

Reproducible Machine Learning

Dr. Rachael Tatman, Data Scientist

August 8, 2018

@rctatman



.@BaumerBen: Reproducibility, or the idea that the same people should be able to reproduce the same analysis with the same data, is such a low bar... and we're still tripping over it. #JSM2018

10:58 AM - 31 Jul 2018

Why should you care about reproducibility?

Because the person most likely to need to reproduce your work... is you.

code + data + environment =
reproducible machine learning

code + data + environment =
reproducible machine learning

+ some ML-specific stuff

Levelling up reproducibility

1. Code

- a. Structuring your project

- b. Stochastic -> static

2. Data

3. Environment

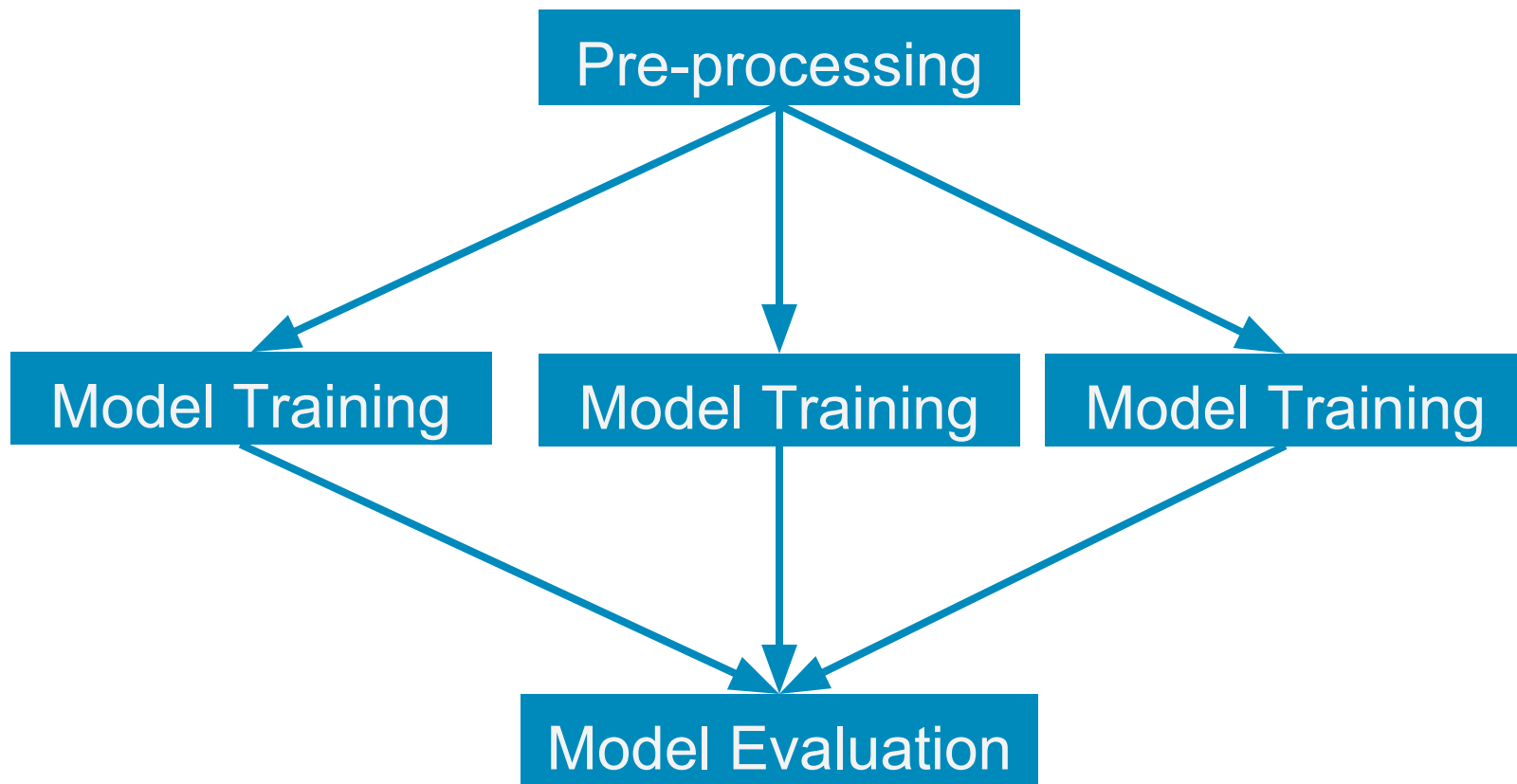
Pre-processing

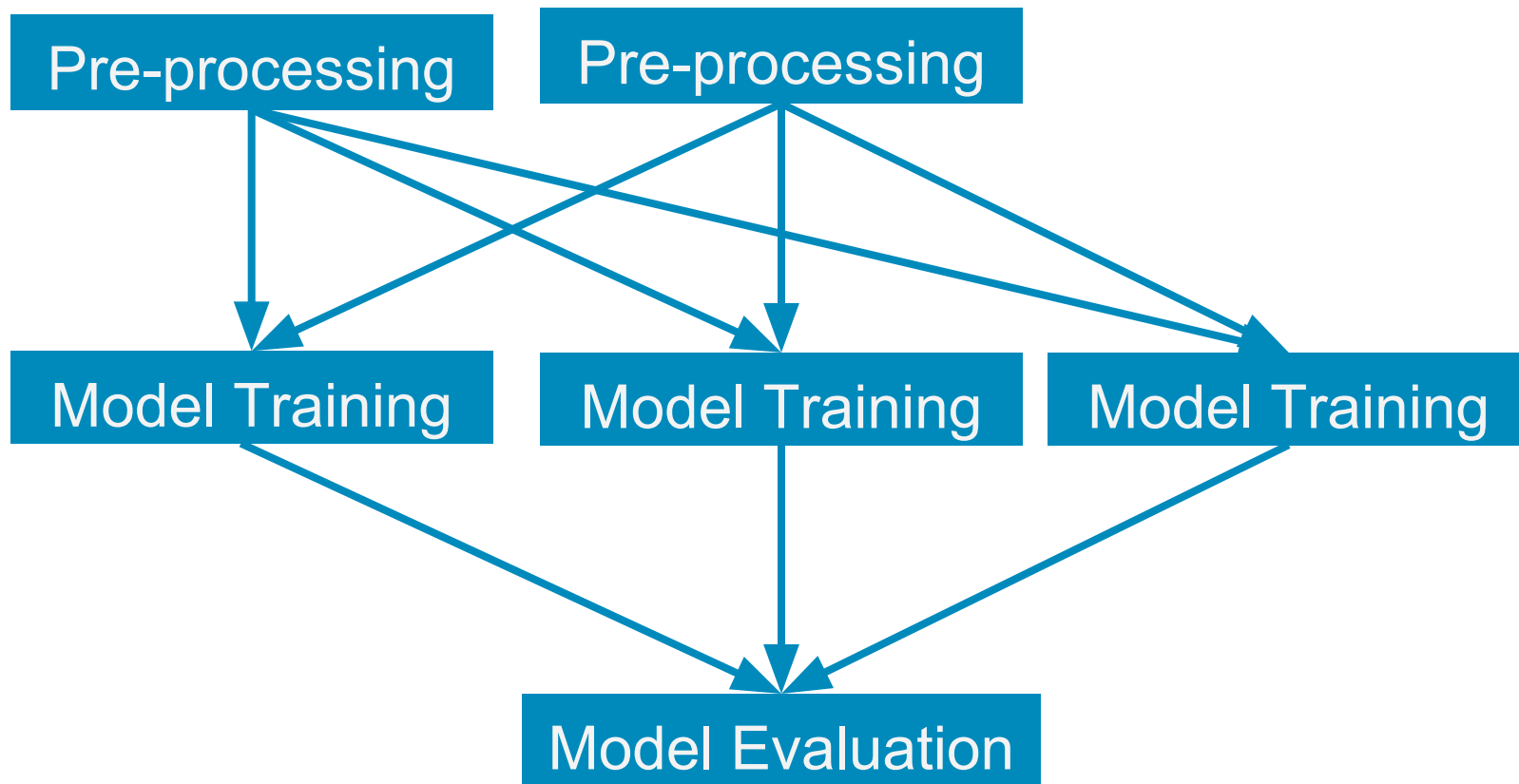


Model Training



Model Evaluation





Stochastic -> Static

Help! I'm getting different results with the same code!

Most machine learning methods rely on some sort of pseudorandomness for things like:

- Weight initialization
- Dropout
- Subsetting/shuffling for mini-batches
- Training/testing/validation split

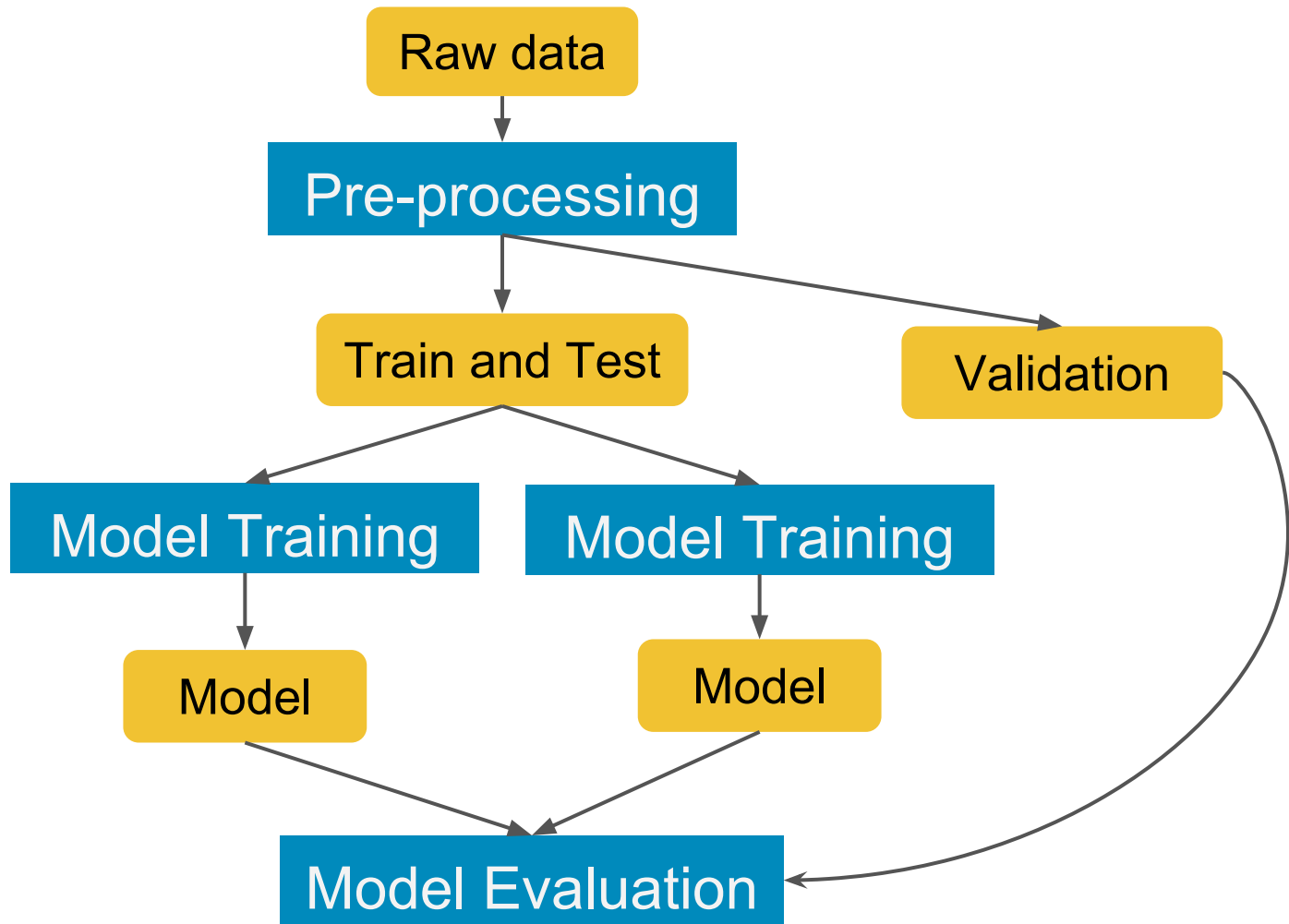
Stochastic -> Static

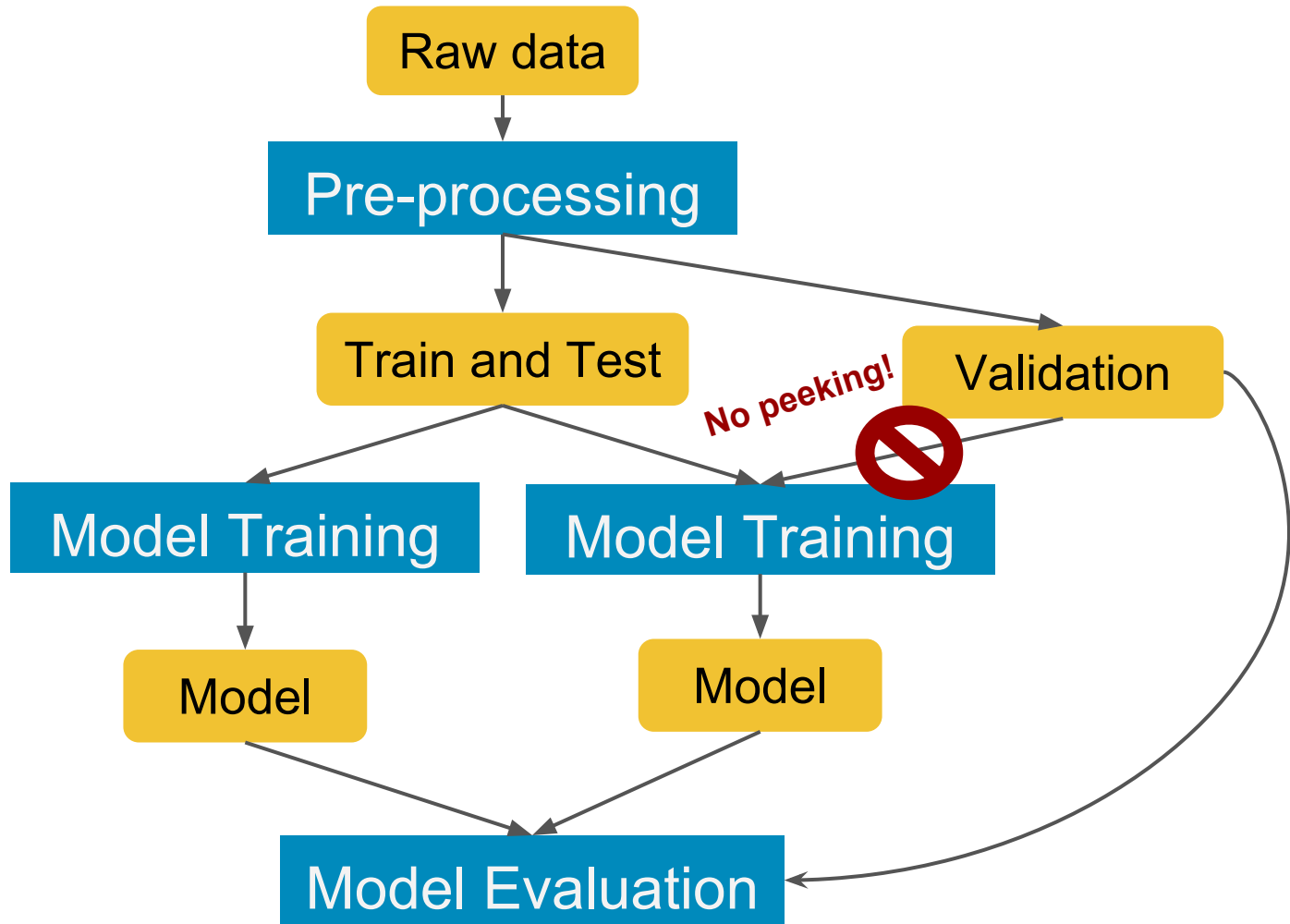
In order to get the same results more than once, you need to make sure to set *all* the random number generators (RNGs) your code depends on.

- Numpy & Keras:
 - `np.random.seed(42)` (<- will bork on multi-threading)
- Tensorflow:
 - `tf.set_random_seed(42)`
- Anything that uses hash randomization (like Theano):
 - `PYTHONHASHSEED=0` (<- this can be a security risk so don't do it by default)
- cuDNN
 - `___(ツ)_/__`
 - Some routines (like backward pass & backward pooling) do not guarantee reproducibility because they use atomic operations

Levelling up reproducibility

1. Code
- 2. Data**
 - a. What should you be saving?**
 - b. Always work from a copy**
 - c. Version your data**
3. Environment





Always work from a copy of your data

Keep a separate copy of your raw data that you never, ever touch. Work from copies of it.

Versioning, not just for code anymore!

- If you're already using version control & have small/medium data, add your data files to the system you use for your code
- For databases, there are many version control options (versionSQL, DBmaestro, etc.)
- For streaming data, save & work from a specific time span

Levelling up reproducibility

1. Code
2. Data
- 3. Environment**

You can think of reproducibility as a scale:
The longer it takes to reproduce a project,
the less reproducible it is.

Sharing your environment

Your environment includes:

- All dependencies, including versions and subversions
- Your file system
- Your OS
- (In some cases) hardware

Options for sharing your environment:

- Containers (like Docker)
- Virtual machines
- Hosted environments

Sharing your environment: Containers

Benefits:

- Contains data, code, file systems, dependencies
- Portable
- Lightweight

Drawbacks:

- Uses the host OS, can be dicey cross-platform
- Can take a while to get set up



Sharing your environment: Virtual machines

Benefits:

- Contains data, code, file systems, dependencies and OS
- Portable, even between platforms

Drawbacks:

- Larger/slower to get started than containers
- Can take a while to get set up



Sharing your environment: Hosted environments

Benefits:

- Very fast set-up
- Extremely easy to share (in many cases just copying & pasting a link)

Drawbacks:

- Less control over environment
- May not be feasible for security/privacy reasons



Name	Price	Languages	GPU	Data hosting	Specs (Free tier)
Kaggle Kernels	Free	Python 3, R	Yes	Yes	GPU: 1xTesla K80 (6 hr/run) RAM: 16 GB Disk: 5 GB
Google Colaboratory	Free	Python 2 & 3	Yes	No	GPU: 1xTesla K80 (12 hr/run) RAM: ~12.6 GB Available Disk: ~33 GB Available
Azure Notebooks	Free	Python 2 & 3, R, F#	No	Yes	4G memory limit & 1G data limit
Amazon SageMaker	Varies, 2 month free trial	Python 2 & 3	Yes	No	250 hours/month of t2.medium notebook 50 hours/month of m4.xlarge 125 hours/month of m4.xlarge
IBM Watson Studio	Varies, limited free tier	Python, R, Scala	Yes	Yes	1 vCPU and 4 GB RAM, 50 hours run-time per month
Codalab	Free	Any	Yes	Yes	non-GPU machine has 4 cores and 14 GB of memory, and each GPU machine has 6 cores and 56 GB of memory
MyBinder	Free	Python 2 & 3, R, Julia	No	No	CPU: x86-64 RAM: ~13 GB (est) Disk: ~100 GB (est)

Name	Price	Languages	GPU	Data hosting	Specs (Free tier)
Kaggle Kernels	Free	Python 3, R	Yes	Yes	GPU: 1xTesla K80 (6 hr/run) RAM: 16 GB Disk: 5 GB
Google Colaboratory	Free	Python 2 & 3	Yes	Yes	GPU: 1xTesla K80 (12 hr/run) RAM: ~12.6 GB Available Disk: ~5 GB Available
Azure Notebooks	Free	Python 2 & 3	Yes	Yes	GPU: 1xTesla K80 (12 hr/run) RAM: ~12.6 GB Available Disk: ~5 GB Available
Amazon SageMaker	Varies, 2 month free trial	Python 2 & 3	Yes	Yes	GPU: 1xTesla K80 (12 hr/run) RAM: ~12.6 GB Available Disk: ~5 GB Available
IBM Watson Studio	Varies, limited free tier	Python, R, Scala	Yes	Yes	GPU: 1xTesla K80 (12 hr/run) RAM: ~12.6 GB Available Disk: ~5 GB Available
Codalab	Free	Any	Yes	Yes	non-GPU: 4 cores and 14 GB of memory, and each GPU machine has 6 cores and 56 GB of memory
MyBinder	Free	Python 2 & 3, R, Julia	No	No	CPU: x86-64 RAM: ~13 GB (est) Disk: ~100 GB (est)

Hosting services change their services from time to time, so be sure to double check the before starting your project!

The Kaggle logo, consisting of the word "kaggle" in a lowercase, sans-serif font, with a small trademark symbol (TM) to the upper right of the "e".

kaggle™

Slides: <https://goo.gl/4kqNzq>

Contact: rachael@kaggle.com
[@rctatman](https://twitter.com/rctatman)