
# DVC: Versionado de datos y accesos remotos

Curso de Ingeniería de Características

Julio Waissman

https://mcd-unison.github.io/ing-caract/`

# DVC (Data Version Control)

- Herramienta de código abierto diseñada para el control de versiones de datos y modelos.
- Está inspirada en Git y permite versionar datos, sin duplicar archivos grandes en el repositorio git.
- En proyectos de ciencia de datos, es crucial poder rastrear los cambios en los conjuntos de datos y los modelos entrenados.

# Cómo funciona

- **Versionado de datos**: DVC rastrea los cambios en tus conjuntos de datos, incluyendo las versiones, las modificaciones y las dependencias. Esto permite rastrear la línea de tiempo de tu trabajo y reproducir resultados con precisión.
- **Almacenamiento eficiente**: DVC almacena los datos en un repositorio Git, pero solo guarda las diferencias entre las versiones. Esto reduce el tamaño del repositorio y acelera las operaciones de clonado y descarga.
- **Acceso remoto**: DVC permite compartir datos con miembros de tu equipo y colaborar en proyectos de forma eficiente. Puedes acceder a las versiones de los datos desde cualquier lugar y en cualquier momento.

# ¿Cómo se utiliza DVC?

1. **Instalación**: DVC se instala fácilmente con `pip install dvc` o `conda install dvc`
2. **Inicialización**: Se crea un archivo `dvc.yaml` para definir los archivos de datos y sus metadatos.
3. **Seguimiento de datos**: Se utiliza el comando `dvc add` para comenzar a rastrear los archivos de datos en DVC.
4. **Almacenamiento remoto**: Se configura un almacenamiento remoto (como un servidor S3 o carpeta local) para almacenar los datos de forma eficiente.
5. **Versionado de datos**: Se utiliza el comando `dvc commit` para realizar un seguimiento de los cambios en los datos y guardar las versiones en el repositorio.
6. **Acceso remoto**: Los miembros del equipo pueden acceder a las versiones de los datos mediante comandos como `dvc pull` y `dvc push`.

# ¿Qué es un remoto en DVC?

Un remoto en DVC es un almacenamiento externo donde se pueden guardar los datos versionados. Esto puede ser un sistema de almacenamiento en la nube (como S3, Google Drive) o un servidor SSH, entre otros.

# Configuración de Remotos

- Para configurar un remoto en DVC, se utiliza el comando `dvc remote add`, especificando la URL del almacenamiento.
- Ejemplo:

```
$dvc remote add -d myremote s3://mybucket/dvcstore
```

añade un remoto llamado `myremote` y lo define como el remoto por defecto (`-d`).

# Sincronización de Datos

- Después de añadir un archivo y comprometer el cambio en Git, puedes enviar los datos al remoto con `dvc push`. Este comando sube los archivos versionados al almacenamiento remoto.
- Para recuperar datos desde un remoto, se utiliza `dvc pull`, que descarga los datos necesarios según el estado actual del repositorio Git.
- DVC también permite importar datos desde un remoto con `dvc import`, lo que facilita el trabajo con datasets externos sin necesidad de duplicarlos localmente.

# Ejemplo de uso

```
# Inicialización del proyecto DVC
dvc init

# Seguimiento de los datos
dvc add data/train.csv

# Almacenamiento remoto
dvc remote add s3 s3://my-bucket/

# Subir los datos a S3
dvc push

# Descargar los datos desde S3
dvc pull
```

# Beneficios de usar DVC

- **Reproducibilidad**: DVC asegura que cualquier persona pueda replicar los experimentos y obtener los mismos resultados utilizando los mismos datos y scripts.
- **Escalabilidad**: Facilita el manejo de grandes volúmenes de datos sin sobrecargar el repositorio Git.
- **Colaboración**: Permite que los equipos colaboren de manera efectiva en proyectos complejos, compartiendo tanto código como datos sin conflictos.