

Tutorial de instalación y configuración para el proyecto ML

PyCharm (JetBrains) + Anaconda

Requisitos mínimos para Anaconda

- **Procesador:** Un procesador de 64 bits de doble núcleo a 1.6 GHz o superior.
- **Memoria RAM:** 8 GB de RAM (se recomiendan 16 GB o más para proyectos grandes).
- **Almacenamiento:** 20 GB de espacio libre en disco (se recomiendan 50 GB o más si vas a trabajar con conjuntos de datos grandes).
- **Sistema operativo:** Windows 10 u 11 (64 bits), macOS 10.15 o posterior, o una distribución de Linux compatible.

Recomendaciones adicionales

- **Procesador:** Se recomienda un procesador de cuatro núcleos o más para un rendimiento óptimo, especialmente si vas a trabajar con modelos de Machine Learning complejos.
- **Memoria RAM:** Si vas a trabajar con conjuntos de datos grandes o modelos que requieren mucha memoria, 16 GB de RAM o más son esenciales.
- **Almacenamiento:** Un disco de estado sólido (SSD) puede mejorar significativamente los tiempos de carga y procesamiento de datos en comparación con un disco duro tradicional (HDD).
- **Tarjeta gráfica (GPU):** Si vas a trabajar con Deep Learning, una tarjeta gráfica dedicada con soporte para CUDA (NVIDIA) o ROCm (AMD) puede acelerar el entrenamiento de modelos.

Consideraciones adicionales

- **Uso de entornos virtuales:** Es altamente recomendable utilizar entornos virtuales en Anaconda para aislar las dependencias de cada proyecto y evitar conflictos entre librerías.
- **Actualización de paquetes:** Mantén tus paquetes de Anaconda actualizados para asegurar el mejor rendimiento y compatibilidad.
- **Optimización del código:** Escribe código eficiente y optimizado para evitar el uso innecesario de recursos.

En resumen

Si tu computadora cumple con los requisitos mínimos y sigues las recomendaciones adicionales, Anaconda debería funcionar de manera fluida para la mayoría de los proyectos de Machine Learning. Sin embargo, si vas a trabajar con proyectos muy grandes o complejos, es posible que necesites un equipo más potente.

1. Instalación de Python

- Si aún no tienes Python instalado, puedes descargarlo desde el sitio web oficial: <https://www.python.org/downloads/>
- Elige la versión más reciente y estable de Python (Python 3.11 o superior).
- Descarga el instalador para tu sistema operativo (Windows, macOS o Linux) y sigue las instrucciones de instalación.
- Asegúrate de marcar la opción para agregar Python a tu variable de entorno PATH durante la instalación.

Versiones de python recomendadas: 3.9 en adelante.

Por qué elegir Python 3.13.2 para Machine Learning?

- **Estabilidad:** Es una versión estable, lo que significa que ha sido probada y corregida para minimizar errores y asegurar un funcionamiento confiable.
- **Soporte:** Cuenta con soporte completo de la comunidad de Python y de las principales librerías de Machine Learning.
- **Actualizaciones:** Incluye mejoras de rendimiento, seguridad y nuevas características que pueden ser útiles para tus proyectos.
- **Compatibilidad:** Es compatible con la mayoría de las librerías y herramientas de Machine Learning más populares, como TensorFlow, PyTorch, scikit-learn, pandas y NumPy.

¿Dónde descargar Python 3.13.2?

Puedes descargar la versión 3.13.2 de Python desde el sitio web oficial:

- <https://www.python.org/downloads/>

2. Instalación de Anaconda

- Anaconda es una distribución de Python que incluye muchas librerías populares para Machine Learning y ciencia de datos.
- Descarga Anaconda desde el sitio web oficial: <https://www.anaconda.com/products/individual>

- Elige la versión de Anaconda que corresponda a tu sistema operativo y a la versión de Python que prefieras (Anaconda incluye su propia versión de Python, por lo que no es necesario instalar Python por separado si vas a usar Anaconda).
- Sigue las instrucciones de instalación.

Qué es Miniconda?

Miniconda es un instalador mínimo de Conda. A diferencia de Anaconda, que instala una gran cantidad de paquetes predefinidos, Miniconda solo instala Conda, Python y las dependencias básicas. Esto lo hace más ligero y rápido de instalar, ideal si solo necesitas Conda para gestionar tus entornos y paquetes.

Pasos para instalar Miniconda en Windows

1. Descarga Miniconda:

- Ve al sitio web de Miniconda: <https://docs.conda.com/en/latest/miniconda.html>
- Elige la versión de Miniconda que corresponda a tu sistema operativo (Windows) y a la versión de Python que desees (por ejemplo, Python 3.9).
- Descarga el instalador.

2. Ejecuta el instalador:

- Ejecuta el archivo instalador descargado.
- Sigue las instrucciones en pantalla. Generalmente, deberás aceptar los términos de licencia y elegir la ubicación de instalación.
- **Importante:** Durante la instalación, asegúrate de marcar la opción para añadir Miniconda a tu variable de entorno PATH. Esto te permitirá ejecutar comandos de Conda desde la línea de comandos.

3. Verifica la instalación:

- Abre una nueva ventana de línea de comandos (símbolo del sistema o PowerShell).
- Escribe el siguiente comando y presiona Enter:

```
Bash
conda --version
```

4.

- Si Miniconda se instaló correctamente, mostrará la versión de Conda.

Ventajas de usar Miniconda

- **Ligero:** La instalación es más pequeña y rápida que Anaconda.
- **Personalizable:** Solo instalas los paquetes que necesitas.

- **Control:** Tienes más control sobre tu entorno.

Desventajas de usar Miniconda

- **Menos paquetes preinstalados:** Tendrás que instalar manualmente las librerías que necesites (NumPy, pandas, scikit-learn, etc.).
- **Requiere más configuración:** Puede requerir un poco más de configuración inicial.

En resumen

Si prefieres tener un control más preciso sobre los paquetes que instalas y no necesitas la gran cantidad de librerías que vienen con Anaconda, Miniconda es una excelente opción. Te permite tener Conda y Python en tu sistema de manera más ligera y personalizada.

3. Instalación de JetBrains PyCharm

- Descarga e instala la versión de PyCharm que mejor se adapte a tus necesidades (Professional o Community). La versión Professional ofrece más funcionalidades para Machine Learning, pero la Community es gratuita y suficiente para muchos proyectos. Puedes descargar PyCharm desde el sitio web oficial de JetBrains:
<https://www.jetbrains.com/pycharm/>

4. Configuración del intérprete de Python en PyCharm

- Abre PyCharm y crea un nuevo proyecto o abre uno existente.
- Ve a "File" > "Settings" (o "PyCharm" > "Preferences" en macOS).
- Selecciona "Project: [Tu proyecto]" > "Python Interpreter".
- Si no tienes un intérprete configurado, haz clic en el icono de engranaje y selecciona "Add...".
- Puedes elegir un intérprete existente (si ya tienes uno instalado) o crear uno nuevo. Se recomienda crear un entorno virtual para cada proyecto.
- Si eliges crear un nuevo entorno virtual, puedes usar **venv** (incorporado en Python 3) o **conda**.

- **venv:**

<!-- end list -->

Bash

```
python3 -m venv .venv # Crea el entorno virtual
source .venv/bin/activate # Activa el entorno virtual (Linux/macOS)
.venv\Scripts\activate # Activa el entorno virtual (Windows)
```

-

- **conda:**

<!-- end list -->

Bash

```
conda create --name mi_entorno python=3.9 # Crea el entorno virtual
conda activate mi_entorno # Activa el entorno virtual
```

-

Dentro del entorno virtual activado, instala las librerías necesarias:

Bash

```
pip install numpy pandas matplotlib scikit-learn tensorflow # Librerías básicas
```

-

5. Instalación de paquetes en PyCharm

- Dentro de PyCharm, puedes instalar paquetes directamente desde la ventana de herramientas "Python Packages".
- Busca el paquete que necesitas y haz clic en "Install".

6. Configuración de herramientas en PyCharm

- **Jupyter Notebook:** PyCharm Professional ofrece integración con Jupyter Notebook. Puedes crear y editar notebooks directamente dentro del IDE.
- **Depurador:** PyCharm cuenta con un potente depurador que te permite ejecutar tu código paso a paso, inspeccionar variables y encontrar errores.
- **Control de versiones:** PyCharm se integra con sistemas de control de versiones como Git, lo que facilita la gestión de tus proyectos.

7. Configuraciones adicionales en PyCharm

- **Esquema de color:** Elige un esquema de color que te guste para que tu entorno de trabajo sea más agradable.
- **Atajos de teclado:** Aprende los atajos de teclado más útiles para aumentar tu productividad.
- **Plugins:** Explora los plugins disponibles para PyCharm. Algunos plugins pueden ser muy útiles para tareas específicas de Machine Learning.

Consejos adicionales

- **Organización del proyecto:** Organiza tu proyecto en carpetas y archivos lógicos para que sea más fácil de mantener y entender.
- **Documentación:** Documenta tu código para que sea más fácil de entender por ti y por otros.
- **Comentarios:** Utiliza comentarios para explicar tu código y para dejar notas importantes.