Cómo Revisar una notebook para que el versionado de la misma implique su reproducibilidad

Usamos Jupyter Lab como herramienta oficial para resolver tickets que impliquen el análisis/experimentación con datos y la creación de una notebook.

Consideramos de mucha importancia poder repetir la ejecución y obtener los mismos resultados que en el reporte de la notebook, por lo que esperamos que quien revise la notebook:

1. Disponga acceso a los datos que se utilizaron (Ya sea acceso vía query o los datos en algún storage compartido como s3). Tanto las queries o paths a los archivos en el storage deberían encontrarse en la notebook.
2. Pueda ejecutar la notebook completa sin problemas (previa instalación de requerimientos), obteniendo los mismos resultados que los que se presentan en el reporte. Para esto es importante que todo paquete python de terceros que se utilice en la notebook se encuentre en el archivo *requirements.txt.*
3. Revise que las funciones útiles que solo aplican a la notebook se encuentren dentro de la misma, y que funciones útiles que podrían reutilizarse por más notebooks se encuentren en algún módulo de utilidades accesible por más notebooks.
4. Revise la duplicación de código (Deberían definirse funciones, de interfaz simple y con la menor dependencia de librerías de terceros posible).

Toda notebook que se utilice para resolver un ticket debería ser versionada en el repositorio del proyecto, siguiendo los pasos usuales de creación de pull requests (gitflow). Para notebooks además de versionar los .ipynb con código, estamos versionando la misma notebook en formato .md, para facilitar la revisión de código. Disponemos de un pre-commit hook, que nos facilita esta conversión (ver proyecto open-source jupytext). Por lo que es importante que al clonar el repo corramos ***$ pre-commit install*** *,* para disponer de este hook. Luego, cada vez que intentemos commitear una notebook en formato *.ipynb* el pre-commit debería encargarse de agregar automáticamente un archivo en formato *.md*

Por último, es **importante** que tanto el branch como el nombre del PR incluya el identificador de ticket ***PROJ-N***, ya que esto nos permite volver hacia atrás (vía *`git checkout <merge-commit>`,* por ejemplo) a ejecutar una notebook con los mismos requerimientos y utilidades con los que se definió en su momento para resolver dicho ticket.