Reproducible machine learning

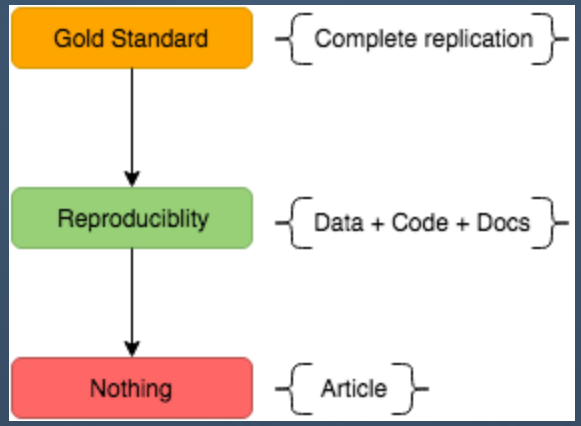
Aneesh Karve, @akarve
Adam Sah, adam@quiltdata.io

Who we are

Triangle of reproducibility

- code
- data
- models

Reproduction vs Replication



[AP]

Why

- trust
- compliance
- auditing
- teams
- banking, healthcare, transportation, insurance

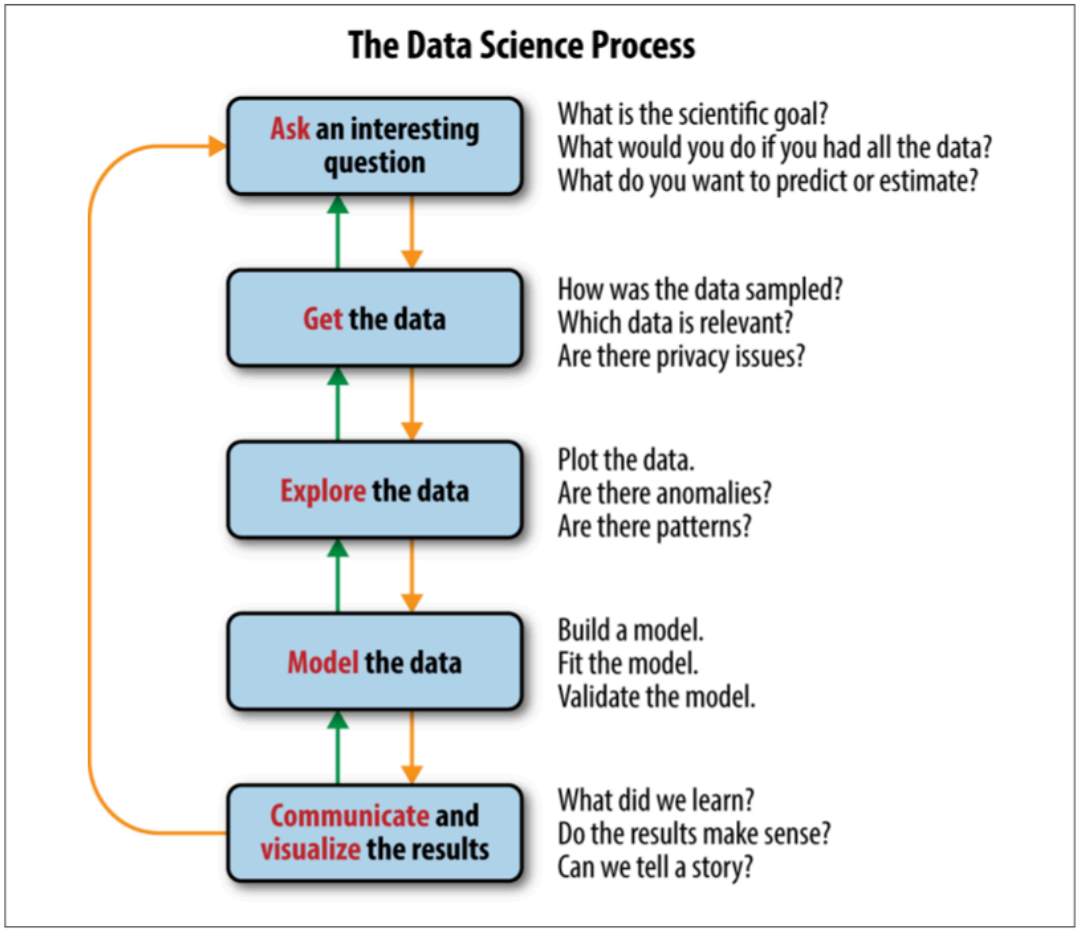


Figure 1-1. One representation of the data science process (courtesy of Joe Blitzstein and Hanspeter Pfister, created for the Harvard data science course)

[CB]

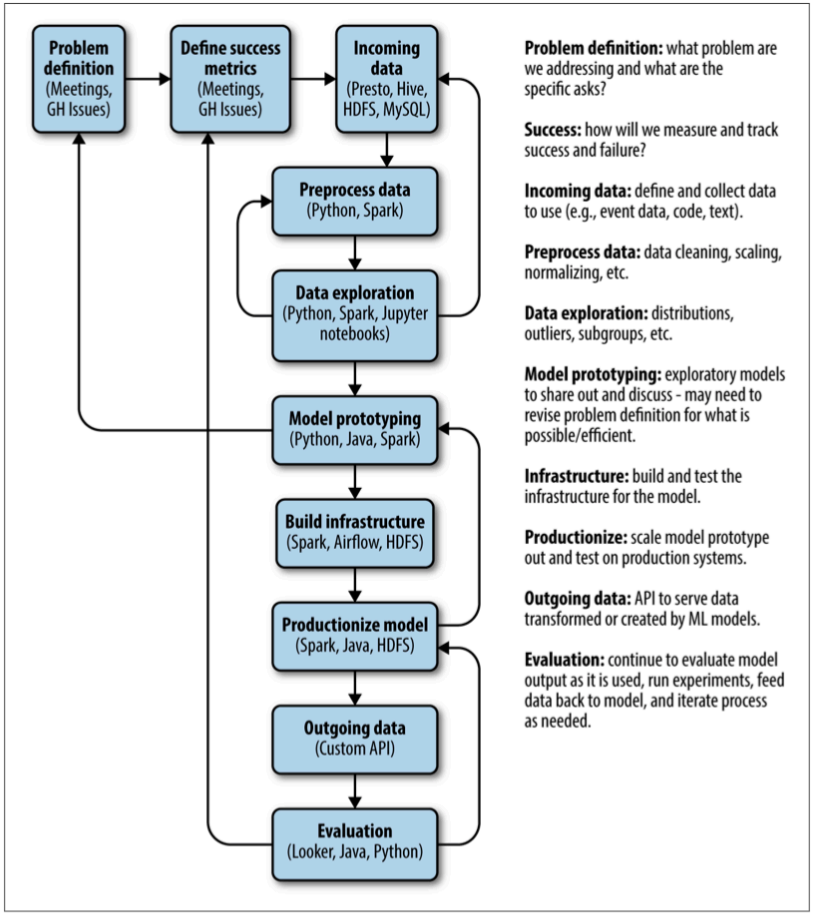


Figure 1-2. The data science workflow of GitHub's machine learning team

[CB]

Out of scope

- Deploying models to production
- Big data (Spark, Hadoop)
- But many of the tools discussed still apply

Tools for reproducibility

- code = conda + git + jupyter + *Docker
- data = pandas + quilt
- models = scikit + TensorFlow + quilt

conda

- packages + environments
- run from the command line
- do not install in root env
- `conda create -n MY_ENV python=3.6`
- `pip freeze`

pip or conda?

- pip - python packages in any env
- conda - any package in conda envs
- Never do this: `sudo pip install`

Don't put data in git

- Slows down repo
- Does not serialize
- Does not quickly handle large data (incl. LFS)
- Conflates code and data changes

Demo

TensorFlow

- Alpha - in development
- Virtual File System (in addition to standard API)
- Release in 2-4 weeks

Demo

Resources

https://github.com/quiltdata/reproducible-ml

Further learning

- A tour of artificial intelligence and its limitations
- [CB] Development Workflows for Data Scientists, Claire Byrne
- [AP] Reproducible Research in Computational Sciences, Arnu Pretorius
- [JV] Installing Python Packages from a Jupyter Notebook, Jake VanderPlas
- [TM] Machine Learning, Tom Mitchell