

# Gestión de código para machine learning



# ¿Qué es el control de versiones?



Control  
versiones final



Control  
versiones final 1



Control  
Versiones V2



Final Control  
Versiones



Versiones final  
final



Control  
Versiones V1



Control  
Versiones V3



Final Contr Ver



Final versión  
doc control v



draft control  
versiones



draft primero  
control ver



Control  
versiones

# Sistemas de control de versiones

- Ayudan llevando un registro de los cambios que sufren los componentes de un programa
- Nos permite aislar completamente el desarrollo de nuevas características
- Permite viajar en el tiempo y destacar cambios importantes



# Sistemas de control de versiones



git

# ¿Qué almacenar ahí?

- Se *puede* almacenar todo lo que se te ocurra: código fuente, configuraciones, imágenes, audio, videos...
- **PERO**, idealmente solo debes almacenar archivos de código fuente: para extraer datos, procesarlos, entrenar modelos, y ponerlos en producción



# ¿Qué NO almacenar ahí?

- Puede ser tentador almacenar todo en *git*, pero no te recomiendo que guardes grandes datasets o archivos binarios como modelos ya entrenados.
- Existen mejores alternativas para almacenar estas dos cosas.



# Data scientists y Git

- Los notebooks de Jupyter son la herramienta favorita en DS, pero estos pueden complicar el uso de Git
- Puedes simplemente poner todos tus notebooks ahí, pero debes estar consciente de que debes tener cuidado de lo guardas en los notebooks



# Limpia manualmente tus notebooks

- Antes de subirlos a *git*, limpia tus notebooks para que no contengan imágenes o grandes tablas de datos.



# Usa *nbdime*

- *nbdime* es una herramienta que te permite ver las diferencias entre notebooks de forma clara, sin necesidad de limpiar los notebooks
- Inclusive cuenta con una interfaz web para fácil navegación.



# Usa *jupytext*

- *jupytext* es otra herramienta que permite crear versiones “solo texto” de tus notebooks, para que no te preocupes por limpieza manual o sobrecargar *git*



# Reduce tu uso de notebooks

- ▶ Los notebooks son excelentes para el prototipado, análisis y presentación de resultados, pero son poco consumibles de otro modo.
- ▶ Procura abstraer funciones en módulos fuera de los *ipynb*



# Consejos para gestionar código fuente

- Limpia tus notebooks
- Usa *nbdime* o *jupytext*
- Abstrai y extrae funciones

