

Instalación simplificada del ambiente Anaconda Navigator para desarrollar proyectos de Machine Learning

Como primer paso para manejar nuestros conjuntos de datos (data sets) utilizaremos la plataforma Anaconda Navigator que nos facilitará la tarea de trabajar con el lenguaje Python y también las Jupyter Notebooks, que funcionan como un cuaderno de notas interactivo donde poder ir paso a paso haciendo nuestros ejercicios y nuestros propios proyectos de Aprendizaje Automático.

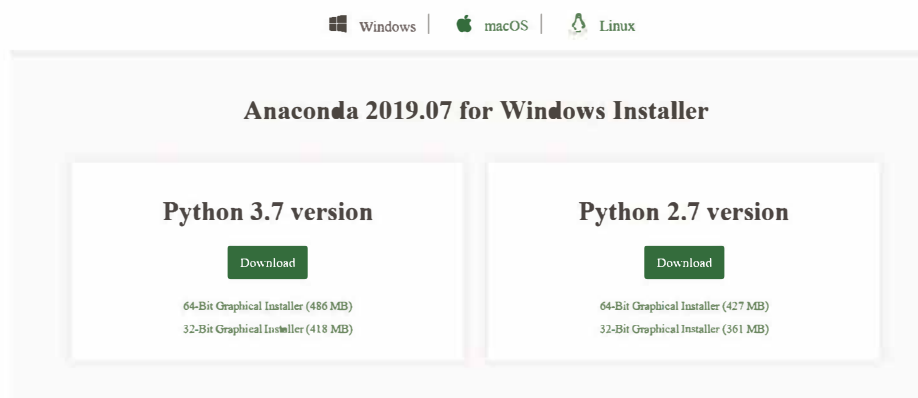
Anaconda Navigator es una interfaz gráfica de usuario (GUI) de escritorio incluida en la distribución Anaconda® que le permite iniciar aplicaciones y administrar fácilmente paquetes, Navigator puede buscar paquetes en el Anaconda Cloud o en otros repositorios y está disponible para ambientes Windows, macOS y Linux de distribución libre y sin ningún coste.

Vamos a simular la instalación de Anaconda Navigator en un ordenador de escritorio con ambiente Windows 10, pero las instrucciones son muy similares para ambientes Mac OS.

1) Lo primero es ingresar a ésta dirección y descargar la versión ó distribución que corresponde a nuestro Ordenador (Windows, Mac , o Linux). OJO no hay disponibilidad de Anaconda para Smartphones ni tablets.

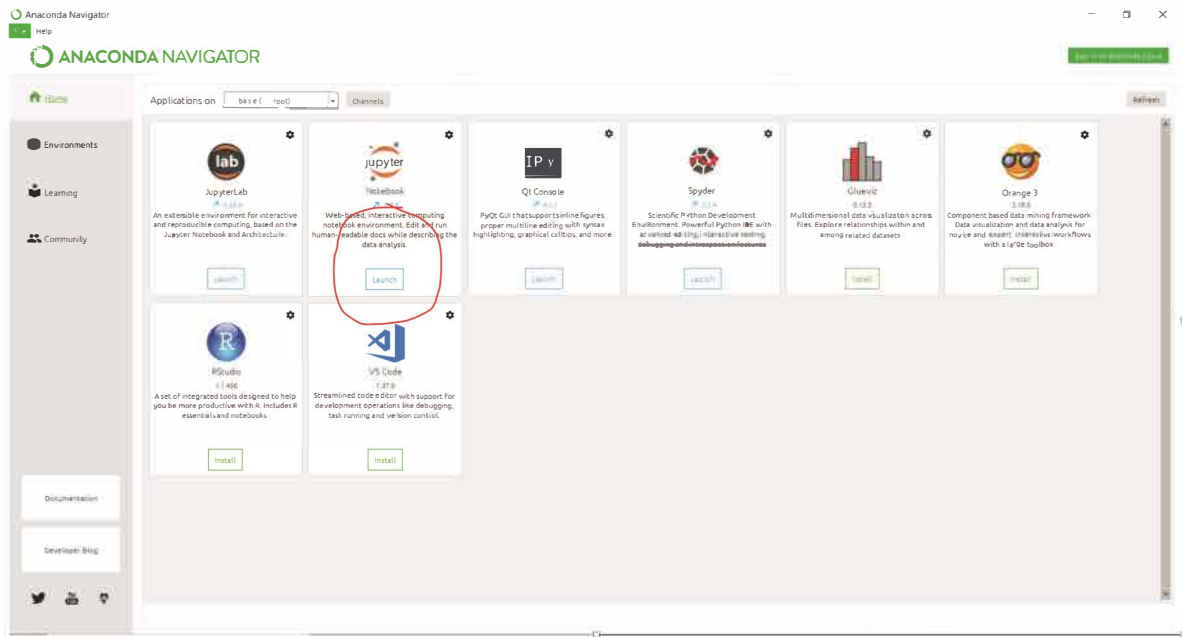
<https://www.anaconda.com/distribution/#download-section>

Escoja la versión 3.7 y la versión de 32 o 64 bits dependiendo del tipo de procesador de su ordenador. (En los últimos años la mayoría de arquitecturas de los procesadores de ls ordenadores son de 64 bits.) seleccionar la opción Graphic Installer.

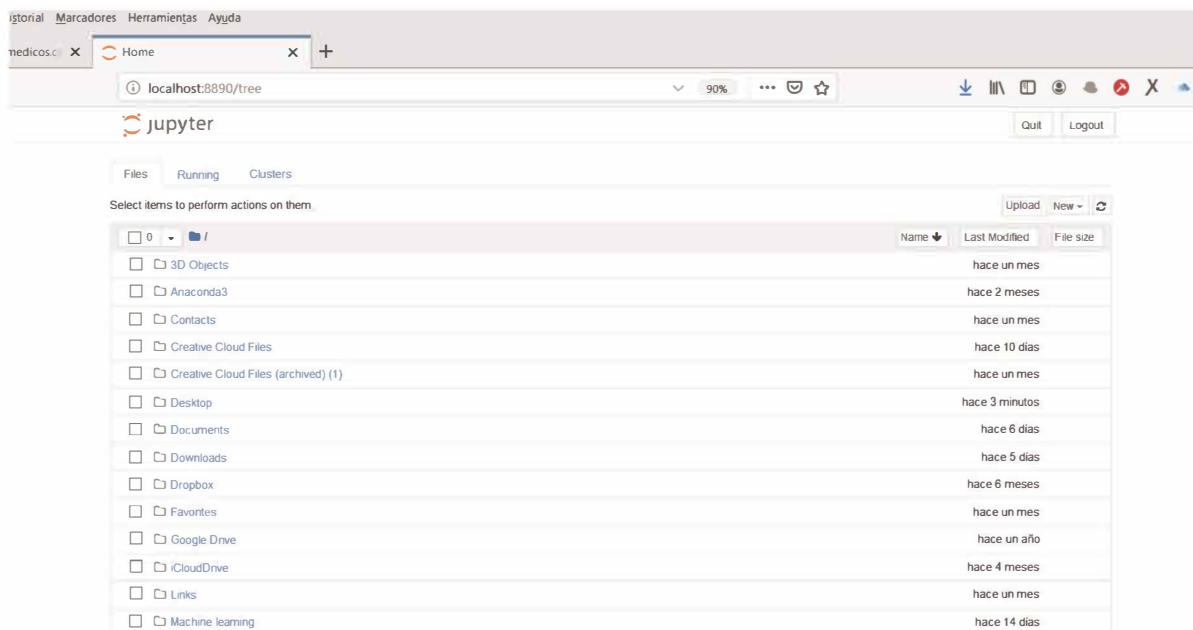


2) El segundo paso es la instalación de Anaconda. Debe tener permisos de administrador en el ordenador antes de instalar esta aplicación. La instalación completa requiere de al menos 1GB de espacio en el ordenador. La instalación es a través de un Asistente (Wizard) lo que facilitará mucho el proceso.

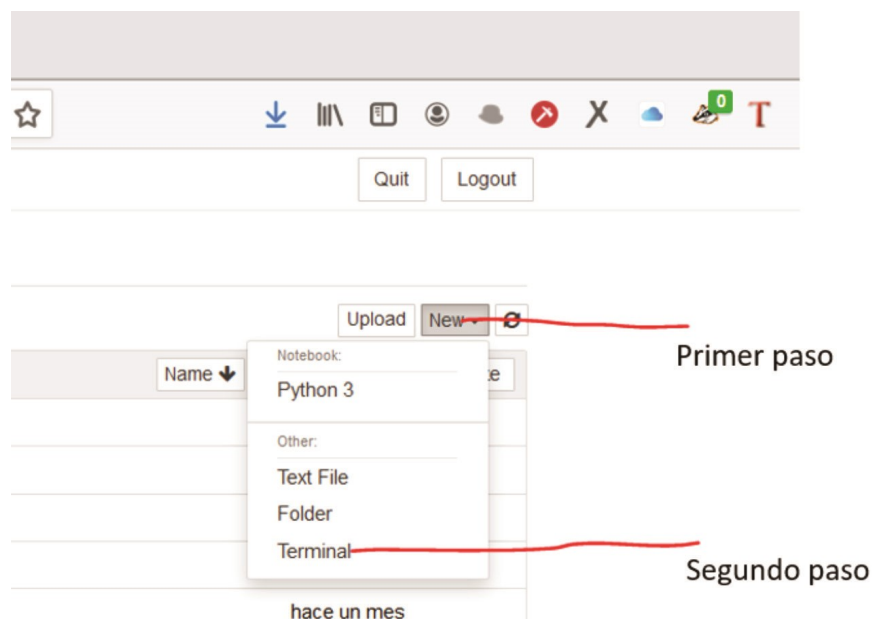
3) El tercer paso es iniciar Anaconda, (puede tardar algún tiempo) donde nos presenta una pantalla similar a esta:



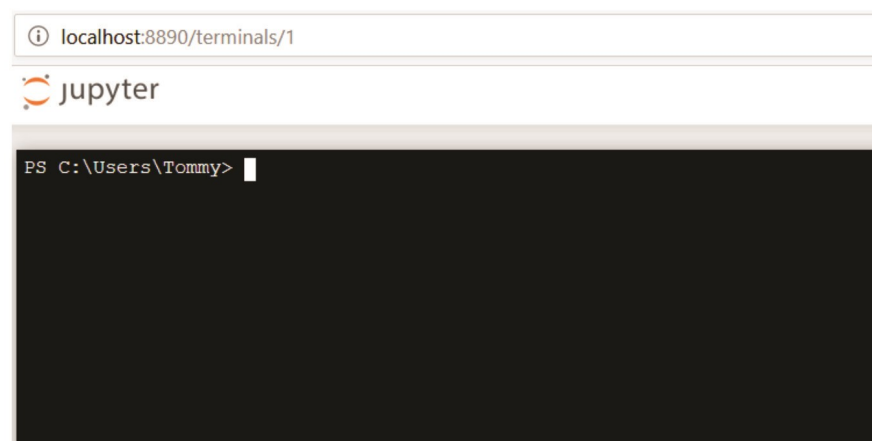
En este paso se nos abrirá una ventana en nuestro navegador de Internet predeterminado. (prefiera Mozilla Firefox) La pantalla de las Jupyter Notebooks es ésta: (No cierre la ventana de Anaconda, simplemente minimízela)



5) Se requiere ahora instalar otra serie de librerías, las cuales se hacen a través de una ventana de Terminal. Para ingresar a la ventana terminal vaya al lado derecho en el botón NEW, y seleccione la última opción: Terminal.



La ventana terminal es similar a ésta: (en su caso saldrá su usuario y el directorio donde se encuentra).



6) Ahora deberá ir escribiendo estos comandos (ponga especial cuidado en los espacios y en las mayúsculas /minúsculas) si obtiene un error, puede escribirlo de nuevo.

7) Ahora actualizamos a la versión mas reciente de Anaconda , con el comando:

`Conda update conda`

Le decimos "y" para proceder e instalamos las actualizaciones disponibles. Luego hacemos lo mismo ejecutando este comando

```
PS C:\Users\Tommy> conda update conda
Collecting package metadata: done
Solving environment: done
```

```
## Package Plan ##
```

```
environment location: C:\Users\Tommy\Anaconda3
```

```
added / updated specs:
- conda
```

```
The following packages will be downloaded:
```

package	build	
ca-certificates-2019.5.15	1	166 KB
certifi-2019.6.16	py37_1	156 KB
conda-4.7.11	py37_0	3.0 MB
Total:		3.3 MB

```
The following packages will be UPDATED:
```

ca-certificates	2019.5.15-0 --> 2019.5.15-1
certifi	2019.3.9-py37_0 --> 2019.6.16-py37_1
conda	4.6.14-py37_0 --> 4.7.11-py37_0

```
Proceed ([y]/n)?
```

```
conda update anaconda
```

Y luego instalamos la librería scikit, y para ellos escribimos:

```
conda update scikit-learn
```

8.) Para verificar que todo está bien instalado escribimos:

```
conda -V
```

y luego

```
python -V
```

```
PS C:\Users\Tommy> conda -V
Conda 4.6.14
PS C:\Users\Tommy> python -V
Python 3.7.0
PS C:\Users\Tommy>
```

9) Instalar las librerías de Aprendizaje Profundo (Deep Learning)

Escribir los siguientes comandos:

```
conda install -c conda-forge tensorflow
```

```
pip install keras
```

10) Las Jupyter Notebooks permiten acceder a todos los directorios del ordenador. Se recomienda abrir un directorio donde pondremos nuestros programas, nuestros conjuntos de datos y todo lo relacionada a cada experimento. El escritorio de Windows por ejemplo se puede utilizar para este fin (A modo de ejemplo he creado una carpeta en el escritorio llamado Modelos de Machine Learning.) y que lo vamos a encontrar en nuestro directorio Escritorio (Desktop)

Si tenemos un editor de texto como notepad de Windows podemos crear un Archivo que incluya estas líneas que serán nuestro primer programa de Python al que llamaremos "versiones.py" y que guardaremos en el escritorio (Desktop) También puede descargar este archivo desde el repositorio (ver ANEXOS) Verificar que el nombre del archivo sea versiones.py y que NO sea versiones.txt



Seguidamente ejecutaremos nuestro programa desde la Terminal
Escribiendo el comando:

```
python versiones.py
```

OJO: Tomar nota que el programa deberá estar ubicado en el escritorio
Y la ventana de Terminal así debe indicarlo con el nombre C:\Usuarios\XXXXX\
Escritorio>

Si la versión instalada es la de Windows en inglés, el escritorio aparecerá como \
Desktop> Si todo está correcto deberá salir un resultado como este:

```
PS C:\Users\Tommy\desktop> python versiones.py
scipy: 1.2.1
numpy: 1.16.4
matplotlib: 3.1.0
pandas: 0.24.2
statsmodels: 0.9.0
sklearn: 0.20.3
PS C:\Users\Tommy\desktop>
```

Si queremos verificar las versiones de las nuevas librerías de Deep learning instaladas podemos escribir
El siguiente código al que llamaremos deep.py
Este trozo de código también puede descargarse desde el repositorio (Ver ANEXO)

```
# tensorflow
import tensorflow
print(' tensorflow: %s' % tensorflow.__version__)
# keras
import keras
print('keras: %s' % keras.__version__)
```

Si ejecutamos el comando

```
python deep.py
```

El resultado será:

```
PS C:\Users\Tommy\desktop> python deep.py
tensorflow: 1.14.0
Using TensorFlow backend.
keras: 2.2.4
PS C:\Users\Tommy\desktop>
```

¿Pero qué es el lenguaje Python?

Python es el lenguaje de programación más popular en proyectos de Machine Learning por su facilidad de uso y versatilidad lo hace un lenguaje amigable y de amplio crecimiento.

Python está diseñado para ser sencillo, elegante, consistente y similar a las matemáticas. ¡Tiene una sintaxis consistente y se asemeja bastante al lenguaje humano y matemático, hasta el punto de que puede ser escrito y leído! como si fuera un pseudocódigo.

Pero además Python es un lenguaje de programación multi paradigma. Esto significa que más que forzar a los programadores a adoptar un estilo particular de programación, permite varios estilos: programación orientada a objetos, programación imperativa y programación funcional. Otros paradigmas están

soportados mediante el uso de extensiones.

Es administrado por la Python Software Foundation. Posee una licencia de código abierto, y por tanto su uso es gratuito y su distribución es libre.

Características de Python

• Lenguaje fácil y fácilmente legible

El siguiente es un ejemplo muy básico de un programa escrito en Python que pregunta cuánto es $2 + 2$, el programa espera por una respuesta, y si la respuesta es 4 escribirá: "Respuesta correcta" en cualquier otro caso escribirá, "Respuesta Incorrecta, lo correcto es 4".

```
puntaje = 0
print ("Cuanto es 2+2?")
answer = input()
answer = int(answer)

if answer == 4:
    print ("Respuesta Correcta")
    puntaje = puntaje + 1
else:
    print ("Respuesta incorrecta, lo correcto es 4")
# este es nuestro primer programa en Python
```

Este pequeño trozo de código, incluye comandos como:

a) If y else son condicionantes, si se cumple tal condición, el programa hace algo, en caso contrario hace otra cosa (else)

b) El comando input espera por una respuesta.

c) Los signos $=$ y $==$ implican asignación de un valor, o bien igualan un valor respectivamente. (al inicio se le asigna a la variable puntaje un valor de CERO, más abajo con el comando if (si respuesta es IGUAL a $==$ 4, entonces responde.....)

d) Cualquier expresion dentro de nuestro programa Python después del signo #, Python lo ignora lo considera simplemente un comentario.

```
# Esta línea el lenguaje Python la ignora.
```

•Python Posibilita la rápida creación de prototipos

Usualmente, los científicos de datos requieren de hacer y probar muchos y diferentes modelos este lenguaje favorece iteraciones rápidas para que el investigador pueda concentrarse en los datos y los algoritmos. Su sintaxis como se ve es sencilla y su naturaleza dinámica lo hace perfecto para la creación de prototipos.

- **Un lenguaje para todos**

Sus características funcionales trabajan como un puente entre los académicos y los ingenieros, ya que muchos de los científicos de datos e ingenieros de Machine Learning provienen de campos matemáticos y estadísticos. No tienen gran experiencia en la creación de código, pero las facultades de legibilidad de Python posibilitan un rápido aprendizaje e implementación en los proyectos de Machine Learning.

- **El ambiente perfecto para Machine Learning y Ciencia de Datos**

Hay numerosas librerías de ciencia de datos y librerías matemáticas que pueden incorporarse con solo un simple comando. El diseño modular y empaquetado de Python permite construir nuevas librerías sobre las ya existentes, lo que conduce a obtener mayor versatilidad y potencia.

- **Excelente Rendimiento.**

A pesar de que es un lenguaje de programación de alto nivel, Python permite el uso de librerías de extensión como NumPy y SciPy, para operaciones de vectorizado rápido en tablas multidimensionales. Todo esto hace que Python sea uno de los lenguajes más rápidos para realizar tareas de Ciencia de Datos y Machine Learning.

