# **Python para todos: 5 formas de cargar datos para tus proyectos de Machine Learning**

#### 1. Usando el [módulo csv](https://docs.python.org/3.6/library/csv.html?highlight=csv#module-csv) de la librería estándar

Para extraer los datos de un fichero *csv,* primero lo «abrimos» con la función [*open*](https://www.w3schools.com/python/python_file_handling.asp)*()*. Después, usamos la función, *csv.reader*(), que lee línea a línea el fichero, y hace una lista de todas las columnas en el objeto *reader.*

#### 2. Usando NumPy

Otra forma de cargar los datos es usando la función [*numpy.loadtxt()*](https://penandpants.com/2012/03/09/reading-text-tables-with-python/) de la librería [NumPy](https://empresas.blogthinkbig.com/python-todos-3-librerias/). Para ello, lo primero que tendremos que hacer es cargar la librería.

Esta función asume que el fichero no tiene cabeceras, y que todos los datos tienen el mismo formato. Como no es el caso de nuestro dataset sobre diabetes, usaremos el modificador *«skiprow*» para indicar que no debe tener en cuenta la primera fila.

#### 3. Usando Pandas

Esta tercera forma de cargar los datos es, probablemente, la más popular entre los científicos de datos como veremos, por muy buenas razones. En este caso, la función que usaremos es [*readcsv()*](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_csv.html). Esta función nos ofrece una gran flexibilidad a la hora de importar datos, ya que de forma automática, o con una simple línea de comando, permite hacer cosas como:

* Detectar automáticamente las cabeceras o «headers»
* «Saltar» líneas con el modificador «*skiprow*«
* Detectar automáticamente el tipo de datos (número entero, decimal, cadena de texto etc)
* Identificar campos con datos erróneos o vacíos
* Convertir los datos en formato csv en un dataframe de Pandas

#### 4. Desde URL.

Otra de las formas más habituales de cargar un dataset es directamente desde la URL donde se alojan los datos. Este método es perfectamente compatible con los anteriores. Por ejemplo, podemos modificar el ejemplo anterior, para que cargue los datos desde su ubicación original.  
5. Desde otras librerías. Cuando lo que queremos es hacer una prueba rápida, nos pueden resultar muy útiles los datasets que traen «de serie» algunos de los paquetes más populares como *statsmodels, scikit-learn, o seaborn.*

## **Importar datos en** [**Python**](https://sitiobigdata.com/2019/05/01/python-5-caracteristicas-avanzadas-como-usarlas/)

*numpy .loadtxt* – Esta función pretende ser un lector rápido para archivos simplemente formateados. La función *genfromtxt* proporciona un manejo más sofisticado de, por ejemplo, líneas con valores perdidos.

*numpy.genfromtxt* – Cargar datos de un archivo de texto, con valores perdidos manejados como se especifica. Función mucho más sofisticada que tiene muchos parámetros para controlar su importación.

data = np.genfromtxt('titanic.csv', delimiter=',', name

#### Pickle, **¿Qué es pickle?**

Se usa para serializar y deserializar una estructura de objeto de Python. Cualquier objeto en Python se puede escanear para que pueda guardarse en el disco.

Lo que Pickle hace es que “serializa” el objeto primero antes de escribirlo en el archivo. Decapado es una forma de convertir un objeto python (lista, dict, etc.) en una secuencia de caracteres.

#### HDF5

Hierarchical Data Format (HDF) es un conjunto de formatos de archivo (HDF4, HDF5) diseñado para almacenar y organizar grandes cantidades de datos.

HDF5 es una suite tecnológica única que hace posible la administración de colecciones de datos extremadamente grandes y complejas.



