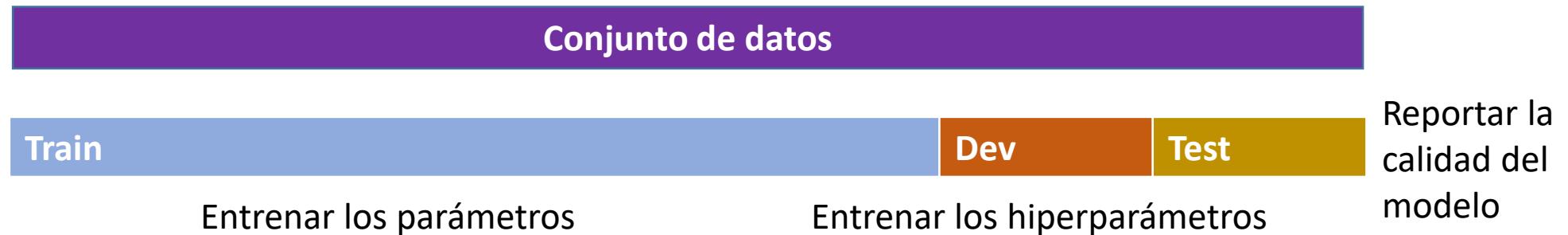


Los hiperparámetros de un modelo se refieren a una configuración del modelo que no se estima a partir de los datos. Se especifican por los científicos.

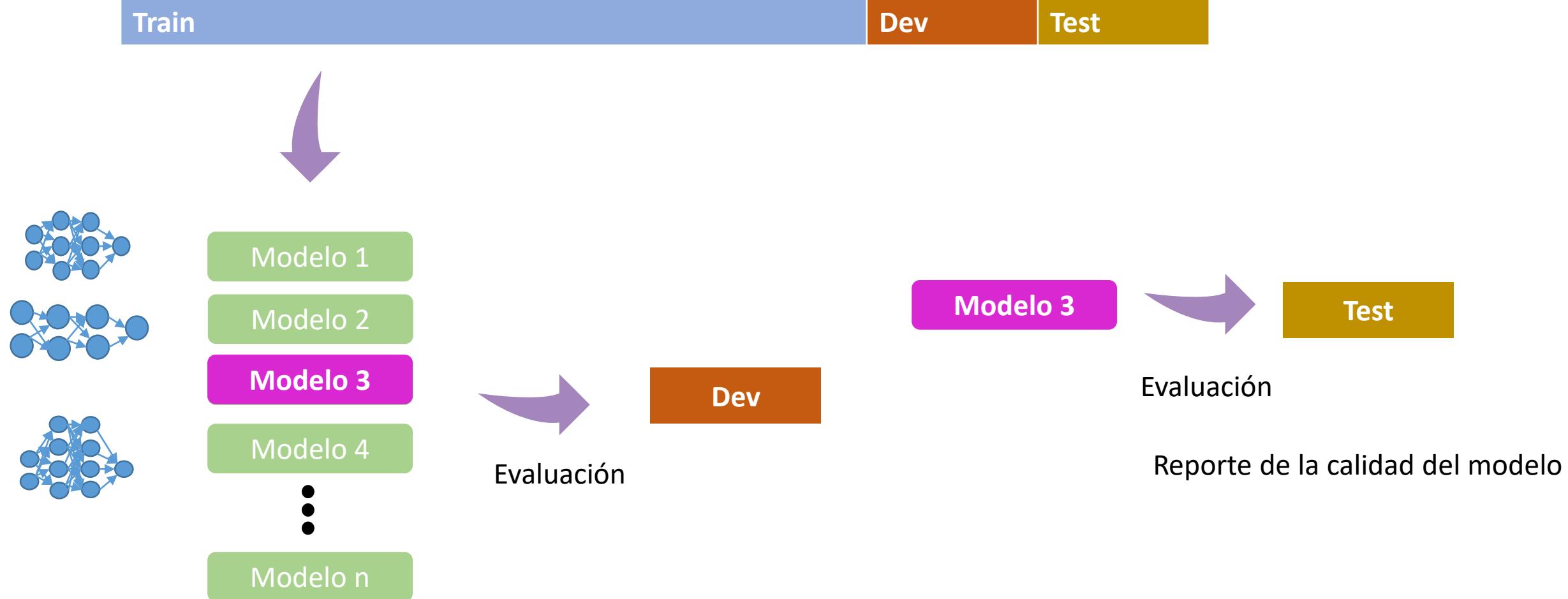
## Ejemplos

- La tasa de aprendizaje en las redes neuronales
- La cantidad de neuronas por capa en las redes neuronales
- La cantidad de capas en las redes neuronales
- El valor de la desviación estándar en las máquinas de vectores de soporte
- ...

# Partición del conjunto de datos



No utilizar **Dev** ni **Test** para la etapa de entrenamiento



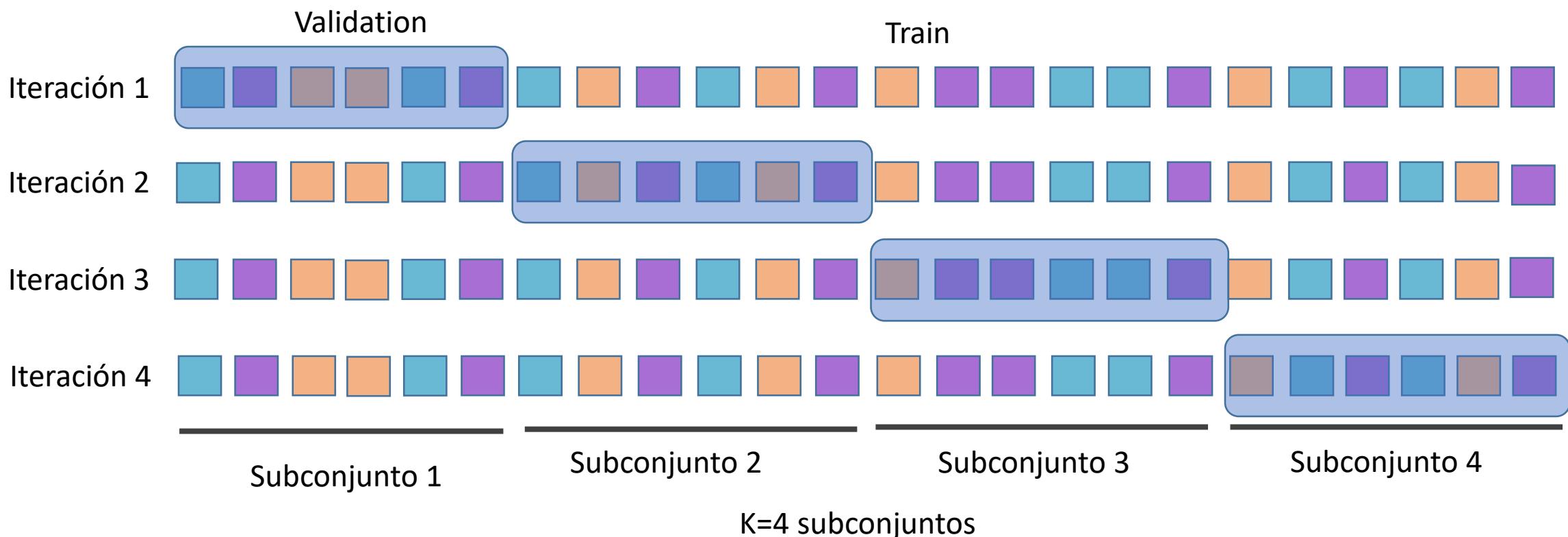
# Validación cruzada

Train

Dev

Test

## Validación cruzada



# Actividad 4

# Rostros con OpenCV y entrenamiento

## Rostros con openCV y entrenamiento

Curso: Aplicaciones del aprendizaje automático con Python

Actividad 4: Entrenamiento para reconocimiento de rostros con openCV

Este código esta basado en el código de Ramiz Raja (<https://www.superdatascience.com/opencv-face-recognition/>)

En esta actividad se realiza reconocimiento de rostros. Se trabaja desde el entrenamiento de un modelo y se hace una predicción.

Objetivos:

1. Trabajar con funciones
2. Detectar rostros (usando código de la actividad 3, detección de objetos)
3. Leer la base de datos para entrenamiento
4. Entrenar un modelo de clasificación
5. Predecir a cuál clase pertenece una imagen de entrada

Dra. Jessica Beltrán Márquez

[www.jessicabeltran.net](http://www.jessicabeltran.net)

### Importar bibliotecas

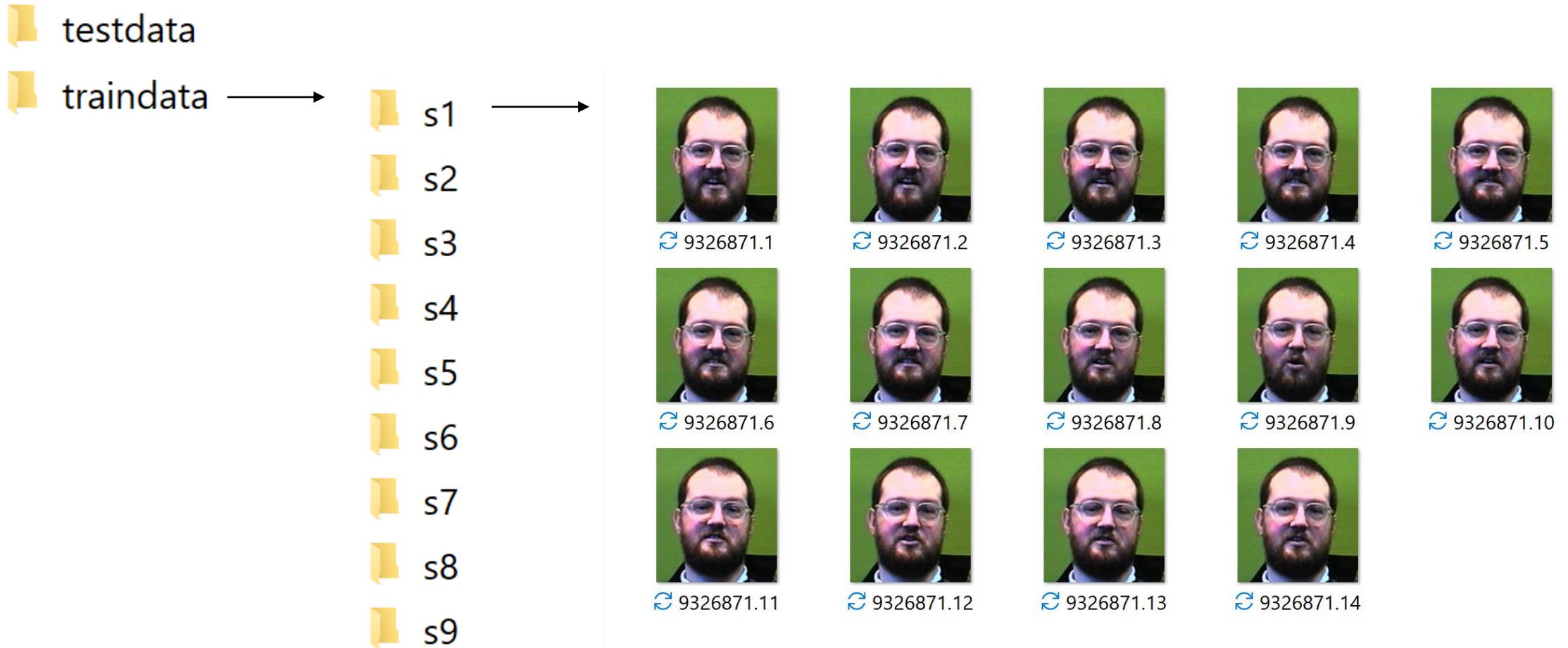
cv2 - OpenCV

os - Biblioteca para la lectura en directorios

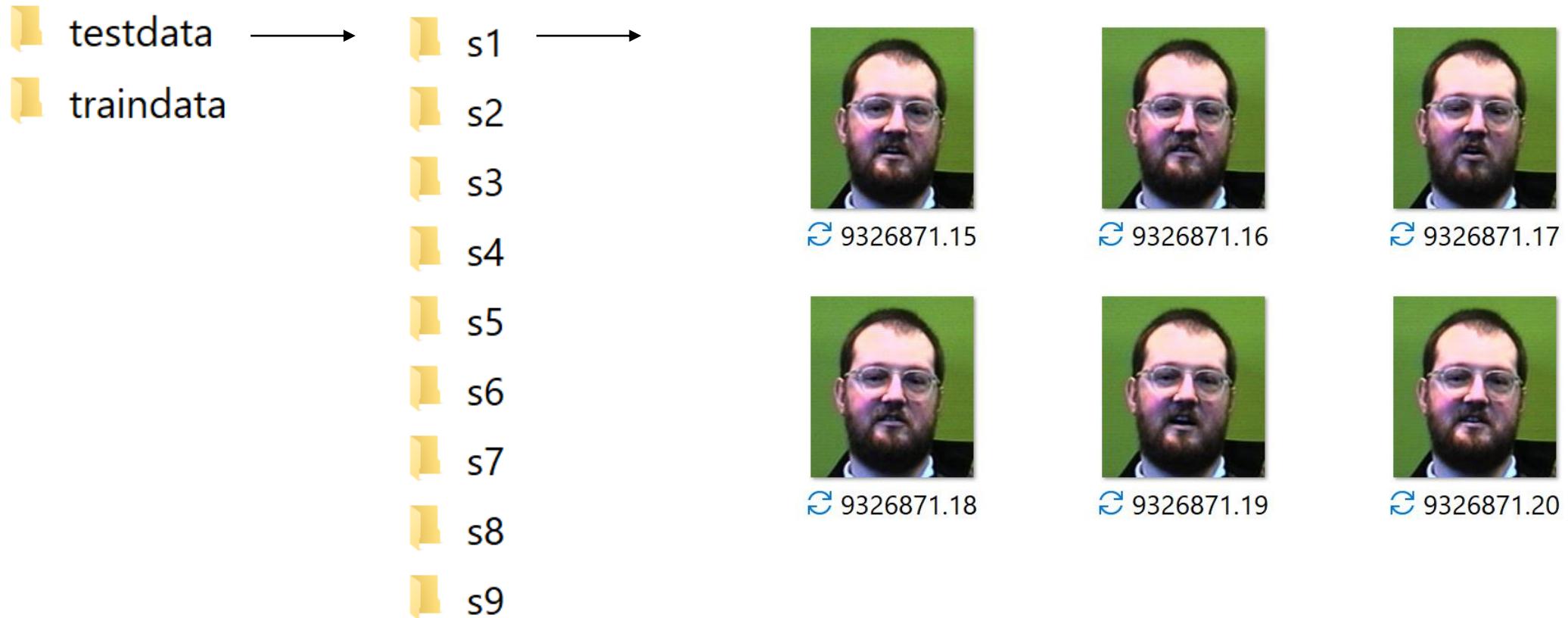
numpy - Biblioteca para calculos numéricos

matplotlib - Biblioteca para graficación

# Rostros con OpenCV y entrenamiento



# Rostros con OpenCV y entrenamiento



# Rostros con OpenCV y entrenamiento

Pepe



Juan



José



Miguel



Toño



Manuel



Adrian



Jesús



Adolfo



# Evaluación

La evaluación es importante para entender la calidad del aprendizaje. Se utilizan métricas para evaluar el desempeño de los algoritmos y parámetros utilizados.

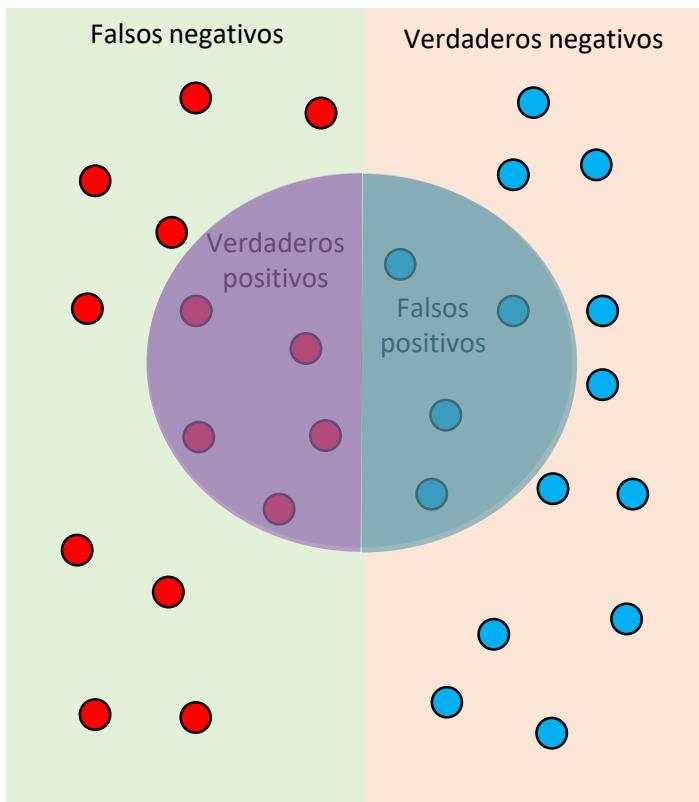
## Matriz de confusión

Herramienta que permite la visualización en forma de matriz del desempeño del algoritmo y parámetros utilizados.

	Iris-setosa	Iris-versicolor	Iris-virginica
Iris-setosa	100.00%	0.0%	0.0%
Iris-versicolor	0.0%	88.7%	6.4%
Iris-virginica	0.0%	11.3%	93.6%

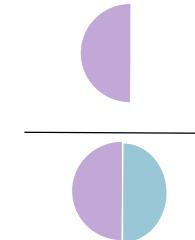
# Evaluación

Elementos relevantes



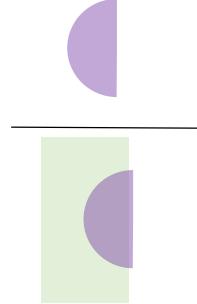
¿Cuántos elementos seleccionados son relevantes?

$$Precision = \frac{TP}{TP + FP}$$



¿Cuántos elementos relevantes son seleccionados?

$$Recall = \frac{TP}{TP + FN}$$



Elementos seleccionados

Verdaderos positivos (*True positives – TP*)

Falsos positivos (*False positives – FP*)

Falsos negativos (*False negatives – FN*)

Verdaderos negativos (*True negatives – TN*)

$$F1score = \frac{2 * precision * recall}{precision + recall}$$

# Actividad 5

# Biblioteca Scikit, algoritmos, partición de BD, evaluaciones

## Biblioteca Scikit, algoritmos, particion de BD, evaluaciones

Curso: Aplicaciones del aprendizaje automático usando Python

Actividad 5: Biblioteca Scikit, algoritmos, particion de BD, evaluaciones

En esta actividad se utiliza la biblioteca libre para machine learning Scikit. Se utiliza la base de datos iris que contiene información de tres tipos de flores distintas. Se hace una partición de la base de datos en datos de entrenamiento, prueba, validación cruzada y se utilizan métricas para probar la eficiencia de los algoritmos.

Parte de este código fue obtenido de Gaël Varoquaux y Jaques Grobler con licencia BSD 3 clause

Objetivos:

1. Trabajar con la biblioteca Scikit
2. Leer y describir la base de datos iris
3. Graficar los elementos de la base de datos
4. Entrenar un modelo de clasificación con regresión logística
5. Hacer predicciones con el modelo de regresión logística
6. Partición de la base de datos
7. Uso de la máquina de soporte vectorial
8. Usar validación cruzada
9. Cambiar métrica para evaluación

Dra. Jessica Beltrán Márquez

[www.jessicabeltran.net](http://www.jessicabeltran.net)



Setosa Iris



Versicolor Iris

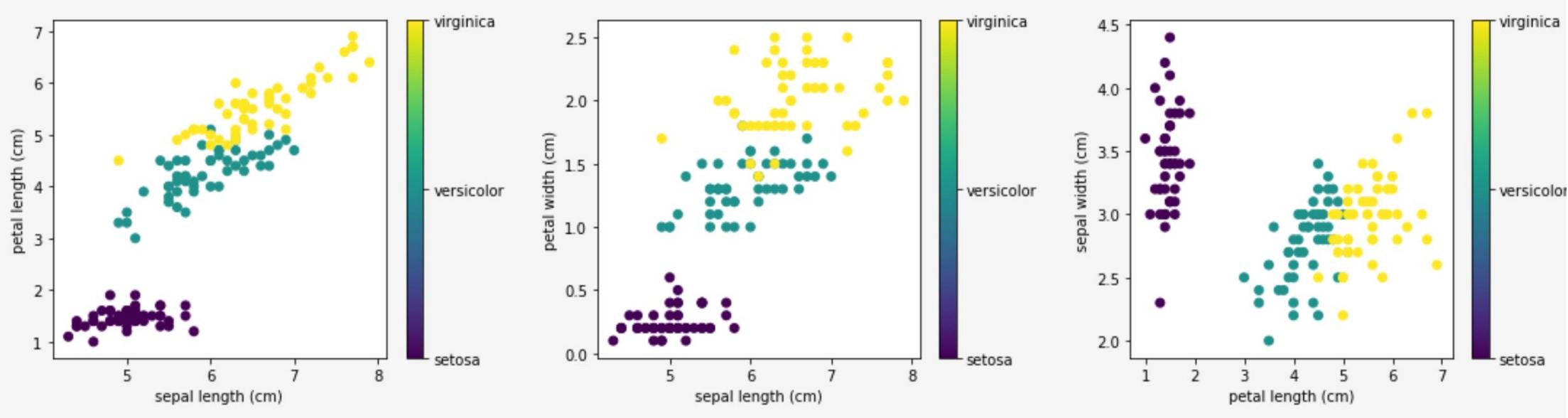


Virginica Iris

# Biblioteca Scikit, algoritmos, partición de BD, evaluaciones

# Biblioteca Scikit, algoritmos, partición de BD, evaluaciones

**Pruébalo:** Modifica el código para desplegar las siguientes gráficas.



# Los índices de las características que se grafican

```
x_axis_index = 2  
y_axis_index = 1
```

# Biblioteca Scikit, algoritmos, partición de BD, evaluaciones

## Preparación de los datos

En este ejemplo se utilizaran únicamente las primeras 2 características para hacer la clasificación.

```
In [15]: X = iris.data[:, :2] # Se toman las dos primeras características  
Y = iris.target  
print("Solo las dos primeras características")  
print(X)  
print("Las etiquetas")  
print(Y)
```

```
Solo las dos primeras características  
[[5.1 3.5]  
[4.9 3. ]  
[4.7 3.2]  
[4.6 3.1]  
[5. 3.6]  
[5.4 3.9]  
[4.6 3.4]  
[5. 3.4]  
[4.4 2.9]  
[4.9 3.1]  
[5.4 3.7]  
[4.8 3.4]  
[4.8 3. ]  
[4.3 3. ]  
[5.8 4. ]  
[5.7 4.4]  
[5.4 3.9]  
[5.1 3.5]  
[5. 3.2]
```

```
-----Nombre de las características-----  
['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']  
-----Datos-----  
[[5.1 3.5 1.4 0.2]  
[4.9 3. 1.4 0.2]  
[4.7 3.2 1.3 0.2]  
[4.6 3.1 1.5 0.2]  
[5. 3.6 1.4 0.2]  
[5.4 3.9 1.7 0.4]  
[4.6 3.4 1.4 0.3]  
[5. 3.4 1.5 0.2]  
[4.4 2.9 1.4 0.2]  
[4.9 3.1 1.5 0.1]]
```

# Biblioteca Scikit, algoritmos, partición de BD, evaluaciones

## Partición del conjunto de datos

Se utiliza la función `train_test_split` de la biblioteca scikit para hacer la partición.

¿Qué pasa si cambias el valor de `test_size`?

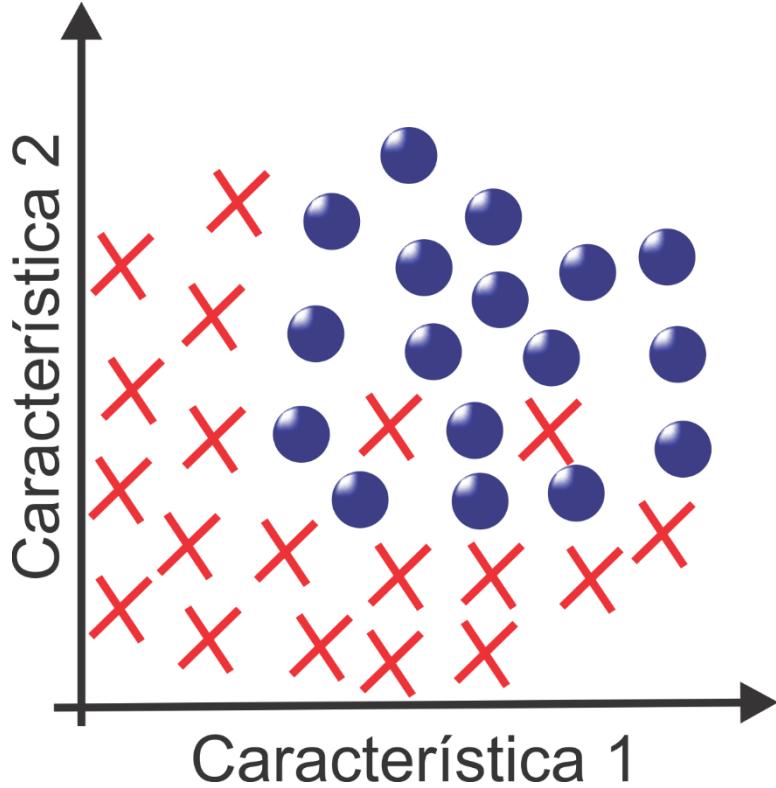
```
In [33]: X_train, X_test, y_train, y_test = train_test_split(  
...      iris.data, iris.target, test_size=0.3, random_state=0)  
print('Forma de la BD de entrenamiento',X_train.shape, y_train.shape)  
print('Forma de la BD de prueba',X_test.shape, y_test.shape)
```

Forma de la BD de entrenamiento (105, 4) (105,)

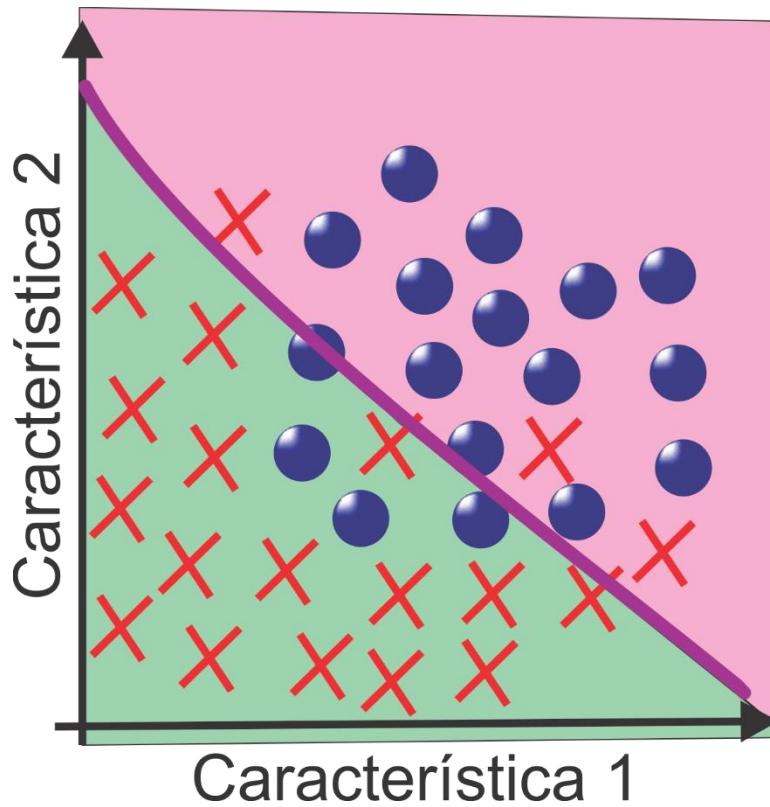
Forma de la BD de prueba (45, 4) (45,)

# Más conceptos

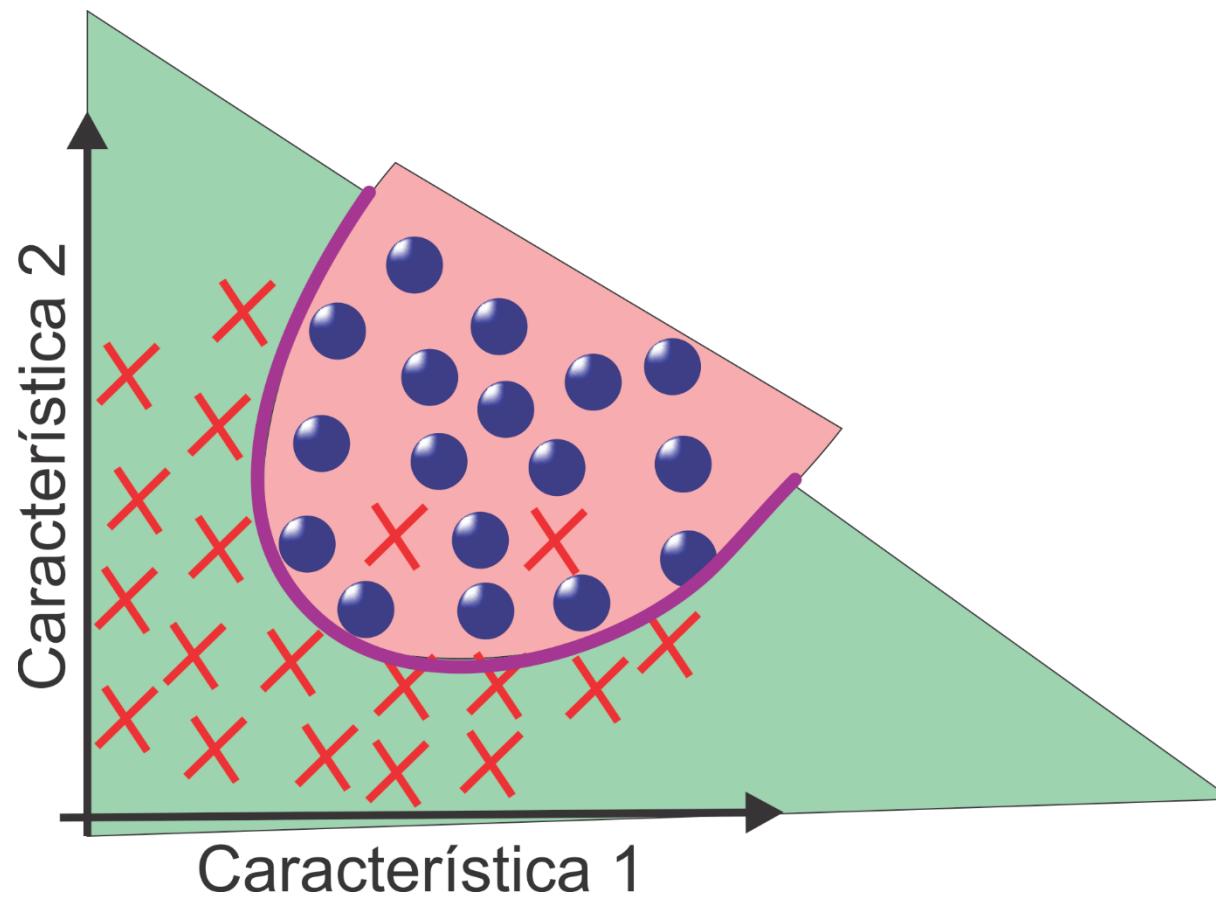
# Ajuste de los datos



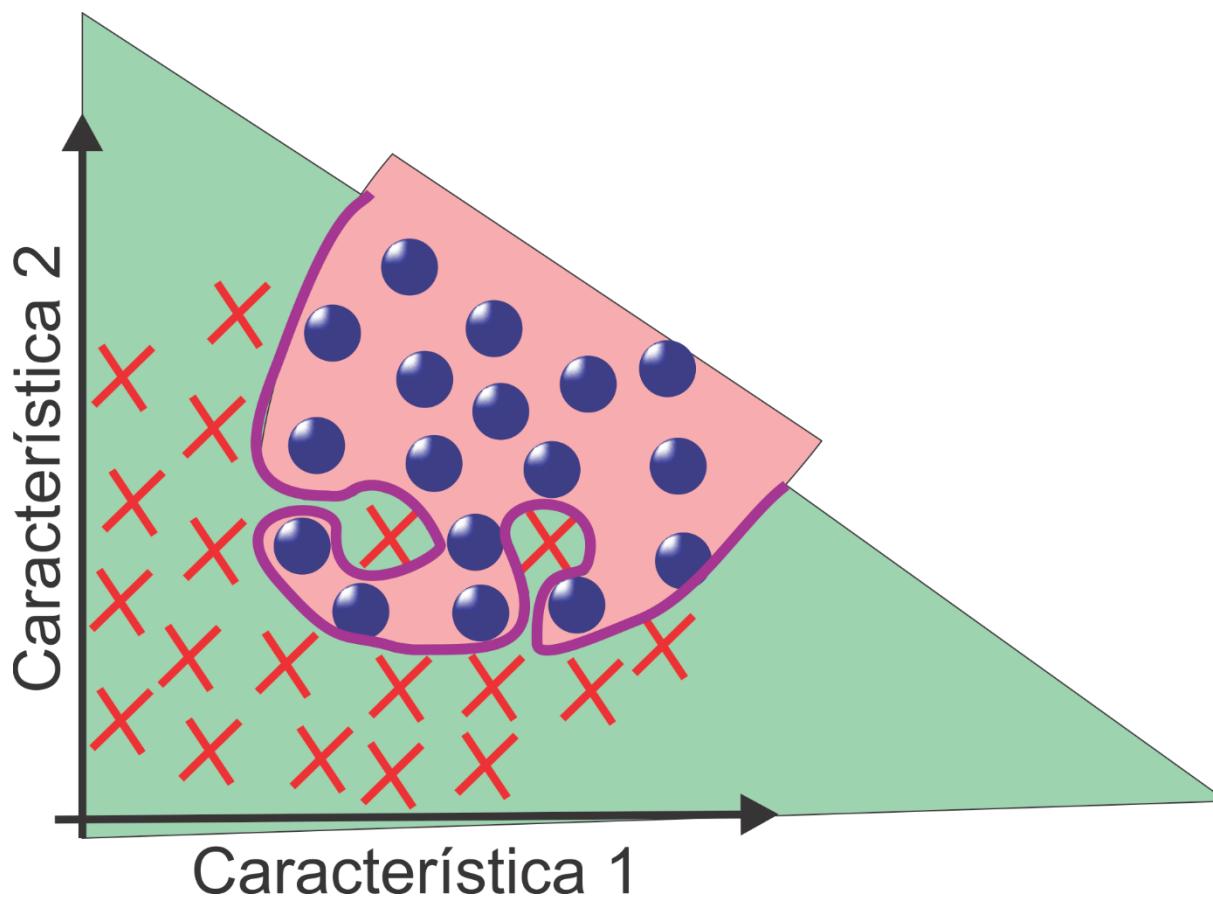
# Datos no ajustados



# “Buen ajuste”

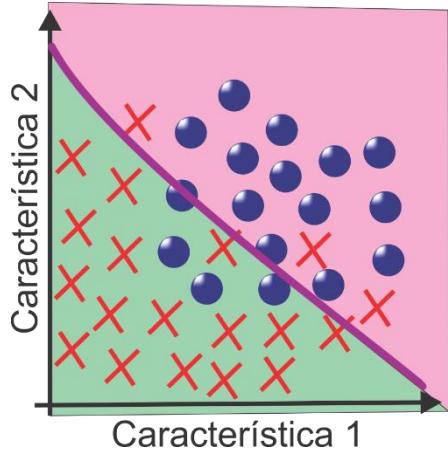


# Datos sobre ajustados

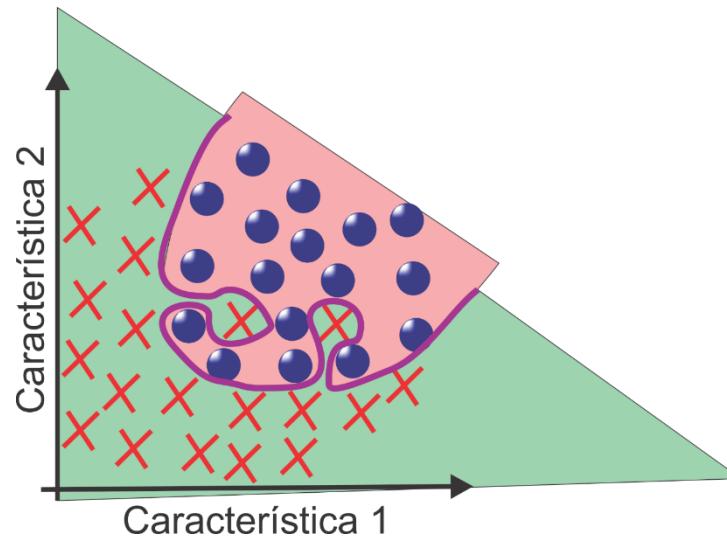


Es el efecto de sobreentrenar un algoritmo de aprendizaje con unos ciertos datos para los que se conoce el resultado deseado.

# Bias y Variance - Bias



# Bias y Variance - Variance



# **High bias**

Probar cambiar algoritmo.

Probar cambiar parámetros y/o hiperparámetros.

Probar con entrenar más tiempo.

Intentar usar otras características.

# **High variance**

Probar con más datos.

Probar con regularización.

Probar otro algoritmo y/o parámetros.

Intentar usar otras características.

# Bias y Variance

## F1 Score

<b>Entrenamiento</b>	<b>99%</b>	<b>85%</b>	<b>85%</b>	<b>99.5%</b>
Desarrollo	89%	84%	70%	99%

High variance

High bias

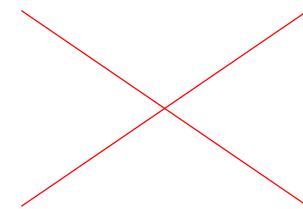
High bias  
High variance

low bias  
low variance

# Datos para entrenamiento y prueba

## Entrenamiento

Imágenes con alta definición

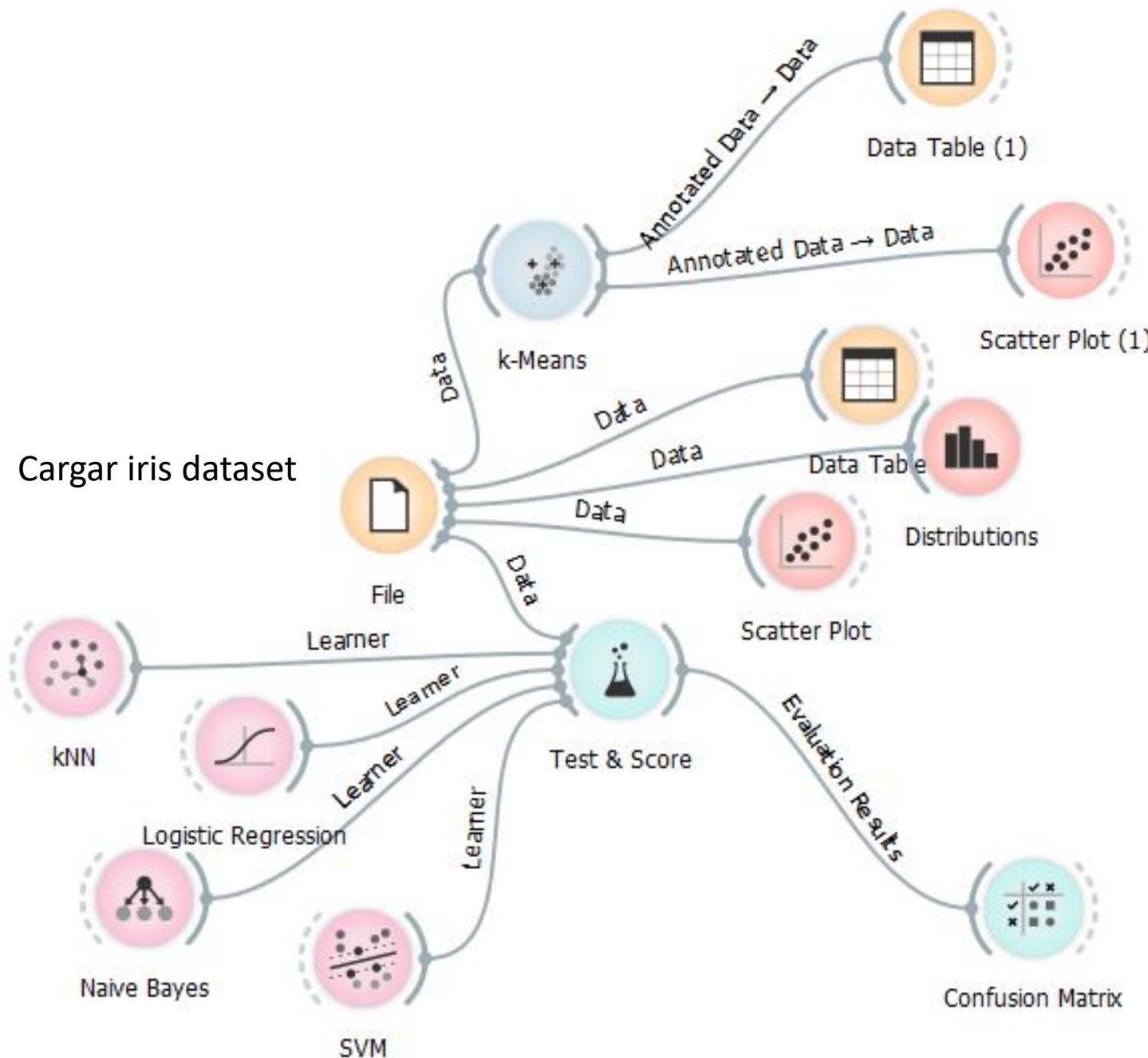


## Desarrollo/Prueba

Imágenes tomadas con el teléfono celular.

**Deben de provenir de una distribución similar**

# Comparación del desempeño de varios algoritmos de clasificación



# Actividad 6



# Redes neuronales con Tensorflow

Curso: Aplicaciones del aprendizaje automático usando Python

Actividad 6: Biblioteca Tensorflow

En esta actividad se utiliza la biblioteca libre para machine learning Scikit. Se utiliza la base de datos iris que contiene información de tres tipos de flores distintas. Se hace una partición de la base de datos en datos de entrenamiento, prueba, validación cruzada y se utilizan métricas para probar la eficiencia de los algoritmos.

Este código fue obtenido del tutorial de Tensorflow "Basic classification" [https://www.tensorflow.org/tutorials/keras/basic\\_classification](https://www.tensorflow.org/tutorials/keras/basic_classification) con licencia <https://creativecommons.org/licenses/by/4.0/>

Objetivos:

1. Trabajar con la biblioteca Tensorflow y Keras
2. Explorar los datos

# Actividad 7



# Text classification with movie reviews

[View on TensorFlow.org](#)[Run in Google Colab](#)[View source on GitHub](#)

En este cuaderno se clasifican las evaluaciones de usuarios a películas como *positivas* o *negativas*. Este es un ejemplo de una clasificación *binaria* o de dos clases, un problema ampliamente aplicable en el aprendizaje automático.

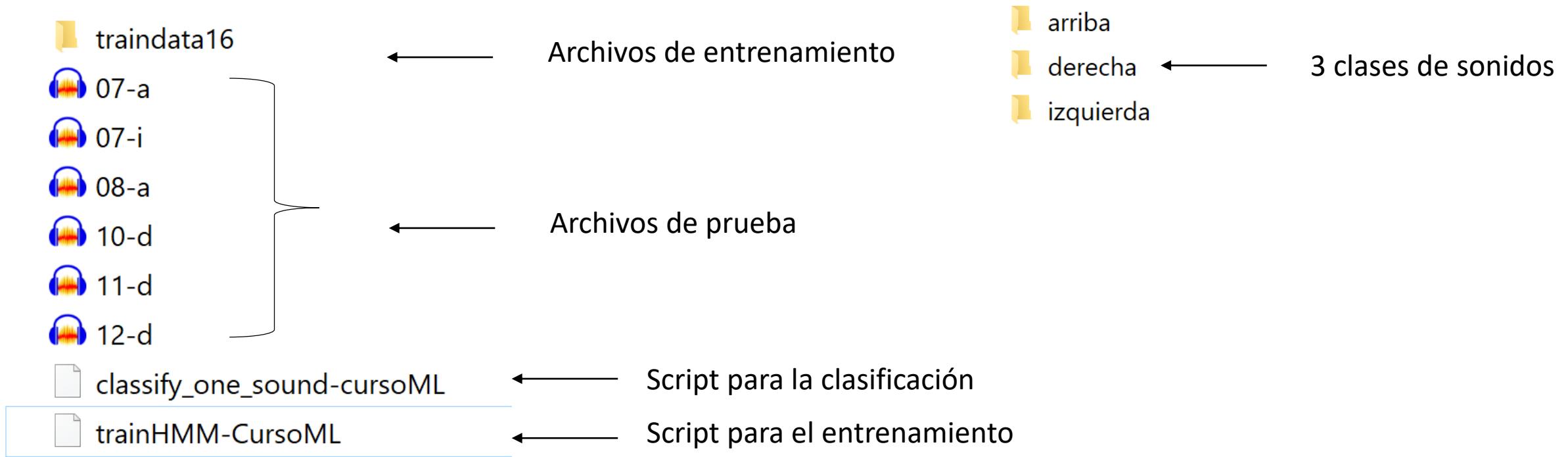
Se usará el conjunto de datos de [IMDB](#) que contiene el texto de 50,000 evaluaciones de películas de la Internet Movie DataBase (IMBD). Este conjunto se divide en 25,000 evaluaciones para el entrenamiento y 25,000 para la prueba. Los conjuntos de entrenamiento y prueba están *balanceados*, lo que significa que ambos contienen la misma cantidad de evaluaciones positivas y negativas.

Este cuaderno usa `tf.keras`, una API de alto nivel para entrenar modelos en TensorFlow. Para un tutorial más avanzado en clasificación de texto usando `tf.keras`, ve a [MLCC Text Classification Guide](#).

```
In [1]: # keras datasets IMDB is broken in 1.12 and 1.14 but nn 1.16.2
```

# Actividad 8

## Clasificación de sonidos con Modelos Ocultos de Markov



Spyder (Python 3.7)

File Edit Search Source Run Debug Consoles Projects Tools View Help

Editor - C:\Users\jessi\OneD trainHMM-CursoML.py

1# -\*- coding: utf-8 -\*-  
2  
3Created on Fri Nov 1 10:30:28 2019  
4  
5@author: jessi  
6  
7  
8import os  
9import argparse  
10  
11import numpy as np  
12from scipy.io import wavfile  
13from hmmlearn import hmm  
14from python\_speech\_features import mfcc  
15from sklearn.externals import joblib  
16  
17# Function to parse input arguments  
18def build\_arg\_parser():  
19 parser = argparse.ArgumentParser(description='Trains the HMM classifier')  
20 parser.add\_argument("--input-folder", dest="input\_folder", required=True,  
21 help="Input folder containing the audio files in subfolders")  
22 return parser  
23  
24# Class to handle all HMM related processing  
25class HMMTrainer(object):  
26 def \_\_init\_\_(self, model\_name='GaussianHMM', n\_components=4, cov\_type='diag', n\_iter=1000):  
27 self.model\_name = model\_name  
28 self.n\_components = n\_components  
29 self.cov\_type = cov\_type  
30 self.n\_iter = n\_iter  
31 self.models = []  
32  
33 if self.model\_name == 'GaussianHMM':  
34 self.model = hmm.GaussianHMM(n\_components=self.n\_components,  
35 covariance\_type=self.cov\_type, n\_iter=self.n\_iter)  
36 else:  
37 raise TypeError('Invalid model type')  
38  
39# X is a 2D numpy array where each row is 13D  
40def train(self, X):  
41 np.seterr(all='ignore')  
42 self.models.append(self.model.fit(X))  
43  
44# Run the model on input data  
45def get\_score(self, input\_data):  
46 return self.model.score(input\_data)

Run F5  
Run cell Ctrl+Return  
Run cell and advance Shift+Return  
Re-run last cell Alt+Return  
Run selection or current line F9  
Re-run last script F6  
Configuration per file... Ctrl+F6  
Profile F10

OneDrive\TRABAJO\MATERIAS\CURSO ML - PYTHON\Taller Aplicaciones del aprendizaje automatico\Actividad 8

Aplicaciones del aprendizaje automatico\Actividad 8\trainHMM-Cur... Help

Source Console Object

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our [tutorial](#)

Variable explorer File explorer Help Profiler

IPython console

Console 1/A

Python 3.7.4 (default, Aug 9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)]  
Type "copyright", "credits" or "license" for more information.

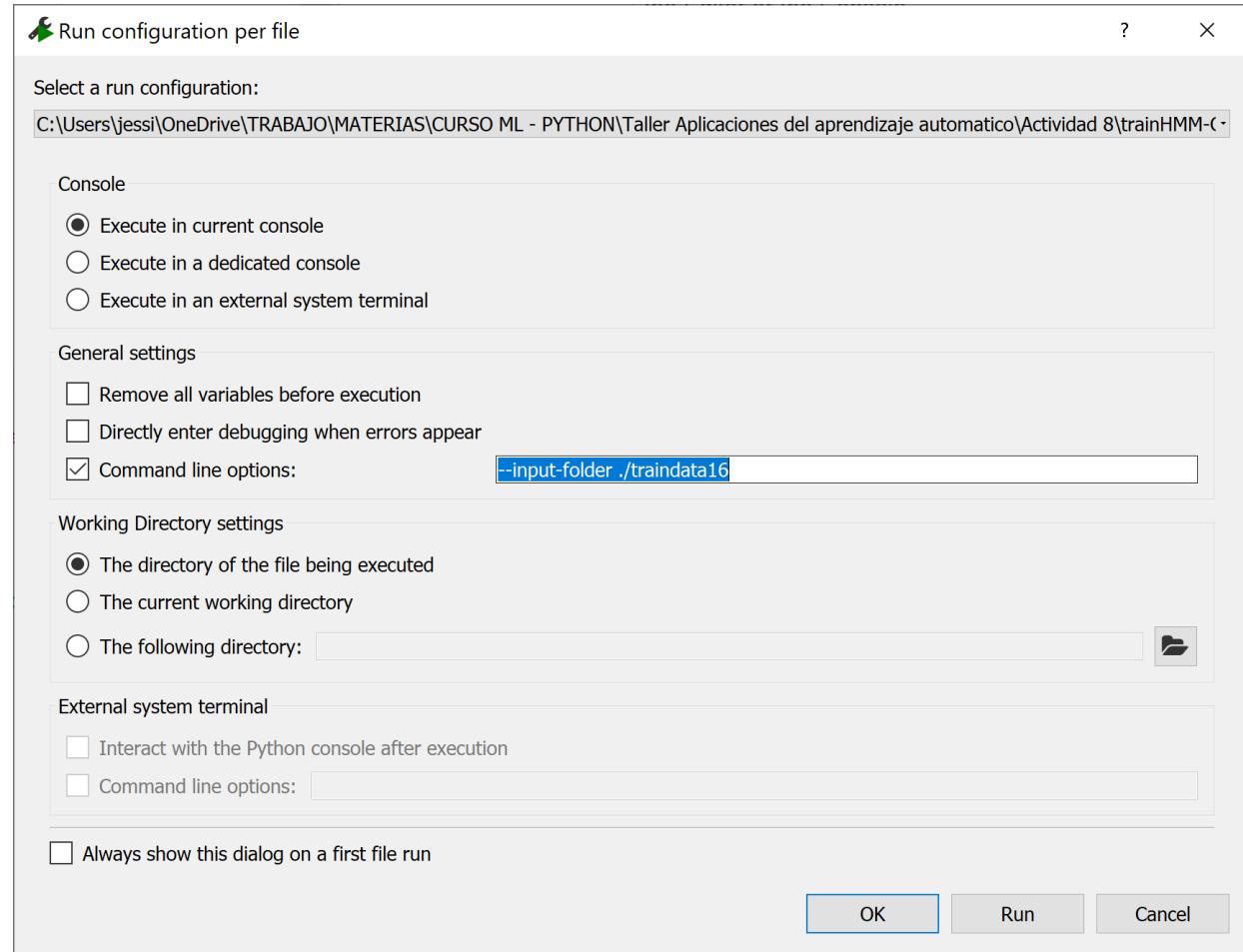
IPython 7.8.0 -- An enhanced Interactive Python.

In [1]:

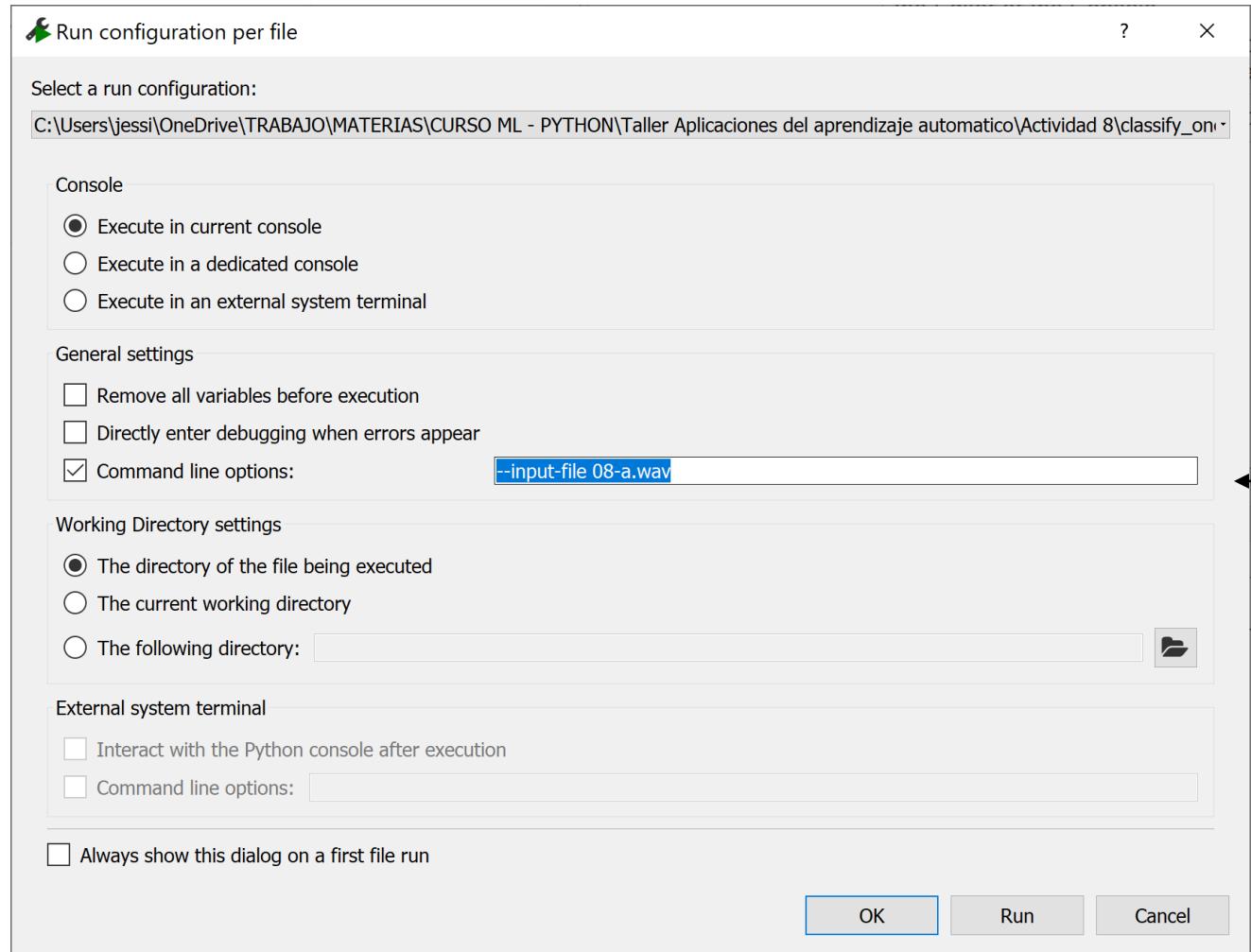
IPython console History log

Aplicaciones del aprendizaje automático con Python | Dra. Jessica Beltrán  
09-Oct-19

186



Ruta con archivos de entrenamiento



Ruta con archivos de prueba

## Spyder (Python 3.7)

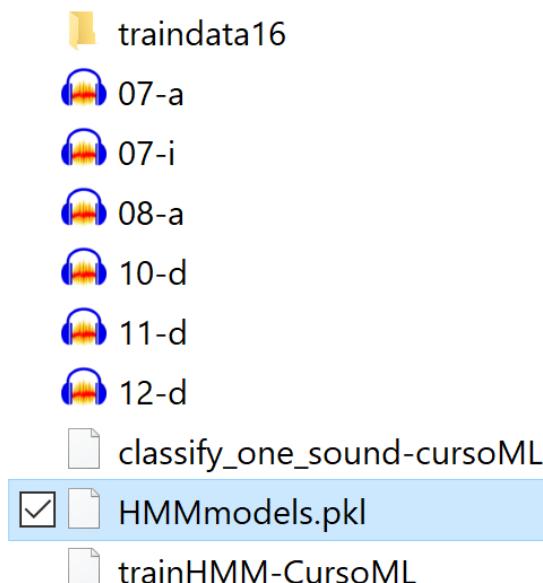
File Edit Search Source Run Debug Consoles Projects Too

Editor - C:\Users\jess... Run file (F5) RABAJO\MATERIAS\CURSO ML - PY

trainHMM-CursoML.py classify\_one\_sound-cursoML.py

```
1# -*- coding: utf-8 -*-

```



Nota1: Revisar si hay bibliotecas que se requieren instalar (hmmlearn, python\_speech\_features)  
Nota2: Abrir Spyder dentro del ambiente con las bibliotecas instaladas (py37).

← Se genera un modelo



Editor - C:\Users\jessi\OneDrive\TRABAJO\MATERIAS\CURSO ML - PYTHON\Taller Aplicaciones del aprendizaje automático\Actividad 8\classify\_one\_sound-cursoML.py

trainHMM-CursoML.py classify\_one\_sound-cursoML.py

```
15     return parser
16
17 # Class to handle all HMM related processing
18 class HMMTrainer(object):
19     def __init__(self, n_states=10, n_features=13, n_iter=1000, tol=0.001):
20         self.n_states = n_states
21         self.n_features = n_features
22         self.n_iter = n_iter
23         self.tol = tol
24
25         self.transitions = np.zeros((n_states, n_states))
26         self.emissions = np.zeros((n_states, n_features))
27         self.pi = np.zeros(n_states)
28
29         self._init_transitions()
30         self._init_emissions()
31
32     def _init_transitions(self):
33         self.transitions = np.ones((self.n_states, self.n_states)) / self.n_states
```

IPython console

Console 1/A

```
wavfilewarning)
X.shape = (541, 13)
X.shape = (631, 13)
C:\Users\jessi\Anaconda3\envs\py37\lib\site-packages\scipy\io\wavfile.py:273: WavFileWarning: Chunk (non-data) not understood, skipping it.
  WavFileWarning)
C:\Users\jessi\Anaconda3\envs\py37\lib\site-packages\scipy\io\wavfile.py:273: WavFileWarning: Chunk (non-data) not understood, skipping it.
  WavFileWarning)
X.shape = (627, 13)

In [2]: runfile('C:/Users/jessi/OneDrive/TRABAJO/MATERIAS/CURSO ML - PYTHON/Taller Aplicaciones del aprendizaje automático/Actividad 8/classify_one_sound-cursoML.py', args='--input-file 08-a.wav', wdir='C:/Users/jessi/OneDrive/TRABAJO/MATERIAS/CURSO ML - PYTHON/Taller Aplicaciones del aprendizaje automático/Actividad 8')
Predicted: traindata16\arriba
C:\Users\jessi\Anaconda3\envs\py37\lib\site-packages\scipy\io\wavfile.py:273: WavFileWarning: Chunk (non-data) not understood, skipping it.
  WavFileWarning)

In [3]:
```

Clase predicha

# Clasificación de sonidos con Modelos Ocultos de Markov

The screenshot shows the Spyder Python 3.7 IDE interface. On the left, the code editor displays a script named `trainHMM-CursoML.py`. A tooltip is open over the `__init__` method of the `HMMTrainer` class, providing documentation for the parameters: `n_components=4, cov_type='diag', n_iter=1000`. The code itself defines the `HMMTrainer` class with its constructor and methods like `train` and `get_score`. It also loads pre-trained HMM models from a file and processes audio files using MFCC features. On the right, the IPython console window shows the Python version and license information, and the In [1]: prompt.

```
Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
Run F5
Run cell Ctrl+Return
Run cell and advance Shift+Return
Re-run last cell Alt+Return
Run selection or current line F9
Re-run last script F6
Configuration per file... Ctrl+F6
Profile F10

OneDrive\TRABAJO\MATERIAS\CURSO ML - PYTHON\Taller Aplicaciones del aprendizaje automatico\Actividad 8
Aplicaciones del aprendizaje automatico\Actividad 8\classify_one_s... Help
SourceConsole Object

15     return parse
16
17 # Class to handle
18 class HMMTrainer:
19     def __init__(self, n_components=4, cov_type='diag', n_iter=1000):
20         self.model = hmm.GaussianHMM(n_components=n_components,
21                                       covariance_type=cov_type, n_iter=n_iter)
22         self.n_iter = n_iter
23         self.models = []
24
25         if self.model_name == 'GaussianHMM':
26             self.model = hmm.GaussianHMM(n_components=self.n_components,
27                                           covariance_type=self.cov_type, n_iter=self.n_iter)
28         else:
29             raise TypeError('Invalid model type')
30
31     # X is a 2D numpy array where each row is 13D
32     def train(self, X):
33         np.seterr(all='ignore')
34         self.models.append(self.model.fit(X))
35
36     # Run the model on input data
37     def get_score(self, input_data):
38         return self.model.score(input_data)
39
40 if __name__ == '__main__':
41     args = build_arg_parser().parse_args()
42     input_file = args.input_file
43
44     hmm_models = joblib.load("HMMmodels.pkl")
45     sampling_freq, audio = wavfile.read(input_file)
46     # Extract MFCC features
47     mfcc_features = mfcc(audio, sampling_freq)
48
49     # Define variables
50     max_score = -99999
51     output_label = None
52
53     for item in hmm_models:
54         hmm_model, label = item
55         score = hmm_model.get_score(mfcc_features)
56         if score > max_score:
57             max_score = score
58             output_label = label
59
60

Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.

New to Spyder? Read our tutorial

Variable explorer File explorer Help Profiler
IPython console
Console 1/A
Python 3.7.4 (default, Aug 9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.8.0 -- An enhanced Interactive Python.

In [1]:
```

# Otras aplicaciones

## Productos de IA y aprendizaje automático

## Try the API

## VISION AI

Faces

Objects

Labels

Logos

Web

Text

Properties

Safe Search

Información general

Demostración de la API Vision

Ventajas

Características

Clientes

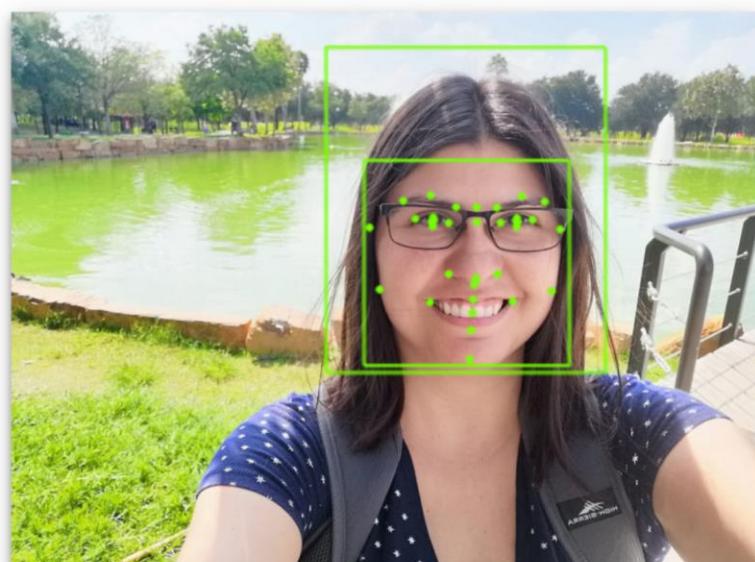
Casos prácticos

Precios

Recursos

Formación

Primeros pasos



mty.png

Joy  Very LikelySorrow  Very UnlikelyAnger  Very UnlikelySurprise  Very UnlikelyExposed  Very UnlikelyBlurred  Very UnlikelyHeadwear  Very Unlikely

Roll: 2° Tilt: -18° Pan: 5°

Confidence  100%

# Mr.Tuit<sup>Beta</sup>

¡me gusta!



Analiza

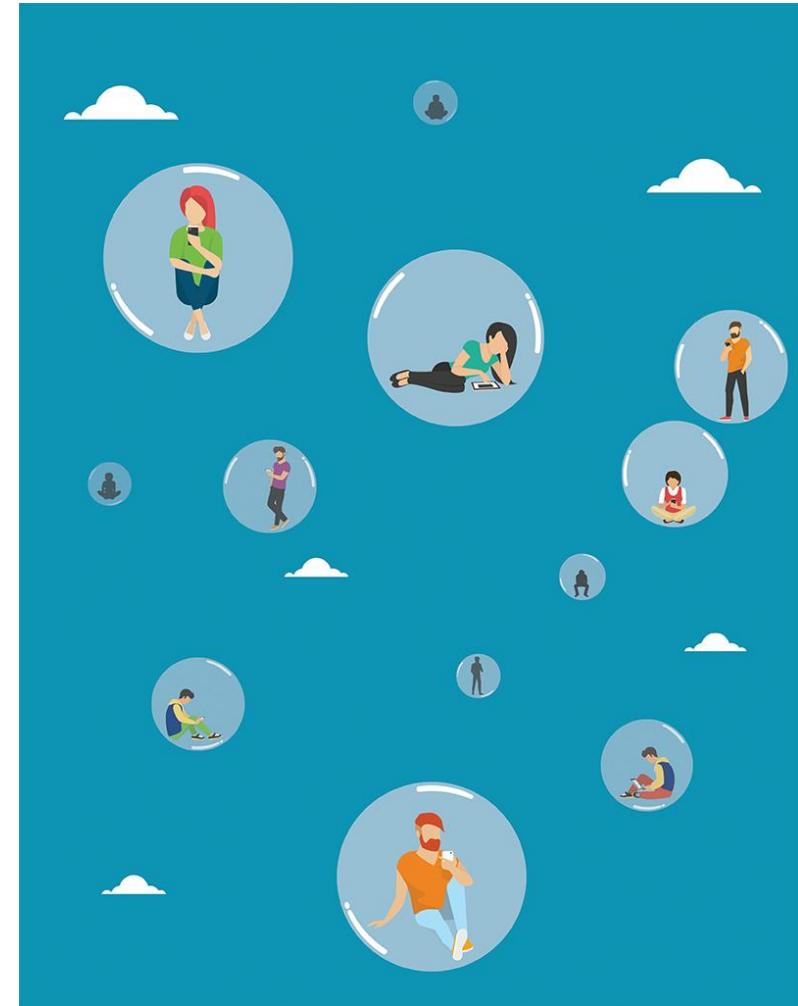
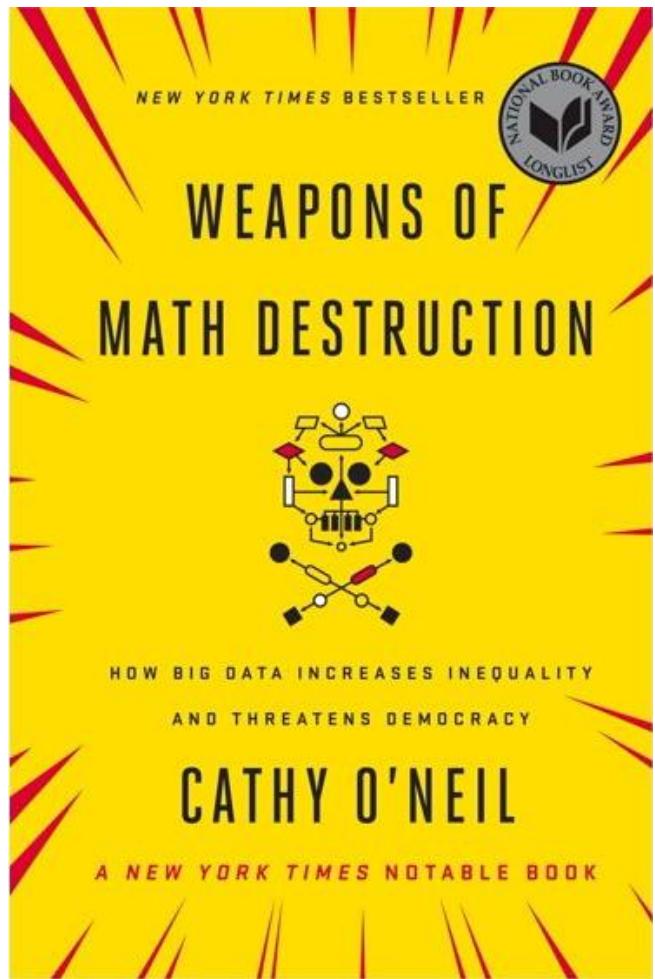


Positivo

[¿No estás de acuerdo?](#)



# Responsabilidades y riesgos



# Gracias!

[jessicabeltran@gmail.com](mailto:jessicabeltran@gmail.com)

jessicabeltran.net