Matemáticas y ciencia de Datos

Profesor: Jorge Dávila Ortiz

2021

Nota: Las notas, libros y datos usados en la charla están disponible en la liga http://webwork.utleon.edu.mx/Paginas/Robotica.html.

¿Qué es la Ciencia de Datos?

• La Ciencia de Datos pretende abarcar a un conjunto de herramientas (basadas en la ciencia) y habilidades (humanas e informáticas).
• Se define como es la extracción de conocimiento a partir de grandes volúmenes de información estructurada o no estructurada.
Segunda Definición:
•
Podemos decir que trata del estudio de la extracción generalizada de conocimiento a partir de información, de datos.

Conclusión:
¡ No existe una definición de consenso, sino que difiere según las fuentes!

¿Esto es algo nuevo? ¿No se parece a alguna ciencia con la que ya estamos familiarizados?

Data Science y Estadística

• **La Estadística** consiste en el estudio de la recolección, análisis, interpretación,
presentación y organización de datos.
• **La Ciencia de Datos** trata del estudio de la extracción generalizada de conocimiento a partir de información, de datos.

¿Estamos hablando de lo mismo?

Veamos la opinión de Jeff Wu, de la University of Michigan:

http://www2.isye.gatech.edu/~jeffwu/presentations/datascience.pdf

Conclusión:

• El enfoque de **Data Science** es más holístico, más global, para partiendo de grande volúmenes de datos poder extraer conocimiento que aporte valor a una determinad organización del tipo que sea.

• El foco principal se sitúa en la extracción de conocimiento, **empleando para ello las herramientas que estén al alcance.**

Holístico: (del griego ὅλος [hólos]: "todo", "por entero", "totalidad") es una posición metodológica y epistemológica que postula cómo los sistemas (ya sean físicos, biológicos, sociales, económicos, mentales, lingüísticos, etc.) y sus propiedades deben ser analizados en su conjunto y no sólo a través de las partes que los compone

Ya hemos podido intuir que se trata de algo más que la Estadística. **Así nuestra definición será:**

• Técnicas y teorías de muchos campos dentro de amplias áreas como la Matemática, la Estadística y las Tecnologías de la Información, incluyendo: procesamiento de señales, modelos probabilísticos, machine learning, aprendizaje estadístico, programación, ingeniería de datos, reconocimiento de patrones, visualización, modelización de la incertidumbre, data warehousing, y computación de altas prestaciones.

Conceptos que involucra

¿Qué es un Científico de Datos?

- Un Científico de Datos (Data Scientists) es una persona con habilidades estadísticas, computacionales (que sabe programar) y de visualización de datos que lo llevan a encontrar los patrones que le servirán a la empresa o institución para capitalizar la información recogida.

- El perfil a nivel internacional de Data Scientist o científico de datos

http://www.norbertogallego.com/data-scientist-empleo-indefinido-sin-reforma/2014/03/05/

¿Qué se necesita saber para ser un científico de datos?

- **Domine** las matemáticas, la estadística y la informática.
- Aprenda a programar.
- Conozca las Bases de Datos.
- Sea ágil en herramientas de procesamiento y visualización.

- **No deje de aprender y practicar.**

- **Aaahh!! y leer muuuuucho!**

- La ciencia de datos, tiene que ver, con, bueno….

“Datos”

¿Para qué la ciencia de datos?

- Toma de Decisiones Racionales
  - i.e tomar la mejor decision basada en la evidencia (datos) disponibles
- Aumento de la inteligencia
  - Plantea el problema, usa datos.
- Aplicar métodos científicos a la toma de decisiones.

  -Se ha intentado desde los 40’s, al parecer ahora si está teniendo impacto.

Retos de la Ciencia de datos

- No ignorar **la complejidad y no linealidad** del fenómeno
- Uno de los principales retos de la ciencia de datos es tratar
- con **la complejidad de los datos**.

Complejidad

¿Por qué no había funcionado?

- Nos habíamos conformado con los pocos datos que podíamos obtener (medir) y basado en eso reducíamos la dimensionalidad a unos pocos indicadores.
- No teníamos datos para hacer frente a la complejidad de la realidad, y jugábamos a la segura.
- Esta situación ya no es la actual.

¿Por qué no funcionaría ahora?

- **Pensar cartesianamente en un mundo no lineal, es en el mejor de los casos temerario, en el peor catastrófico.**

- **El cerebro humano piensa en causa/efecto lineal y coloca las causas fuera del sistema siempre que puede, ignorando las relaciones. (Fenomenos aleatorios, incertidumbre).**
- **No se pueden ignorar los ciclos de retroalimentación.**
- **No se puede ignorar los delays (retrasos) en el sistema.**

Complejidad de los datos

Data Complexity

- Volumen
- Semi-estructurado/No estructurado
- Variedad
- Conectividad
- Velocidad

Los datos y su topología.

Gunnar Carlsson Uno de los pioneros del ATD—“los datos tienen forma y esta forma importa”;

El arte de la ciencia de datos

En la práctica, la ciencia de datos exige

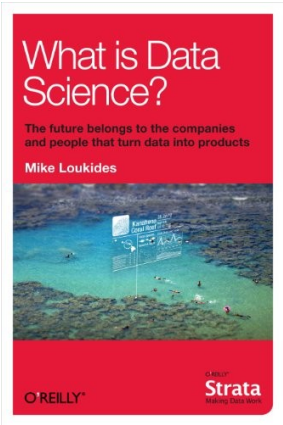
- ***Comprender mejor el problema o proyecto;***
- ***Perfiles/equipos multidisciplinarios;***
- ***considerar múltiples alternativas de implementación;***
- ***Tener paciencia para probar y probar alternativas hasta encontrar la mejor solución.***

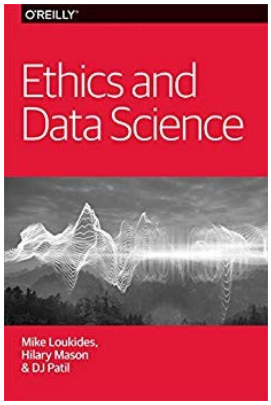
Algunas Lecturas Recomendadas:

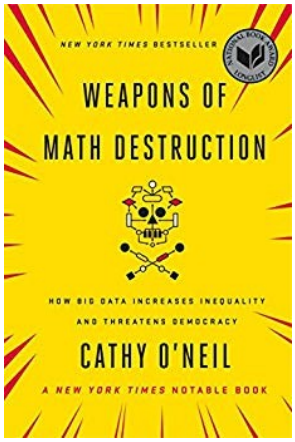
White Pater de Mike Loukides “What is Data Science”, el futuro pertenece a

las empresas que puedan encontrar la manera de recoger y utilizar los datos con éxito. El

futuro pertenece a las empresas y personas que conviertan los datos en productos







Ética y Ciencia de Datos

(Lecturas polémicas)

- ***Cambridge Analytica***

Cambridge Analytica

Python: Anaconda

- Es una plataforma integrada para ciencia de datos
- Varios entornos gráficos de programación
- Integra numpy, scpy, matploit, , etc.
- Disponible para Windows, Mac y Linux.
- Incluye instalación de Python.

Instalación de Anaconda

1) *Descargar Anaconda de* ***https://www.anaconda.com/distribution/#download-section***

la versión con python 3.8 , adecuada a tu sistema operativo.

2) Disfruta tu instalación.

Más información:

- **Documentación oficial de Python** https://docs.python.org/3/
- **Libro gratuito de "Python para todos": http://mundogeek.net/tutorial-python/**

Estadística en Python: Pandas, NumPy, SciPy

¿Por qué Python para estadística?

Python se ha convertido en uno de los lenguajes más usados en la comunidad de *data science*. Se trata de un lenguaje cómodo para los matemáticos, sobre el que es fácil iterar y cuenta con unas librerías muy maduras.

Usaremos en este curso Python 3. Tanto si estamos en Windows, macOS o Linux podemos instalar NumPy, SciPy.

¿Para qué sirve cada cosa?

> **NumPy** sirve para realizar cálculos numéricos con matrices de forma sencilla y eficiente y tanto SciPy como Pandas la usan de forma interna. NumPy es la base del stack científico de Python.
>
> **SciPy** es una colección de módulos dedicados a diversas áreas científicas. Nosotros usaremos principalmente el módulo **stats.**

scipy.stats: Este submodulo del paquete científico Scipy es el complemento perfecto para Numpy, las funciones estadísticas que no encontremos en uno, las podemos encontrar en el otro.

Pandas es una librería que permite manipular grandes conjuntos de datos en tablas con facilidad. Además permite importar y exportar esos datos.

Matplotlib permite realizar gráficos y diagramas con facilidad, mostrarlos en pantalla o guardarlos a archivos.

Más herramientas …

statsmodels: Esta librería nos brinda un gran número de herramientas para explorar *datos*, estimar modelos estadísticos, realizar pruebas estadísticas y muchas cosas más.

seaborn: Esta librería es un complemento ideal de matplotlib para realizar gráficos estadísticos.

statsmodels: Esta librería nos brinda un gran número de herramientas para explorar *datos*, estimar modelos estadísticos, realizar pruebas estadísticas y muchas cosas más.

Estadística descriptiva

1.- la estadística trata de darnos un resumen de un muestra (Conjunto) de datos.

*2.- Para ello vamos a definir el concepto de **variable estadística** como la magnitud o cualidad de los datos que queremos medir de la muestra (estatura, calificaciones en el examen, dinero en la cuenta,...). Las variables pueden ser **cualitativas** o **cuantitativas** y dentro de las cuantitativas pueden ser **continuas** y **discretas**.*

Fichero de ejemplo (Muestra Ejemplo)

```
alumno,nota
Araceli,9
Manuel,5
Pablo,7
Íñigo,4
Mario,3
Raúl,4
Verónica,6
Darío,10
Laura,4
Silvia,6
Eduardo,2
Susana,8
María,5
```

Nota: Guardamos los datos en un archivo de nombre notas.csv, para los ejemplo

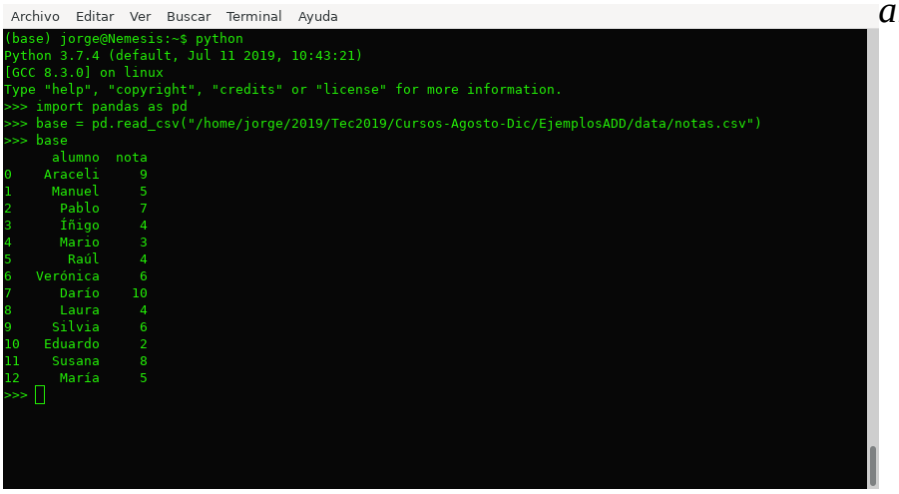
Cargando datos (forma local)

*Para cargar datos y manipular los datos usamos **Pandas**. Pandas permite cargar datos de distintos formatos: CSV, JSON, HTML, HDF5, SQL, msgpack, Excel, ...*

Para estos ejemplos usaremos CSV, valores separados por comas:

```
>>> import pandas as pd
>>> df = pd.read_csv("notas.csv")
```

***df** es un objeto de tipo **DataFrame**. Es la base de Pandas y como veremos, se trata de un tipo de dato muy flexible y muy fácil de usar. Si ahora hacemos **print** a df obtenemos algo así:*



Estadística en Python: media, mediana, varianza, percentiles

Siguiendo en nuestro camino de la **estadística descriptiva**, hoy vamos a ver como calcular ciertas medidas relativas a una variable.

Medidas de centralización: media, mediana y moda

La **media aritmética** se define como la suma de N elementos dividida entre N. Se trata una medida bastante conocida entre la gente, aunque tiene el inconveniente de que es muy susceptible a valores extremos.

> **Media aritmética:** La media aritmética es el valor obtenido al sumar todos los *datos* y dividir el resultado entre el número total elementos. Se suele representar con la letra griega μ. Si tenemos una muestra de n valores, ,

la *media aritmética*, μ, es la suma de los valores divididos por el numero de elementos; en otras palabras:

La **mediana** es el valor que dentro del conjunto de datos es menor que el 50% de los datos y mayor que el 50% restante.

La **moda** es el valor más repetido (solo aplicable a variables discretas).

Para calcular estas medidas, simplemente seleccionamos la **variable estadística** del DataFrame y usamos los métodos **mean, median** y **mode** respectivamente.

Para calcular estas medidas, simplemente seleccionamos la **variable estadística** del DataFrame y usamos los métodos **mean, median** y **mode** respectivamente.

Ejemplo: Calcula la media, la mediana y la moda de las notas de los alumnos en el examen

```
>>> import pandas as pd
>>> df = pd.read_csv("notas.csv")
>>> media = df["nota"].mean()
>>> mediana = df["nota"].median()
>>> moda = df["nota"].mode()
>>> print("""
    Media: %d
    Mediana: %d
    Moda: %d

        """ % (media,mediana,moda))
```

- ## Aprendizaje de Máquina (Machine Learning)

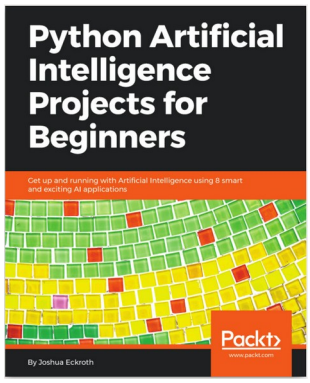
!! Bienvenido al mundo de scikit-learn!!, una biblioteca de aprendizaje muy poderosa, simple y expresiva.

¡¡Estoy realmente emocionado de ver qué se les ocurre¡¡

¡¡ Proyectos:!!

1.- Formar equipos de 4 personas.

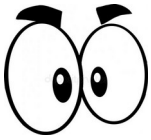
2. Elegir un problema para trabajar las siguientes 3 semana (Del siguiente libro).



Forma de Trabajo

En el transcurso de 3 semanas (semana 2,3 y 4 del bloque), los equipos darán exposiciones de los avances (iii y temas necesarios !!!) de sus trabajos.

Entorno de Trabajo: **Anaconda** con la librería **Scikit-learn.**



!! Checar que ya este instalada!!

Una forma de ver las librerias instaladas es usar el comando:

conda list (en windows)

```
(numpy2) C:\Users\admin>conda list
# packages in environment at C:\Users\admin\Anaconda3\envs\numpy2:
#
# Name                    Version                   Build  Channel
blas                      1.0                         mkl
ca-certificates           2019.1.23                     0
certifi                   2018.11.29               py27_0
curl                      7.26.0                        0
distribute                0.6.45                   py27_1
icc_rt                    2019.0.0                h0cc432a_1
intel-openmp              2019.1                        144
mkl                       2019.1                        144
numpy                     1.9.3                    py27_0
openssl                   1.1.1b                  h0c8e037_0
pip                       1.4.1                    py27_0
python                    2.7.15                  h2880e7c_4
scipy                     0.15.0                  np19py27_0
see                       1.4.1                    pypi_0    pypi
setuptools                0.6c11                   pypi_0    pypi
vc                        9                       h7299396_1
vs2008_runtime            9.00.30729.1            hfaea7d5_1
vs2015_runtime            14.15.26706             h3a45250_0

(numpy2) C:\Users\admin>
```

En caso linux:

```
(base) jorge@Nemesis:~/anaconda3/bin$ conda list | grep scikit*
scikit-image              0.15.0           py37he6710b0_0
scikit-learn              0.21.3           py37hd81dba3_0
(base) jorge@Nemesis:~/anaconda3/bin$
```

Instalación: (Comando)

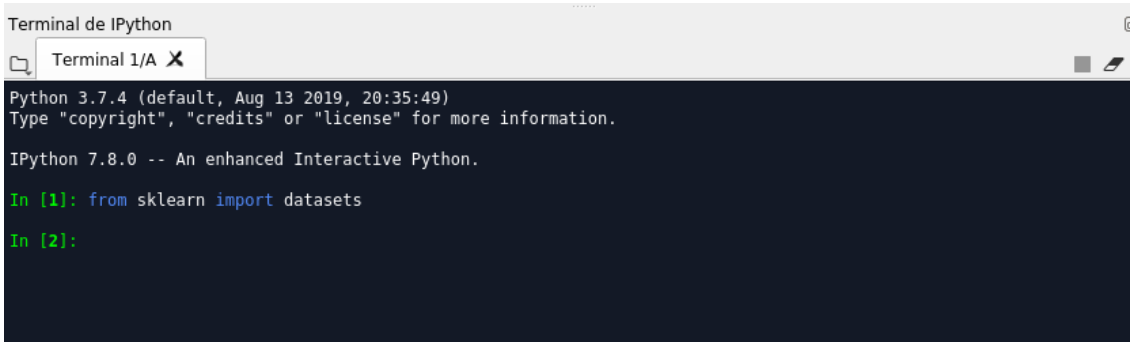
conda install scikit-learn

En el Principio fue la línea de comandos. ¡Algo de lectura!

... Comenzando con scikit-learn

Las bases de datos en scikit-learn están contenidos dentro del módulo **datasets** . Usamos el siguiente comando para importar estas bases de datos:

>>> **from sklearn import datasets**



Nota: Desde IPython, ejecutando **datasets.*?** , se enumerará **todo lo disponible** dentro de módulo **datasets**.

```
Python 3.7.4 (default, Aug 13 2019, 20:35:49)
Type "copyright", "credits" or "license" for more information.

IPython 7.8.0 -- An enhanced Interactive Python.

In [1]: from sklearn import datasets

In [2]: datasets.*?
datasets.__all__
datasets.__builtins__
datasets.__cached__
datasets.__class__
datasets.__delattr__
datasets.__dict__
datasets.__dir__
datasets.__doc__
datasets.__eq__
datasets.__file__
datasets.__format__
datasets.__ge__
datasets.__getattribute__
datasets.__gt__
datasets.__hash__
datasets.__init__
datasets.__init_subclass__
datasets.__le__
datasets.__loader__
datasets.__lt__
datasets.__name__
datasets.__ne__
datasets.__new__
datasets.__package__
```

Hay dos tipos principales de bases de datos dentro del módulo **datasets,** bases de prueba pequeñas están incluidas en el **paquete sklearn,** y se pueden ver ejecutando:

>>> **datasets.load_*?**

```
In [3]: datasets.load_*?
datasets.load_boston
datasets.load_breast_cancer
datasets.load_diabetes
datasets.load_digits
datasets.load_files
datasets.load_iris
datasets.load_linnerud
datasets.load_sample_image
datasets.load_sample_images
datasets.load_svmlight_file
datasets.load_svmlight_files
datasets.load_wine

In [4]:
```

Son las bases de datos que estan incluidas por default en **sklearn.**

Nota: Las bases de datos más grandes también están disponibles para descargar según sea necesario. **Estas últimas no están incluidas en sklearn por defecto**; sin embargo, a veces, son mejores para probar modelos y algoritmos debido a su **suficiente complejidad para representar situaciones realistas.**

Ejemplo 1: Carguemos la base de datos **Boston** y la examinamos un poco.

>>> **boston = datasets.load_boston()**

>>> boston

```
In [5]: boston = datasets.load_boston()

In [6]: boston
Out[6]:
{'data': array([[6.3200e-03, 1.8000e+01, 2.3100e+00, ..., 1.5300e+01, 3.9690e+02,
         4.9800e+00],
        [2.7310e-02, 0.0000e+00, 7.0700e+00, ..., 1.7800e+01, 3.9690e+02,
         9.1400e+00],
        [2.7290e-02, 0.0000e+00, 7.0700e+00, ..., 1.7800e+01, 3.9283e+02,
         4.0300e+00],
        ...,
        [6.0760e-02, 0.0000e+00, 1.1930e+01, ..., 2.1000e+01, 3.9690e+02,
         5.6400e+00],
        [1.0959e-01, 0.0000e+00, 1.1930e+01, ..., 2.1000e+01, 3.9345e+02,
         6.4800e+00],
        [4.7410e-02, 0.0000e+00, 1.1930e+01, ..., 2.1000e+01, 3.9690e+02,
         7.8800e+00]]),
 'target': array([24. , 21.6, 34.7, 33.4, 36.2, 28.7, 22.9, 27.1, 16.5, 18.9, 15. ,
        18.9, 21.7, 20.4, 18.2, 19.9, 23.1, 17.5, 20.2, 18.2, 13.6, 19.6,
        15.2, 14.5, 15.6, 13.9, 16.6, 14.8, 18.4, 21. , 12.7, 14.5, 13.2,
        13.1, 13.5, 18.9, 20. , 21. , 24.7, 30.8, 34.9, 26.6, 25.3, 24.7,
        21.2, 19.3, 20. , 16.6, 14.4, 19.4, 19.7, 20.5, 25. , 23.4, 18.9,
        35.4, 24.7, 31.6, 23.3, 19.6, 18.7, 16. , 22.2, 25. , 33. , 23.5,
        19.4, 22. , 17.4, 20.9, 24.2, 21.7, 22.8, 23.4, 24.1, 21.4, 20. ,
        20.8, 21.2, 20.3, 28. , 23.9, 24.8, 22.9, 23.9, 26.6, 22.5, 22.2,
        23.6, 28.7, 22.6, 22. , 22.9, 25. , 20.6, 28.4, 21.4, 38.7, 43.8,
        33.2, 27.5, 26.5, 18.6, 19.3, 20.1, 19.5, 19.5, 20.4, 19.8, 19.4,
        21.7, 22.8, 18.8, 18.7, 18.5, 18.3, 21.2, 19.2, 20.4, 19.3, 22. ,
        20.3, 20.5, 17.3, 18.8, 21.4, 15.7, 16.2, 18. , 14.3, 19.2, 19.6,
Terminal de IPython  Historial de comandos
```

¿Que tipo de estructura es boston?

>> type(boston)

```
In [8]:

In [8]: type(boston)
Out[8]: sklearn.utils.Bunch

In [9]:
```

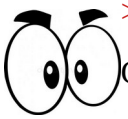
Nota:!! Cuando se cargan estos conjuntos de datos, no se cargan como matrices NumPy, ni un DataFrame de Pandas . Son de tipo **Bunch**. Un Bunch es una estructura de datos común en Python. Es esencialmente un diccionario con las llaves agregado al objeto como atributos.

>>>print (boston.DESCR)

DESCR nos dara una descripción básica de los datos para darle un poco de contexto.

```
Terminal 1/A
In [10]: print (boston.DESCR)
.. _boston_dataset:

Boston house prices dataset
---------------------------

**Data Set Characteristics:**

    :Number of Instances: 506

    :Number of Attributes: 13 numeric/categorical predictive. Median Value (attribute 14) is usually the target.

    :Attribute Information (in order):
        - CRIM     per capita crime rate by town
        - ZN       proportion of residential land zoned for lots over 25,000 sq.ft.
        - INDUS    proportion of non-retail business acres per town
        - CHAS     Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
        - NOX      nitric oxides concentration (parts per 10 million)
        - RM       average number of rooms per dwelling
        - AGE      proportion of owner-occupied units built prior to 1940
        - DIS      weighted distances to five Boston employment centres
        - RAD      index of accessibility to radial highways
        - TAX      full-value property-tax rate per $10,000
        - PTRATIO  pupil-teacher ratio by town
        - B        1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
        - LSTAT    % lower status of the population
        - MEDV     Median value of owner-occupied homes in $1000's
Terminal de IPython | Historial de comandos
```

Ejemplo 2. Buscando una base de Datos (Viviendas en california).

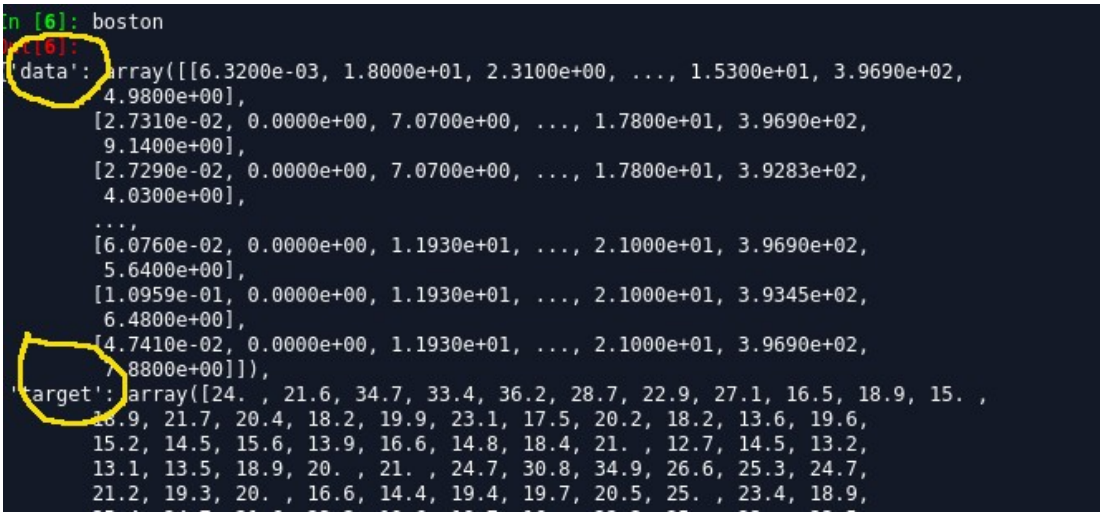


>>> housing = datasets.fetch_california_housing()

Conexión a Internet ….cuidado!!

Nota : usamos el atributo fetch.

```
In [11]: housing = datasets.fetch_california_housing()
Downloading Cal. housing from https://ndownloader.figshare.com/files/5976036 to /home/jorge/scikit_learn_data

In [12]: print (housing.DESCR)
.. _california_housing_dataset:

California Housing dataset
--------------------------

**Data Set Characteristics:**

    :Number of Instances: 20640

    :Number of Attributes: 8 numeric, predictive attributes and the target

    :Attribute Information:
        - MedInc        median income in block
        - HouseAge      median house age in block
        - AveRooms      average number of rooms
        - AveBedrms     average number of bedrooms
        - Population    block population
        - AveOccup      average house occupancy
        - Latitude      house block latitude
```



>>> X, y = boston.data, boston.target

¡sorpresa!, X, y son matrices NumPy.

>>> type(X)

El internet de las cosas IoT.

Algunas lecturas.